С.Г. СЕМЕНОВ, А.О.ПОДОРОЖНЯК, О.І.БАЛЕНКО, С.Ю.ГАВРИЛЕНКО

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ ТА МЕРЕЖАХ

Харків
НТУ «ХПІ»
2014

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

С. Г. СЕМЕНОВ, А. О. ПОДОРОЖНЯК, О. І. БАЛЕНКО, С. Ю. ГАВРИЛЕНКО

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ ТА МЕРЕЖАХ

Навчальний посібник

Рекомендовано Міністерством освіти і науки України для студентів вищих навчальних закладів, що навчаються за освітнім напрямком «Комп'ютерна інженерія»

Харків
НТУ «ХПІ»
2014

УДК 004.7: 621.391 (075)
ББК 32.811.3
З-38

Рецензенти:

Г.А.Рудницький, д-р техн. наук, проф., Черкаський Державний технічний університет;

Л.І.Нефьодов, д-р техн. наук, проф., Харківський національний автомобільно-дорожний університет;

Г.В.Кривуля, д-р техн. наук, проф. Харківський національний університет радіоелектроніки.

Авторський колектив:

Семенов С. Г., д-р техн. наук, с.н.с., розділи 1–4 (спільно з Подорожняком А. О., Баленко О. І., Гавриленко С. Ю.) ;

Подорожняк А.О., канд. техн. наук, доц. розділи 1–3 (спільно з Семеновим С. Г., Баленко О. .І.);

Баленко О. І. канд. техн. наук, доц. розділи 1–2 (спільно з Семеновим С. Г, Подорожняком А. О)

Гавриленко С.Ю. канд. техн. наук, доц. розділ 4 (спільно з Семеновим С. Г.)

Рекомендовано Міністерством освіти і науки України як навчальний посібник для студентів вищих навчальних закладів, що навчаються за освітнім напрямком «Комп'ютерна інженерія», лист № 1/11-7520 від 20 травня 2014 р.

З-38 Захист інформації в комп'ютерних системах та мережах : навч. посіб. / С.Г.Семенов, А.О.Подорожняк, О.І.Баленко, С.Ю.Гавриленко – Х.: НТУ «ХПІ», 2014.– 251 с.

ISBN 978-966-8944-71-0

У навчальному посібнику викладено основні концептуальні питання створення, функціонування, розвитку і використання національної системи конфіденційного зв'язку. Розглянуто системи захисту інформації на різних рівнях моделі взаємодії відкритих систем, основні загрози інформаційної безпеки комп'ютерних систем і мереж, а також описано протоколи та засоби захисту інформації у мережі Інтернет.

Призначено для студентів денної та заочної форм навчання напрямків «Комп'ютерна інженерія», «Комп'ютерні науки» та «Інформаційна безпека».

Іл. 149 . Табл. 18. Бібліогр. 30 назв.

УДК 004.7: 621.391 (075)
ББК 32.811.3

ISBN 978-966-8944-71-0

ЗМІСТ

ВСТУП

Великі вимоги до своєчасності, достовірності і скритності інформаційних процесів у різних галузях діяльності сучасного суспільства, а також розширення можливостей обчислювальної техніки привели до удосконалення і впровадження методів розподіленої обробки даних за рахунок реалізації мережевого доступу до комп'ютерних систем. Найбільш широкого застосування такі системи та мережі набули в так званих сферах критичного застосування, до яких належить діяльність військово-промислового комплексу, інститутів державної влади, правоохоронних органів, фінансових структур, енергетики, транспорту і та ін. Не дивлячись на очевидну різнорідність сфер критичного застосування, їх об'єднує одна дуже важлива обставина − значний збиток від порушення інформаційної безпеки. Таким чином, цінність даних, що зберігаються і оброблюються, в комп'ютерних системах і мережах, зумовила розробку і вдосконалення методів, засобів і протоколів захисту інформації.

Метою даного навчального посібника є аналіз послуг і механізмів захисту інформації, а також особливостей їх реалізації на різних рівнях моделі взаємодії відкритих систем, розгляд основних концептуальних питань створення, функціонування, розвитку і використання національної системи конфіденційного зв'язку, вивчення основних загроз інформаційної безпеки комп’ютерних систем і мереж, а також протоколів та засобів захисту інформації в мережі Інтернет.

У навчальному посібнику системно викладені особливості забезпечення основних послуг безпеки, а також основні напрями розв’язання проблеми захисту інформації в комп'ютерних системах і мережах.

У посібнику подані:

- аналіз послуг і механізмів захисту інформації, а також особливостей їх реалізації на різних рівнях моделі взаємодії відкритих систем;
- концептуальні питання створення, функціонування, розвитку і використання національної системи конфіденційного зв'язку;
- понятійний апарат, який використовується при описі послуг і механізмів захисту інформації;
- особливості розподілу послуг безпеки по рівнях моделі взаємодії відкритих систем;
- класифікація загроз інформаційної безпеки комп'ютерних систем і мереж;
- основні системи захисту інформації в комп'ютерній мережі Інтернет;
- основні принципи забезпечення безпеки та особливості реалізації систем захисту інформації в телекомунікаційній мережі Інтернет;
- методи захисту локальних комп’ютерних мереж за допомогою брандмауерів.

Новизна навчального посібника полягає в тому, що в ньому на основі аналізу існуючих теоретичних підходів до захисту інформації в комп'ютерних системах і мережах здійснено системний виклад основних механізмів і протоколів забезпечення їх інформаційної безпеки.

Наданий в навчальному посібнику матеріал дозволить фахівцям значно підвищити обґрунтованість управлінських рішень з організації діяльності в сферах критичного застосування, виробити практичні рекомендації щодо впровадження нових інформаційних технологій, досліджувати шляхи побудови захищених комп'ютерних систем і мереж. Крім того, навчальний посібник може бути корисним фахівцям з інформаційної безпеки, в коло завдань яких входить виявлення і припинення злочинних посягань на інформацію.

Зміст навчального посібника побудований на відкритих матеріалах вітчизняних і зарубіжних літературних джерел, а також авторських досліджень у сфері захисту інформації в комп’ютерних системах і мережах.

Навчальний посібник призначений для підготовки сучасних фахівців за напрямком “Комп’ютерна інженерія”, “Комп’ютерні науки” та “Інформаційна безпека” і відповідає навчальній програмі дисциплін «Захист інформації в комп’ютерних системах та мережах» і «Аналіз і синтез захищених комп’ютерних систем та мереж».

1. ОСНОВИ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Розробка нових інформаційних технологій, їх впровадження в усі сфери життєдіяльності сучасного суспільства обумовили виникнення нових загроз інформаційної безпеки. Комплексне рішення задач захисту національного інформаційного простору України і забезпечення інформаційної безпеки найбільш важливих (критичних) систем управління і комп'ютерних мереж покладається на Національну систему конфіденційного зв'язку.

У даному розділі розглянуто концептуальні питання створення, функціонування, розвитку і використання Національної системи конфіденційного зв'язку, наведено загальну класифікацію загроз безпеки інформації, проаналізовано їх дію на різні інформаційні системи і технології. Відповідно до основних положень міжнародних стандартів у галузі захисту інформації наведено загальну класифікацію послуг і механізмів щодо забезпечення безпеки інформації, розглянуто аспекти їх реалізації на різних рівнях еталонної моделі взаємодії відкритих систем. Наведено приклади криптографічних і технічних механізмів захисту інформації в комп'ютерних системах і мережах.

1.1. Концептуальні питання створення, функціонування, розвитку і використання національної системи конфіденційного зв'язку

Існування і прогресивний розвиток України як суверенної держави залежить від здійснення цілеспрямованої політики захисту її національних інтересів. Основи такої політики визначає Концепція національної безпеки України.

Національна безпека України як стан захищеності життєво важливих інтересів фізичних осіб, суспільства і держави від внутрішніх і зовнішніх загроз є необхідною умовою збереження і примноження духовних і матеріальних цінностей. В умовах стрімкої інформатизації суспільства, широкого застосування засобів обчислювальної техніки і комп'ютерних систем особливої актуальності набувають питання інформаційної безпеки держави, найбільш складними з яких є необхідність захисту цінної конфіденційної і секретної інформації в державних і приватних підприємствах, в органах і установах державного управління, банківській та інших системах.

Соціально-економічні наслідки інформатизації сучасного суспільства відокремили ряд питань інформаційної безпеки, наприклад:

- необхідність захисту майнових прав громадян, підприємств і держави на інформацію і обчислювальні ресурси відповідно до вимог цивільного, адміністративного і господарського права;

- необхідність забезпечення ефективного функціонування найважливіших (критичних) автоматизованих і інформаційних систем і технологій, яка обумовлена вимогами фізичної безпеки людей, екологічної обстановки, збереження духовних і матеріальних цінностей та ін.;
- необхідність захисту цивільних прав і свобод, гарантованих чинним законодавством та ін.

Серед найбільших загроз національній безпеці України в інформаційній сфері вважаються:

- незважена державна політика і відсутність необхідної інфраструктури в інформаційній сфері;
- повільне входження України в світовий інформаційний простір, недостатньо об'єктивне уявлення про Україну серед міжнародного співтовариства;
- інформаційна експансія з боку інших держав;
- витік інформації, що становить державну та іншу, передбачену законом, таємницю, а також конфіденційної інформації, яка є власністю держави;
- введення цензури.

Основними напрямками державної політики Національної безпеки України в інформаційній сфері є:

- застосування комплексних заходів щодо захисту інформаційного простору і вхід України до світового інформаційного простору;
- виявлення і усунення причин інформаційної дискримінації України;
- усунення негативних чинників порушення інформаційного простору, інформаційної експансії з боку інших держав;
- розроблення і впровадження необхідних засобів і режимів отримання, збереження, розповсюдження і використання суспільно значущої інформації, створення розвиненої інфраструктури в інформаційній сфері.

Державна політика національної безпеки визначається пріоритетами національних інтересів і загроз національній безпеці України. Вона здійснюється шляхом реалізації відповідних доктрин, стратегій, концепцій і програм у різних сферах національної безпеки відповідно до чинного законодавства. Основним комплексним заходом щодо захисту національного інформаційного простору України є побудова Національної системи конфіденційного зв'язку.

Національна система конфіденційного зв'язку – це сукупність спеціальних систем (мереж) зв'язку подвійного призначення, які за допомогою криптографічних та/або технічних засобів забезпечують обмін конфіденційною інформацією на користь органів державної влади і органів місцевого самоврядування, створюють належні умови для їх взаємодії в мирний час і у

разі введення особливого чи військового стану. Національна система конфіденційного зв'язку входить до складу Єдиної національної системи зв'язку України.

Складовими Національної системи конфіденційного зв'язку є:

- спеціальні системи (мережі) зв'язку;
- стаціонарні і мобільні компоненти спеціальних систем (мереж) зв'язку;
- централізовані системи захисту інформації і оперативно-технічного управління.

Спеціальна система (мережа) зв'язку – система (мережа) зв'язку, призначена для обміну інформацією з обмеженим доступом.

Спеціальна система (мережа) зв'язку подвійного призначення – спеціальна система (мережа) зв'язку, призначена для забезпечення зв'язку на користь органів державної влади і органів місцевого самоврядування з використанням частини її ресурсу для надання послуг іншим споживачам.

*Суб'єктами Національної системи конфіденційного зв'яз*ку є органи державної влади і органи місцевого самоврядування, юридичні і фізичні особи, які беруть участь у створенні, функціонуванні, розвитку і використанні цієї системи.

Управління Національною системою конфіденційного зв'язку, її функціонування, розвиток, використання і захист інформації забезпечуються спеціально уповноваженим центральним органом виконавчої влади у сфері конфіденційного зв'язку відповідно до законодавства. Централізовані системи захисту інформації і оперативно-технічного управління знаходяться в державній власності і не підлягають приватизації. Власниками інших компонентів Національної системи конфіденційного зв'язку можуть бути суб'єкти господарської діяльності, незалежно від форми власності.

Захист інформації в структурі Національної системи конфіденційного зв'язку забезпечується:

- дотриманням суб'єктами правового відношення норм, вимог і правил організаційного і технічного характеру щодо захисту оброблюваної інформації;
- використанням засобів обчислювальної техніки, програмного забезпечення, засобів зв'язку і автоматизованих систем у цілому, засобів захисту інформації, які відповідають установленим вимогам до захисту інформації (мають відповідний сертифікат);
- перевіркою відповідності засобів обчислювальної техніки, програмного забезпечення, засобів зв'язку і автоматизованих систем у цілому встановленим вимогам до захисту інформації (сертифікація засобів обчислювальної техніки, засобів зв'язку і автоматизованих систем);
- здійсненням контролю захисту інформації.

Таким чином, відповідно до основних нормативно-правових актів України рішення задач захисту національного інформаційного простору України покладається на Національну систему конфіденційного зв'язку, концептуальні питання створення, функціонування, розвитку і використання якої регулюються Конституцією України, законами України "Про інформацію", "Про державну таємницю", "Про захист інформації в автоматизованих системах", "Про зв'язок", "Про підприємництво", "Про ліцензування певних видів господарської діяльності", "Про Національну систему конфіденційного зв'язку".

Відповідно до основних функціональних завдань використання Національної системи конфіденційного зв'язку повинне забезпечувати ефективний захист цінної конфіденційної і секретної інформації в державних і приватних підприємствах, в органах і установах державного управління, банківській та інших системах.

Проведемо аналіз можливих загроз безпеки, розглянемо їх дію на інформаційні системи й технології. Виходячи з принципів і положень державної політики забезпечення інформаційної безпеки, найбільшу небезпеку становлять загрози в політичній, економічній, оборонній і інших сферах діяльності держави (рис. 1.1).

У політичній сфері найбільш серйозній небезпеці підлягають:

- суспільна свідомість і політична орієнтація різних груп населення країни (регіонів), що безперервно формуються під впливом вітчизняних та іноземних засобів масової інформації (друк, радіо, телебачення);
- система ухвалення політичних рішень, що істотно залежить від якості і своєчасності її інформаційного забезпечення;
- право політичних організацій, партій, об'єднань і рухів на вільне вираження своїх соціально-політичних і економічних програм через засоби масової інформації;
- система регулярного інформування населення органами державної влади і управління про політичне і соціально-економічне життя через засоби масової інформації, прес-центри, центри суспільних зв'язків та ін.;
- система формування громадської думки, що включає спеціальні інститути, центри і служби виявлення, вивчення і аналізу громадської думки.

Рисунок 1.1 – Критичні інформаційні системи і технології в різних сферах

Рисунок 1.2 – Основні загрози безпеки інформації

У сфері економіки найбільш схильні до дії загроз інформаційної безпеки: система державної статистики, джерела, що надають інформацію про комерційну діяльність господарських суб'єктів усіх форм власності, про споживчі властивості товарів і послуг, системи збору і обробки фінансової, біржової, податкової, митної інформації, інформації про зовнішньоекономічну діяльність держави і комерційних структур.

В оборонній сфері до найуразливіших ланок належать:

- інформаційні ресурси апарату Міністерства оборони, Генерального штабу, Головних штабів видів Збройних сил і родів військ, науково-дослідних установ, що містять відомості і дані про оперативні і стратегічні плани підготовки і ведення бойових дій, про склад і дислокацію військ, про мобілізаційну готовність, тактико-технічні характеристики озброєння і військової техніки;
- інформаційні ресурси підприємств оборонного комплексу, що містять відомості про науково-технічний і виробничий потенціал, про обсяги постачань і запаси стратегічних видів сировини і матеріалів, основні напрями розвитку озброєння, військової техніки, їх бойові можливості й фундаментальні і прикладні науково-дослідні роботи, що проводяться в інтересах Міністерства оборони;
- системи зв'язку і управління військами і зброєю, їх інформаційне забезпечення;
- політична стійкість військ щодо інформаційно-пропагандистської дії;
- інформаційна інфраструктура, зокрема, центри обробки та аналізу інформації Генерального штабу та інформаційні підрозділи штабів видів Збройних сил, штабів об'єднань і з'єднань видів Збройних сил і родів військ, пункти управління, вузли і лінії радіозв'язку, радіорелейного, тропосферного і супутникового, а також лінії проводового зв'язку, що розгортаються і орендуються Міністерством оборони та іншими силовими структурами.

Під загрозою в широкому розумінні зазвичай розуміють потенційно можливу подію, дію (вплив), процес або явище, які можуть призвести до нанесення збитку будь-якій зі сторін. Загрозою інтересам суб'єктів інформаційних відносин називають потенційно можливу подію, процес або явище, яке за допомогою дії на інформацію або інші компоненти інформаційної системи може прямо чи опосередковано призвести до завдання збитку.

У процесі зберігання і обробки інформація може піддатися діям чинників випадковим або умисним. Найчастішими і найнебезпечнішими, з огляду на розмір збитків, є ненавмисні помилки користувачів, операторів, системних адміністраторів й інших осіб, що обслуговують інформаційні системи, а також помилки, що виникають при обробці і передачі інформації. Згідно зі статистикою 65 % втрат – наслідок ненавмисних помилок. Найрадикальніший

спосіб боротьби з ненавмисними помилками – максимальна автоматизація і суворий контроль за правильністю виконуваних дій.

Загрози збоку навколишнього середовища відрізняються великою різноманітністю. В першу чергу це порушення інфраструктури – аварії в системі електроживлення, тимчасова відсутність зв'язку, перебої з водопостачанням, цивільні безлади та ін. Особливу небезпеку становлять стихійні лиха: пожежі, повені, землетруси, урагани. За даними статистики на ці загрози доводиться 13 % втрат, завданих інформаційним системам. На рис. 1.2 проілюстровано в загальному вигляді класифікацію загроз інформації.

Особливої небезпеки останнім часом набули умисні загрози, реалізація яких цілком досяжна для терористичних груп і організацій. Потенційною мішенню злочинних намірів можуть бути інформаційні системи Міністерства оборони й інших силових структур, системи управління атомних, хімічних та інших небезпечних виробництв, обчислювальні системи банків і великих промислових підприємств. Реалізація умисних загроз може призвести до тяжких наслідків в обороні, промисловості, економіці, банківській сфері та інших галузях господарської діяльності, екології, житті і здоров'ї населення. Слід зазначити, що можливості реалізації умисних загроз останнім часом різко зросли. Це пояснюється стрімким розвитком обчислювальної техніки, зокрема, “суперкомп'ютерів”, широким розповсюдженням технологій розподілених обчислень.

Слід зазначити, що кількість і продуктивність сучасних суперкомп'ютерів останніми роками стрімко зросла. Відповідно до загальної тенденції розвитку обчислювальної техніки продуктивність комп'ютерів зростає вдесятеро кожні п'ять років.

У таблиці 1.1 наведено характеристики десяти наймогутніших суперкомп'ютерів у світі за станом на листопад 2013 року.

Таблиця 1.1 − Продуктивність десяти суперкомп'ютерів (TOP500)

	Приналежність	Комп'ютер	Кіл. процес.	TFLOPS
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2), Intel Xeon E5-2692 12C 2.200GHz NUDT	3120000	54902.4
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz Cray Inc.	560640	27112.5
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz IBM	1572864	20132.7
4	RIKEN Advanced Institute for Computational Science Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705024	11280.4
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz IBM	786432	10066.3
6	Swiss National Supercomputing Centre (CSCS) Switzerland	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115984	7788.9
7	Texas Advanced Computing Center/Univ. of Texas United States	Stampede - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Intel Xeon Phi SE10P Dell	462462	8520.1
8	Forschungszentrum Juelich (FZJ) Germany	JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz IBM	458752	5872.0
9	DOE/NNSA/LLNL United States	Vulcan - BlueGene/Q, Power BQC 16C 1.600GHz IBM	393216	5033.2
10	Leibniz Rechenzentrum Germany	SuperMUC - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz IBM	147456	3185.1

На рис. 1.3 наведено сумарну продуктивність суперкомп'ютерів по кожній країні з переліку TOP 500.

На рис. 1.4 наведено прогноз розвитку обчислювальних систем на період до 2023 року з зазначенням продуктивності, що досягається.

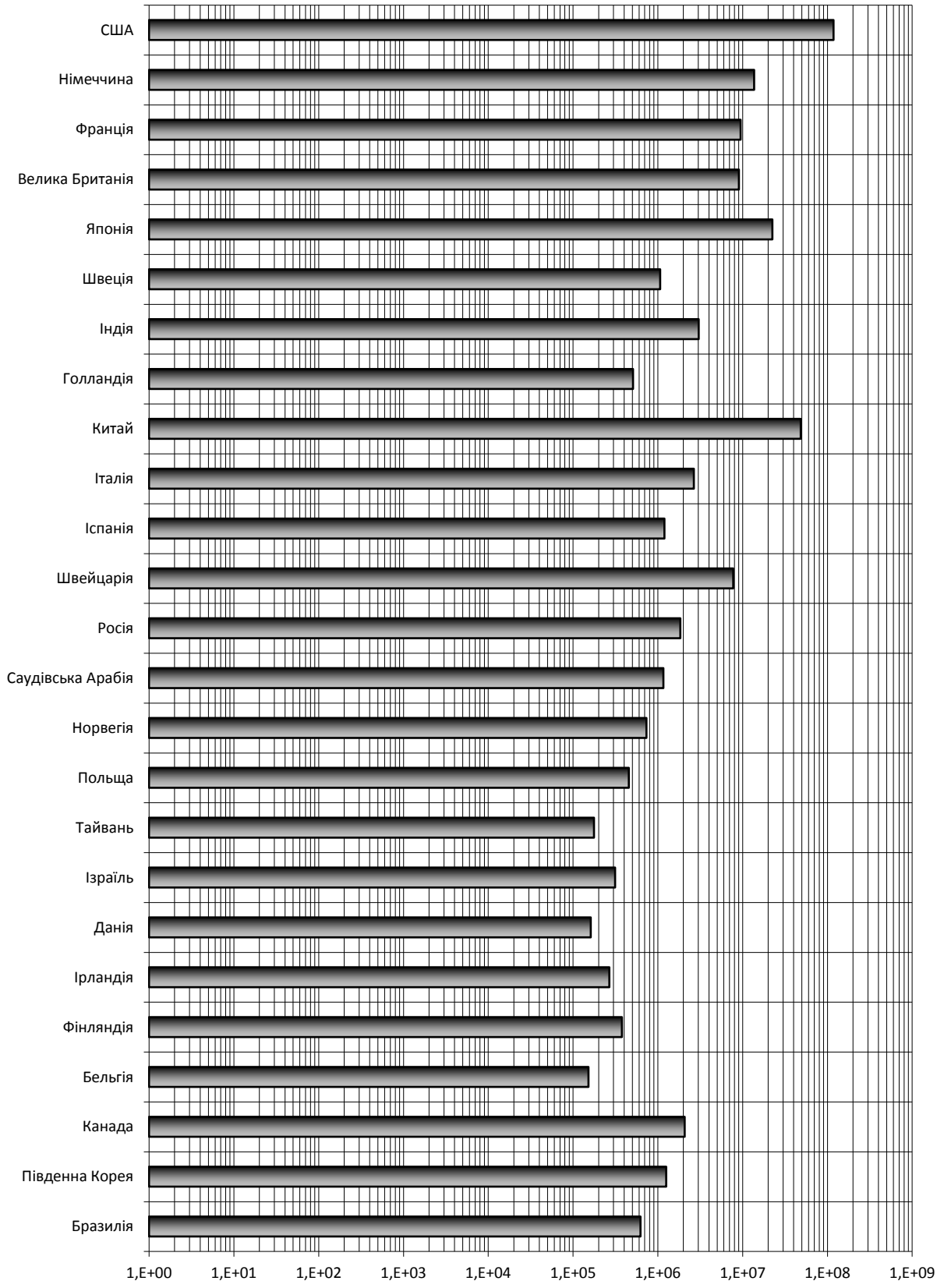


Рисунок 1.3 – Сумарна продуктивність суперкомп'ютерів по кожній країні з переліку TOP 500

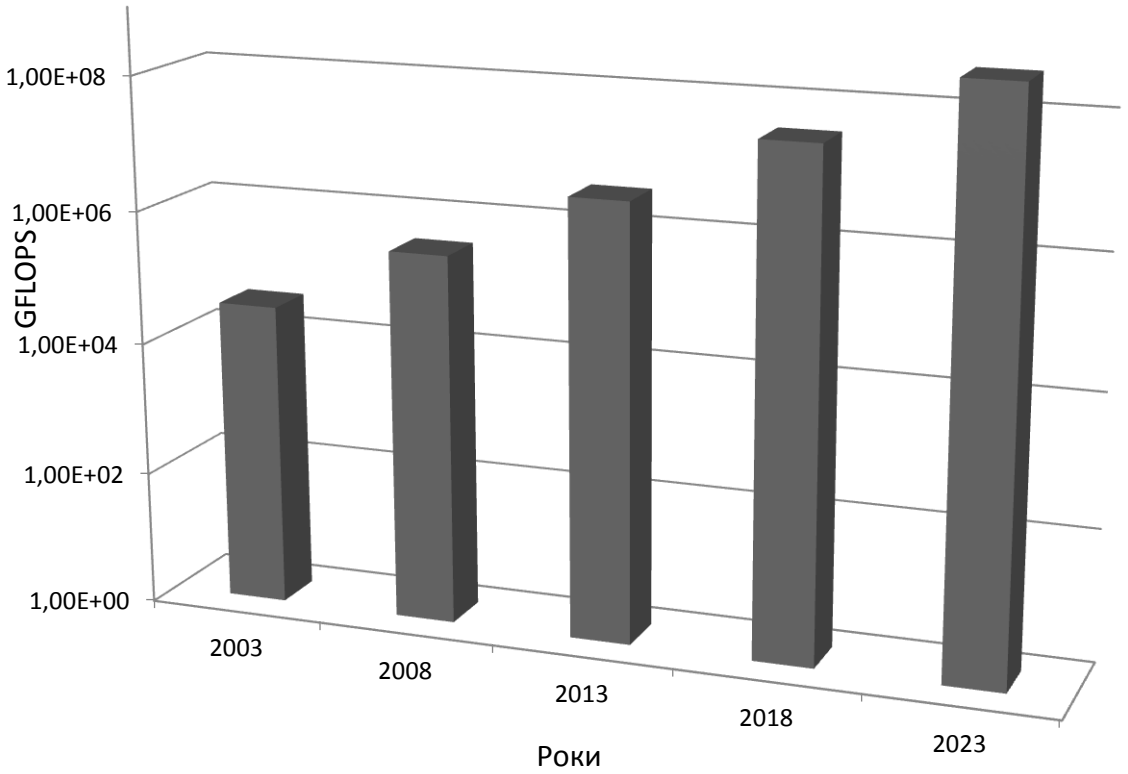


Рисунок 1.4 – Прогноз збільшення продуктивності суперкомп'ютерів на період до 2023 року

Аналіз наведених даних показує: сумарна продуктивність суперкомп'ютерів по країнах розрізняється на кілька порядків. Економічно розвинені країни (США, Китай, Японія, Німеччина, Франція та ін.) володіють обчислювальними системами величезної потужності. Їх сумарні можливості вже зараз перевершують 10^9 GFLOPS.

За станом на 2023 рік обчислювальні можливості окремих суперкомп'ютерів перевищуватимуть $3x10^9$ GFLOPS, що дозволить розкривати секретні ключі методом тотального перебору завдовжки до 80 біт. Зі зростанням кількості комп'ютерів, підключених до мережі Internet, зростає популярність і доступність проектів розподілених обчислень.

Сьогодні середня продуктивність деяких проектів розподілених обчислень перевищує $5x10^7$ GFLOPS, що можна зіставити за обчислювальними потужностями з найкращими суперкомп'ютерами світу. Вказані ресурси розподілених обчислень можуть бути доступними будь-якій фізичній та юридичній особі, здатній сплатити за функціонування проекту або надати переконливі аргументи до його використання. Іншими словами, сьогодні існує об'єктивна небезпека використання зловмисниками (наприклад, активними терористичними організаціями) обчислювальних ресурсів величезної потужності.

З урахуванням загальносвітової активізації терористичної діяльності в інформаційній сфері ця загроза є найбільш небезпечною. Отже, забезпечення інформаційної безпеки є актуальним завданням. Таким чином, сучасний розвиток інформаційних технологій, високий рівень комп'ютеризації й інформатизації сучасного суспільства зумовили виникнення нових загроз безпеки інформаци. Швидке зростання обсягів оброблюваних даних у сучасних комп'ютерних системах і мережах, створення нових форм і способів обробки інформації, стрімкий розвиток обчислювальної техніки висувають підвищені вимоги до криптографічних засобів захисту інформації.

Проаналізуємо існуючі послуги і механізми захисту інформації, розглянемо можливі підходи до забезпечення безпеки інформаційних систем і технологій.

1.2. Основні положення базового міжнародного стандарту ISO/IEC 15408 “Common Criteria”

1.2.1. Історія розробки

1990 рік: Міжнародна організація із стандартизації (ISO) розпочинає роботи із створення стандарту у сфері оцінки безпеки інформаційних технологій (ІТ). Основні цілі розробки:

- уніфікація національних стандартів у сфері оцінки безпеки ІТ;
- підвищення рівня довіри до оцінки безпеки ІТ;
- скорочення витрат на оцінку безпеки ІТ на основі взаємного визнання сертифікатів;

Червень 1993 року: організації із стандартизації та забезпеченню безпеки США, Канади, Великої Британії, Франції, Германії та Нідерландів об’єднали свої зусилля в рамках проекту із створення єдиної сукупності критеріїв оцінки безпеки ІТ. Цей проект отримав назву “Загальні критерії" (ЗК). Задача Загальних критеріїв – забезпечити взаємне визнання результатів стандартизованої оцінки безпеки на світовому ринку ІТ.

Січень 1996 року: Завершено розробку версії 1.0 “Загальних критеріїв”.

Квітень 1996 року: Версія 1.0 ЗК ухвалена ISO.

Було проведено ряд експериментальних оцінок на основі версії 1.0 ЗК, а також організовано широке обговорення документу.

Травень 1998 року: Оприлюднена версія 2.0 ЗК.

Червень 1999 року: На основі версії 2.0 ЗК прийнято міжнародний стандарт ISO/IEC 15408.

1 грудня 1999 року: Видано офіційний текст стандарту.

Зміни, що були внесені в стандарт на завершальній стадії його прийняття, враховані у версії 2.1 ЗК, що ідентична стандарту за змістом.

Вже після прийняття стандарту з урахуванням досвіду його використання з'явився ряд інтерпретацій ЗК, які після розгляду Комітетом з інтерпретацій (CCIMB) приймаються, офіційно оприлюднюються та набувають чинності як діючі зміни й доповнення до ЗК.

2005 рік: Версія 2.2 ЗК стала основою для внесення змін у стандарт ISO. Версію стандарту позначили як ISO/IEC 15408:2005. Ідентична їй за змістом версія ЗК – 2.3.

Протягом певного часу паралельно проводили урахування інтерпретацій (була завершена версія 2.4 ЗК і розроблювали версію 2.6 ЗК) і вели розробку версії 3.0 ЗК, яка має дещо змінену (спрощену) структуру вимог.

2008 рік: замість очікуваної версії 2.4 ЗК в якості стандарту ISO/IEC 15408:2008 ухвалили версію 3.0 ЗК.

Методологія застосування ЗК оформлена у вигляді окремого документа "Загальна методологія оцінювання безпеки інформаційних технологій". Також набула статусу стандарту, діюча версія: ISO/IEC 18045:2005

1.2.2. Основні відомості та структура загальних критеріїв

Стандарт ISO/IEC 15408 регламентує усі стадії розробки, кваліфікаційного аналізу та експлуатації продуктів інформаційних технологій, при цьому пропонує досить складну і бюрократичну концепцію процесу розробки і кваліфікаційного аналізу продуктів ІТ.

У застосуванні до оцінки безпеки продуктів інформаційних технологій (ПІТ) стандарт є по суті метазасобом, що задає систему понять термінів, в яких повинна проводитись оцінка.

Стандарт ISO/IEC 15408 містить відносно повний набір вимог безпеки (функціональних та гарантій), але не надає конкретних рішень щодо вимог та критеріїв для тих чи інших типів ПІТ, виконання яких необхідно перевіряти. Текст основних положень стандарту вимог та критеріїв формулюються у профілях захисту (ПЗ) та завданнях з безпеки (ЗБ). Саме офіційно прийняті профілі захисту утворюють побудовану на основі ЗК нормативну базу, що використовується на практиці у сфері інформаційної безпеки.

Загальні критерії є сукупністю самостійних, але взаємопов'язаних частин. Все це складається згідно змістом у структуру з ряду розділів:

1. «Представлення і загальна модель».
2. «Вимоги до функцій безпеки».
3. «Вимоги гарантій безпеки».
4. «Визначені профілі захисту».

Розділ «Представлення і загальна модель» – визначає загальну концепцію й принципи оцінки безпеки ІТ і подає загальну модель оцінки, а також конструкції для:

- формування задач захисту ІТ;
- вибору й визначення вимог безпеки ІТ;
- опису специфікацій високого рівня для продуктів і систем.

У розділі «Вимоги до функцій безпеки» – встановлюється набір функціональних компонентів, як стандартний шлях формулювання функціональних вимог до об'єктів оцінки.

Наступний розділ «Вимоги гарантій безпеки» – включає компоненти вимог гарантій оцінки, а також рівні гарантій оцінки, які визначають ранжирування за ступенем задоволення вимог.

Ще один розділ «Визначені профілі захисту» (був передбачений з самого початку, але після версії 1.0 ЗК був винесений за межі стандарту) – містить приклади профілів захисту, що включають функціональні вимоги безпеки та вимоги гарантій оцінки, що були ідентифіковані у вихідних критеріях (ITSEC, CTCPEC, FC, TCSEC), а також інші профілі.

1.2.3. Базові поняття

Серед базових понять, що визначені в стандарті ISO/IEC 15408 слід виділити наступні:

- Задачі захисту (Security Objectives).
- Профіль захисту (Protection Profile).
- Завдання з безпеки (Security Target).

Поняття ***задач захисту*** визначає потребу споживачів продукту ІТ:

- у протистоянні заданій множині загроз безпеці;
- у необхідності реалізації політики безпеки.

Профіль захисту – це спеціальний нормативний документ, що містить:

- задачі захисту;
- функціональні вимоги;
- вимоги адекватності;
- їхнє обґрунтування,

і служить керівництвом для розробника при створенні завдання з безпеки.

Завдання з безпеки – це спеціальний нормативний документ, що містить:

- задачі захисту;
- функціональні вимоги;

- вимоги адекватності;
- загальні специфікації засобів захисту;
- їхнє обґрунтування,

та у ході кваліфікаційного аналізу служить як опис продукту ІТ.

1.2.4. Процес розробки та кваліфікаційного аналізу

Матеріалами для проведення кваліфікаційного аналізу є: завдання з безпеки, що описує функції захисту продукту ІТ і вимоги безпеки, що відповідають вимогам профілю захисту, на реалізацію якого претендує продукт ІТ.

Докази можливостей продукту ІТ, представлені його розробником.

Згідно стандарту продукт ІТ проходить три стадії процесу кваліфікаційного аналізу.

Аналіз «Профілю захисту» на предмет:

- повноти;
- несуперечності;
- можливості реалізації;
- можливості використання як набору вимог для продукту, що аналізують.

Аналіз «Завдання з безпеки» на предмет:

- відповідності вимогам профілю захисту;
- повноти;
- несуперечності;
- можливості реалізації;
- можливості використання як опису продукту ІТ.

Загальну структуру процесу розробки та кваліфікаційного аналізу згідно зі стандартом ISO/IEC 15408 можна представити у виді схеми рис. 1.5.

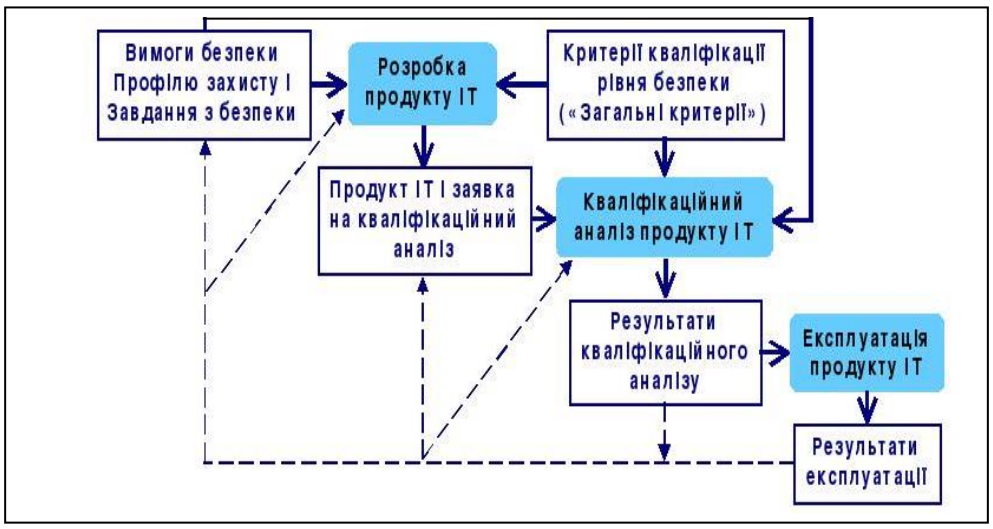


Рисунок 1.5 – Схема загальної структури процесу розробки та кваліфікаційного аналізу

Структуру профілю захисту складає:

1. Введення – інформація, необхідною для пошуку профілю в бібліотеці профілів:

- ідентифікатор – унікальне ім'я, придатне для пошуку серед подібних профілів і для посилань на нього;
- огляд змісту – коротка анотація профілю захисту, на підставі якої споживач може зробити висновок про придатність даного профілю для його потреб.

2. Опис ПІТ – коротка характеристика, функціональне призначення, принципи роботи, методи використання і т.д. Ця інформація не підлягає аналізу і сертифікації.

3. Середовище експлуатації – це опис усіх аспектів функціонування ПІТ, пов'язаних з безпекою:

- загрози безпеці – опис загроз безпеці, яким повинний протистояти захист. Для кожної загрози: джерело, метод впливу, об'єкт;
- політика безпеки – визначення і пояснення (при необхідності) правил політики безпеки;
- умови експлуатації – вичерпна характеристика середовища експлуатації з погляду безпеки.

4. Задачі захисту – це потреби користувачів у протидії зазначеним загрозам безпеці та/або в реалізації політики безпеки:

- задачі захисту ПІТ;

- інші задачі захисту.

5. Вимоги безпеки – це вимоги безпеки, яким повинний задовольняти ПІТ для рішення задач захисту:

- функціональні вимоги – тільки типові вимоги, передбачені відповідними розділами «Загальних критеріїв». Можуть наказувати чи забороняти використання конкретних методів і засобів;
- вимоги гарантій – також тільки типові вимоги;
- вимоги до середовища експлуатації – необов'язковий розділ.

Функціональні вимоги та/або вимоги гарантій до середовища експлуатації. Використання типових вимог є бажаним, але не обов'язковим.

6. Додаткові відомості.

7. Обґрунтування – це демонстрація того, що профіль захисту містить повну і зв'язну множину вимог, і що ПІТ, який їм задовольняє, буде ефективно протистояти загрозам безпеці середовища експлуатації:

- обґрунтування задач захисту – демонстрація того, що задачі захисту, запропоновані в профілі, відповідають властивостям середовища експлуатації;
- обґрунтування вимог безпеки – демонстрація того, що вимоги безпеки дозволяють вирішити задачі захисту.

Для обґрунтування вимог безпеки демонструють, що:

- сукупність цілей, переслідуваних окремими функціональними вимогами, відповідає встановленим задачам захисту;
- вимоги безпеки є погодженими (не суперечать одна одній);
- усі взаємозв'язки між вимогами враховані або за допомогою їхньої вказівки у вимогах, або за допомогою встановлення вимог до середовища експлуатації;
- обраний набір вимог і рівень гарантій можуть бути обґрунтовані.

Структуру «Завдання з безпеки» складає:

1. Введення – це інформація, необхідна для ідентифікації Завдання з безпеки, визначення призначення, а також огляд його змісту:

- ідентифікатор – унікальне ім'я, необхідне для пошуку й ідентифікації;
- завдання з безпеки і відповідного йому ПІТ;
- огляд змісту – докладна анотація Завдання з безпеки, на підставі якої споживач може зробити висновок про придатність ПІТ для рішення його задач;
- заявка на відповідність ЗК – опис усіх властивостей ПІТ, що підлягають кваліфікаційному аналізу на основі ЗК.

2. Опис ПІТ:

- середовище експлуатації;
- загрози безпеці;
- політика безпеки;
- умови експлуатації.

3. Задачі захисту:
- задачі захисту ПІТ;
- інші задачі захисту.

Вищезазначені розділи збігаються з однойменними розділами профілю захисту.

4. Вимоги безпеки – це вимоги, якими керувався розроблювач ПІТ, що дозволяє йому заявляти про успішне рішення задач захисту:
- функціональні вимоги – вимоги, які на відміну від відповідного розділу профілю захисту, допускають використання крім типових вимог ЗК інших, специфічних для даного продукту і середовища його експлуатації вимог;
- вимоги гарантій можуть включати рівні гарантій, не передбачені в ЗК;
- вимоги до середовища експлуатації – необов'язковий розділ.

5. Загальні специфікації ПІТ – це відображення реалізації ПІТ вимог безпеки за допомогою визначення високорівневих специфікацій функцій захисту:
- специфікації функцій захисту – опис функціональних можливостей засобів захисту ПІТ, що заявлені розроблювачем як ті, що реалізують вимоги безпеки. Форма представлення специфікацій повинна дозволяти визначати відповідності між функціями захисту і вимогами безпеки;
- специфікації рівня гарантій – визначення заявленого рівня гарантій захисту ПІТ і його відповідність вимогам гарантій у вигляді подання параметрів технології проектування і створення ПІТ;

6. Заявка на відповідність профілю захисту – необов'язковий пункт. Завдання з безпеки претендує на задоволення вимог одного чи декількох профілів захисту, для кожного з яких цей розділ повинний містити таку інформацію:
- посилання на профіль захисту – однозначно ідентифікує профіль захисту, на реалізацію якого претендує Завдання з безпеки. Реалізація профілю захисту передбачає коректну реалізацію всіх його вимог без винятку;
- відповідність профілю захисту – можливості ПІТ, що реалізують задачі захисту і вимоги, що містяться в профілі захисту;
- удосконалення профілю захисту – можливості ПІТ, що виходять за рамки профілю.

6. Обґрунтування – це демонстрація того, що Завдання з безпеки містить повну і зв'язну множину вимог, що ПІТ, який його реалізує, буде ефективно протистояти загрозам безпеці середовища експлуатації, і що загальні специфікації функцій захисту відповідають вимогам безпеки:

- обґрунтування задач захисту – демонстрація того, що задачі захисту, запропоновані в Завданні з безпеки, відповідають властивостям середовища експлуатації;
- обґрунтування вимог безпеки – демонстрація того, що вимоги безпеки дозволяють вирішити задачі захисту. Необхідно показати, що:

– функціональні вимоги безпеки відповідають задачам захисту;

– вимоги гарантій відповідають функціональним вимогам і підсилюють їх;

– сукупність усіх функціональних вимог забезпечує рішення задач захисту;

– усі взаємозв'язки між вимогами ЗК враховані або за допомогою зазначення їх у вимогах, або за допомогою встановлення вимог до середовища експлуатації;

– усі вимоги безпеки успішно реалізовані;

– заявлений рівень гарантій може бути підтверджений;

- обґрунтування функцій захисту – демонстрація того, що функції захисту відповідають функціональним вимогам безпеки і задачам захисту, при цьому позначимо, що:

– зазначені функції захисту відповідають заявленим задачам захисту;

– сукупність зазначених функцій захисту забезпечує ефективне рішення сукупності задач захисту;

– заявлені можливості функцій захисту відповідають дійсності.

- обґрунтування рівня гарантій – підтвердження, що заявлений рівень безпеки відповідає вимогам гарантій;
- обґрунтування відповідності профілю захисту – демонстрація того, що вимоги Завдання з безпеки підтримують усі вимоги профілю захисту.

Повинно бути показано, що:

1. Усі удосконалення задач захисту у порівнянні з профілем захисту здійснені коректно й у напрямку їхнього розвитку і конкретизації.
2. Усі задачі захисту профілю успішно вирішені і усі вимоги профілю захисту задоволені.
3. Ніякі додатково введені Завдання з безпеки, спеціальні задачі захисту і вимоги безпеки не суперечать профілю захисту.

Структура вимог містить:

Функціональні вимоги та вимоги гарантій подані в одному спільному стилі, які використовують одну і ту саму організацію та термінологію.

Термін клас використовується для найбільш загального групування вимог безпеки. Усі члени класу поділяють спільний намір при різниці в охопленні цілей безпеки.

Члени класу названі сімействами. Сімейство – це групування наборів вимог безпеки, які забезпечують виконання певної частини цілей безпеки, але можуть відрізнятись в акценті або жорсткості.

Члени сімейства названі компонентами. Компонент описує визначений набір вимог безпеки – найменший набір вимог безпеки, що обирається для включення у структури, визначені в ЗК.

Компоненти побудовані з елементів. Елемент – це найнижчий і неподільний рівень вимог безпеки, на якому проводиться оцінка їх виконання.

Перетворення компонентів. Компоненти можуть бути перетворені за допомогою дозволених дій, для забезпечення виконання певної політики безпеки або протистояння певній загрозі. Не усі дії припустимі на усіх компонентах. Кожний компонент ідентифікує і визначає дозволені дії або обставини, за якими дія може застосовуватись до компонента, а також результати застосування дії.

До дозволених дій належать: призначення, вибір і обробка

Призначення дозволяє заповнити специфікацію ідентифікованого параметра при використанні компонента. Параметр може бути ознакою або правилом, що конкретизує вимоги до певної величини або діапазону величин.

Вибір – це дія вибору одного чи більшої кількості пунктів із списку, щоби конкретизувати можливості елементу

Обробка дозволяє включити додаткові деталі в елемент, і передбачає інтерпретацію вимоги, правила, константи або умови, засновану на задачах захисту. Обробка повинна лише обмежувати набір можливих функцій або механізмів, для здійснення вимог, але не збільшувати їх. Обробка не дозволяє створювати нові вимоги або видаляти існуючі, і не впливає на список залежності, що пов’язані з компонентом.

Набори структур. ЗК визначають набори структур, що поєднують компоненти вимог безпеки

Проміжна комбінація компонентів називається пакетом. Він включає набір вимог, які забезпечують виконання піднабору задач захисту. Пакет призначений для багаторазового використання, визначає вимоги, які є необхідними для досягнення ідентифікованих задач. Пакет може використовуватись для формування Профілів захисту і Завдань з безпеки.

Рівні гарантій оцінки – це визначені пакети вимог гарантій. Рівень гарантій – це набір базових вимог гарантій для оцінки. (EAL).

1.2.5. Таксономія вимог. Функціональні вимоги, вимоги гарантій

До таксономії вимог відносяться функціональні вимоги та вимоги гарантій.

Функціональними вимогами є :

- FAU – Аудит безпеки – вимоги до розпізнання, реєстрації, зберігання та аналізу інформації, яка пов'язана з діями, що стосуються безпеки об'єкта оцінювання (ОО).
- FCO – Інформаційний обмін – вимоги до визначення ідентичності сторін, що беруть участь в обміні даними.
- FCS – Криптографічна підтримка.
- FDP – Захист інформації користувача – вимоги до функцій безпеки ОО та політики функцій безпеки ОО, що пов'язані із захистом даних користувача.
- FIA – Ідентифікація й автентифікація.

Вимоги гарантій або класи – це вимоги до функцій, що призначені для встановлення й перевірки ідентичності користувача.

Такими функціями є:

- FMT – Керування безпекою – вимоги, що пов'язані з керуванням безпекою ОО.
- FPR – Конфіденційність доступу до системи – вимоги таємності, що забезпечують захист користувача від розкриття й невірного використання його ідентифікаторів іншими користувачами.
- FPT – Захист функцій безпеки – вимоги, що стосуються цілісності та контролю механізмів, що забезпечують функції безпеки, та цілісності й контролю даних функцій безпеки.
- FRU – Контроль за використанням ресурсів.
- FTA – Контроль доступу до системи – функціональні вимоги, понад вимог ідентифікації й автентифікації, для керування сеансом роботи користувача.
- FTP – Забезпечення прямої взаємодії – вимоги до забезпечення надійного маршруту зв'язку між користувачами і функціями безпеки та надійного каналу зв'язку між функціями безпеки, що мають такі спільні характеристики:

– маршрут комунікацій, побудований із застосуванням внутрішніх і зовнішніх каналів комунікацій, що ізолюють ідентифікований піднабір даних та

команд функцій безпеки від інших частин функцій безпеки та даних користувача;

– використання маршруту комунікацій може бути ініційовано користувачем та/або функцією безпеки;

– маршрут комунікацій здатний забезпечити гарантії того, що користувач взаємодіє з потрібною функцією безпеки і що функція безпеки взаємодіє з потрібним користувачем, тобто забезпечується надійна ідентифікація кінцевих пунктів.

- APE – Оцінка профілю захисту – вимоги до оцінки профілю захисту з метою підтвердження того, що він є повним, несуперечливим, технічно вірним, і таким чином придатним для розробки завдань з безпеки і для занесення в реєстр.
- ASE – Оцінка завдання з безпеки – вимоги до оцінки завдання з безпеки з метою підтвердження того, що воно є повним, несуперечливим, технічно вірним, і таким чином придатним для використання в якості основи для оцінки відповідного ОО.
- ADV – Розробка – вимоги до процесу розробки, що дозволяють перевірити, чи були фактично відпрацьовані функції безпеки.
- AGD – Документація – вимоги до зрозумілості, повноти й завершеності експлуатаційної документації.
- ALC – Підтримка життєвого циклу – вимоги до прийняття добре визначеної моделі етапів життєвого циклу ОО.
- ATE – Тестування – вимоги до об’єму, глибини та виду тестування ОО.
- AVA – Оцінка вразливості – вимоги, що спрямовані на виявлення вразливих місць.
- ACO – Інтеграція – вимоги, що надають впевненості у тому, що інтегрований ОО буде функціонувати безпечно, якщо він спирається на функції захисту раніше оцінених компонентів.

1.3. Послуги і механізми захисту інформації

Перш ніж визначати засоби захисту інформації і будувати відповідну систему, необхідно проаналізувати основні послуги й механізми захисту інформації. Для цього розглянемо еталонну модель взаємодії відкритих систем (OSI) і класифікацію можливих атак на кожний з рівнів цієї моделі.

Еталонну модель OSI (англ. Open Systems Interconnection) було розроблену інститутом стандартизації ISO з метою розмежування функцій різних протоколів у процесі передачі інформації від одного абонента іншому. Подібних класів функцій було виділено 7. Вони одержали назву рівнів. Кожен

рівень виконує певні завдання в процесі передачі блоку інформації, причому відповідний рівень на приймальній стороні проводить перетворення, зворотні тим, які проводив той же рівень на передавальній стороні. В цілому проходження блоку даних від відправника до одержувача показане на рис. 1.6. Кожен рівень додає до пакета невеликий обсяг своєї службової інформації – префікс (на рисунку вони зображені як P1...P7). Деякі рівні в конкретній реалізації цілком можуть бути відсутніми. Дана модель дозволяє провести класифікацію мережевих атак відповідно до рівня їх дії.

Фізичний рівень відповідає за перетворення електронних сигналів у сигнали середовища передачі інформації (імпульси напруги, радіохвилі, інфрачервоні сигнали). На цьому рівні основним класом атак є "відмова в сервісі". Постановка шумів по всій смузі пропускання каналу може призвести до розриву зв'язку.

Канальний рівень керує синхронізацією двох і більшої кількості мережевих адаптерів, підключених до єдиного середовища передачі даних. Прикладом його є протокол Ethernet. Дії на цьому рівні також полягають в основному в атаці "відмова в сервісі". Проте, на відміну від попереднього рівня, тут проводиться перебій синхропосилань або самої передачі даних періодичною передачею "без дозволу або передачею не в свій час". Мережевий рівень відповідає за систему унікальних імен і доставку пакетів за цим іменем, тобто за маршрутизацію пакетів. Відповідно й атаки на цьому рівні найчастіше спрямовані на конфіденційність і цілісність службової інформації, пов'язаної з адресацією і наявністю унікальних імен.

Транспортний рівень відповідає за доставку великих повідомлень по лініях з комутацією пакетів. Оскільки в подібних лініях розмір пакета є зазвичай невеликим числом (від 500 байт до 5 кілобайт), то для передачі великих обсягів інформації їх необхідно розбивати на передавальній стороні й збирати на приймальній. Вся річ у тому, що пакети на приймальну сторону можуть приходити й іноді приходять не в тому порядку, в якому вони були відправлені. Причина зазвичай полягає у втраті деяких пакетів через помилки або переповненість каналів, рідше – у використанні для передачі потоку двох альтернативних шляхів у мережі.

Отже, операційна система повинна зберігати деякий буфер пакетів, чекаючи приходу тих, що затрималися в процесі передачі. А якщо зловмисник з наміром формує пакети так, щоб послідовність була великою і свідомо неповною, то тут можна чекати як постійної зайнятості буфера, так і небезпечних помилок через його переповнення.

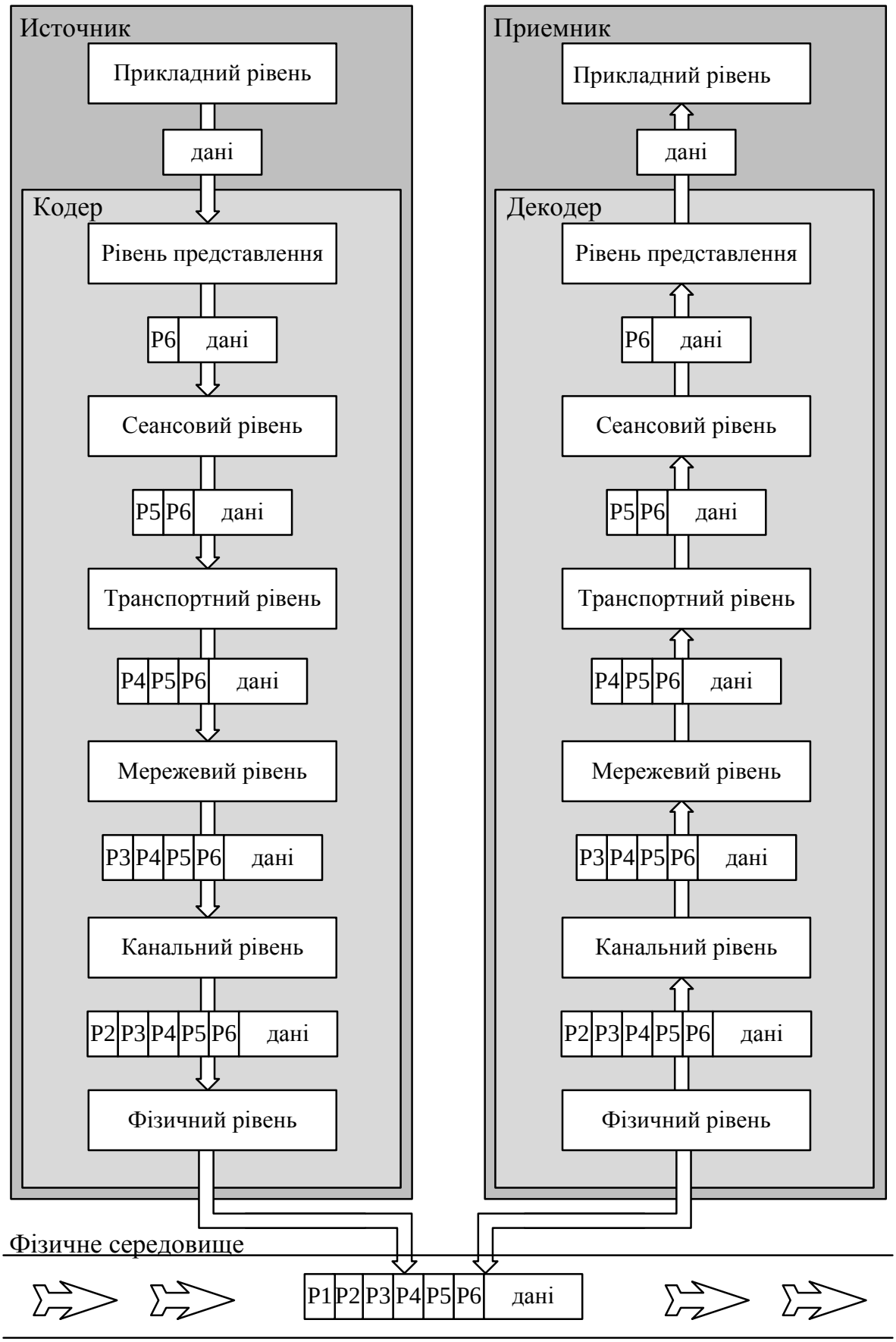


Рисунок 1.6 – Проходження блоку даних від відправника до одержувача відповідно до моделі взаємодії відкритих систем

Сеансовий рівень відповідає за процедуру встановлення початку сеансу і підтвердження (квитування) приходу кожного пакета від відправника одержувачу. У мережі Інтернет протоколом сеансового рівня є протокол TCP (він займає і 4-й, і 5-й рівні моделі OSI). Відносно сеансового рівня дуже поширена специфічна атака класу "відмова в сервісі", основана на властивостях процедури встановлення з'єднання в протоколі TCP. Вона одержала назву SYN-Flood (flood - англ. "великий потік").

Представницький рівень визначає формат, використаний для обміну даними між мережевими комп'ютерами і відповідає за перетворення протоколів, трансляцію даних, їх шифрування, зміну або перетворення вжитого набору символів (кодової таблиці) і розширення графічних команд. Представницький рівень, крім того, керує стисненням даних для зменшення переданих бітів. У зв'язку з цим особливо небезпечними є атаки, що спрямовані на спотворення даних (порушення цілісності), які часто призводять до зупинки в роботі окремих вузлів у комп'ютерних мережах.

Прикладний рівень – найвищій рівень моделі OSI. Він є вікном для доступу прикладних процесів до мережевих послуг. Цей рівень забезпечує послуги, що безпосередньо підтримують додатки користувача, такі, як програмне забезпечення для передачі файлів, доступу до баз даних, електронна пошта. Прикладний рівень керує загальним доступом до мережі і обробкою помилок. Широко використовуваним розподіленим додатком на цьому рівні є електронна пошта. У зв'язку з цим спостерігається інтерес до засобів забезпечення автентифікації і конфіденційності оброблюваних даних.

Методологічною основою розробки системи захисту інформації є стандарт ISO/IEC 15408, згідно з яким основними нормативними документами, що характеризують інформаційну систему з точки зору безпеки, є профіль захисту (protection profile) і проект забезпечення безпеки (security target). Під профілем захисту розуміють незалежну множину функціональних вимог безпеки і вимог адекватності, спрямованих на задоволення потреб споживача. Проектом безпеки є безліч вимог безпеки й специфікацій функцій безпеки.

Відповідно до основних положень міжнародних стандартів життєвий цикл системи захисту інформації складається з п'яти етапів:

1. Визначення політики безпеки, яка містить абстрактний ряд вимог до безпеки системи.

2. Аналіз вимог безпеки, включаючи аналіз ризиків, аналіз урядових, правових і стандартних вимог.

3. Визначення послуг безпеки, необхідних для задоволення поставлених вимог.

4. Побудова і впровадження системи безпеки, включаючи вибір механізмів безпеки, що забезпечують конкретні вибрані послуги безпеки.

5. Безперервне управління безпекою.

Послуга безпеки призначена для забезпечення захисту від ідентифікованої загрози. Існують абстрактні поняття, які можуть бути використані для характеристики вимог безпеки. Механізмом безпеки є засіб, за допомогою якого реалізується і застосовується відповідна послуга.

Стандарти ISO 7498, ISO/IEC 10181 визначають п'ять базових загальноприйнятих послуг безпеки (рис. 1.7):

- автентифікація (authentication);
- керування доступом (access control);
- конфіденційність даних (data confidentiality);
- цілісність даних (data integrity);
- невідмова (причетність) (non-repudiation).

Додатково в ISO/IEC 10181-7 розглядається перевірка безпеки (security audit).

Автентифікація – послуга, що гарантує надійність ідентифікації джерела повідомлень або електронного документа, а також оригінальність джерела.

Керування доступом – послуга, що забезпечує доступ до ресурсів лише авторизованих користувачів (процесів), що гарантує відповідні права доступу для авторизованих користувачів і запобігає неавторизованому доступу як “внутрішніх”, так і ”зовнішніх” користувачів. Ця послуга застосовується до різних типів доступу до ресурсів, наприклад, використання комунікаційних ресурсів, читання, запис або видалення інформаційних ресурсів, використання ресурсів обчислювальних систем з обробки даних і та ін. й використовується для встановлення політики управління/обмеження доступу.

Під **конфіденційністю** розуміють властивість системи, яка гарантує, що інформація не може бути доступна або розкрита для неавторизованих (неуповноважених) осіб, об'єктів або процесів.

Цілісність даних – послуга, що гарантує можливість модифікації тієї інформації, яка міститься в комп'ютерній системі і зв'язку, й пересилається по каналах, тільки суб'єктами, які мають на це право.

Під **причетністю** розуміють здатність запобігання можливості відмови одним з реальних учасників комунікацій від факту його повної або часткової участі в передачі даних.

Доступність визначається як додаткова послуга забезпечення захищеності інформаційних систем. Механізми забезпечення доступності запобігають атакам, що мають на меті унеможливити доступ до ресурсів або послуг інформаційної системи (або зробити їх “якість” незадовільною) для користувача. На рис. 1.8 подано розподіл послуг безпеки по рівнях моделі взаємодії відкритих систем (ВВС). Як видно з наведеного рисунка, велика частина послуг безпеки доводиться на верхні рівні моделі ВВС, переважно, на

рівень прикладного процесу. Розглянемо основні системи і протоколи, що забезпечують захист інформації на різних рівнях моделі ВВС.

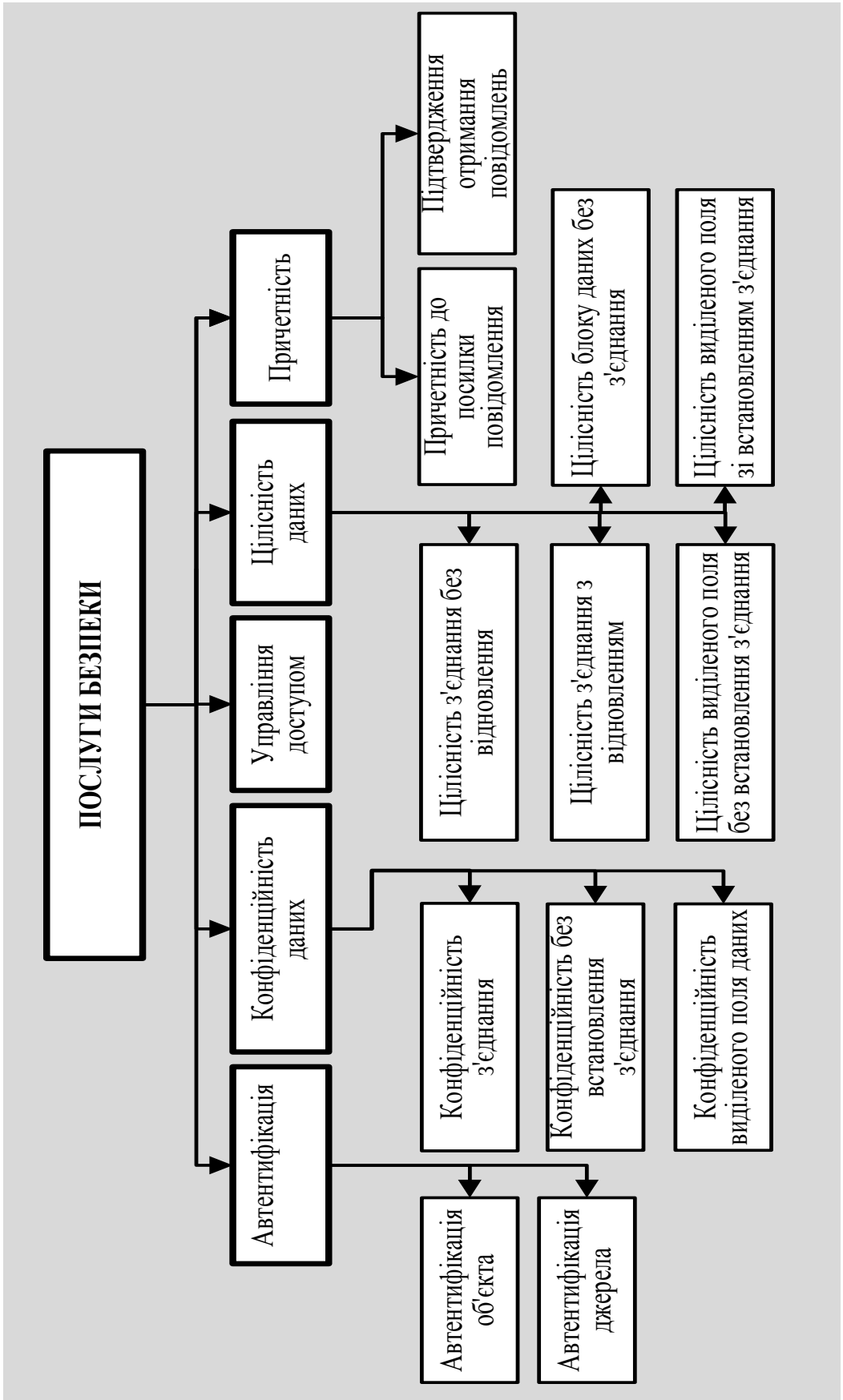


Рисунок 1.7 – Загальна класифікація послуг безпеки

На прикладному рівні в теперішній час для забезпечення захисту інформації найчастіше використовуються дві системи (рис. 1.9) – PGP і S/MIME.

PGP – широко розповсюджена система захисту, незалежна від будь-якої організації або органу влади. Тому вона підходить як для індивідуального користування, так і для включення в конфігурацію мережі будь-якої організації. S/MIME є системою захисту, спеціально розробленою як стандарт мережі Інтернет.

Зростання популярності Word Wide Web для електронної комерції і розповсюдження інформації привело до виникнення гострої необхідності в забезпеченні захисту відповідних даних у Web. Забезпечити виконання ряду послуг дозволяє застосування протоколів SSL/TLS і SET (рис. 1.10). Протокол призначений для забезпечення надійного захисту наскрізної передачі даних з використанням протоколу TCP. Протокол SET – це відкриті специфікації шифрування і захисту, розроблені з метою захисту транзакцій, що виконуються в мережі Інтернет за допомогою пластикових платіжних карток.

Незважаючи на розробку цілого ряду механізмів захисту на прикладному і сеансовому рівні, існує необхідність забезпечення безпеки на мережевому рівні. Наприклад, підприємство може захистити свою мережу TCP/IP за допомогою заборони доступу до ненадійних вузлів, шифруючи пакети даних, що передаються з мережі підприємства, і вимагаючи автентифікації пакетів, що входять у цю мережу із зовні. За допомогою реалізації захисту на мережевому рівні організація може забезпечити роботу в мережі не тільки додатків, які мають свої засоби захисту, але і додатків, що не володіють такими засобами. Захист на рівні IP (мережевому) (рис. 1.11) охоплює 3 сфери безпеки: автентифікацію, конфіденційність, керування ключами.

Механізми безпеки є конкретними заходами для реалізації послуг безпеки (рис. 1.12). Взаємозв'язок послуг і механізмів безпеки подано на рис. 1.13.

Стандарт поділяє механізми безпеки на два класи, а саме: спеціальні механізми забезпечення безпеки, які використовуються для реалізації специфічних послуг і різняться для різних послуг, та загальні механізми, які не належать до конкретних послуг безпеки.

До **спеціальних механізмів** забезпечення безпеки належать такі:

- шифрування (encipherment);
- механізми цифрового підпису (digital signature mechanisms);
- механізми управління доступом (access control mechanisms);
- механізми забезпечення захисту цілісності даних (data integrity mechanisms), які включають криптографічні контрольні функції;
- механізми автентифікації (authentication exchange mechanisms);
- механізми заповнення трафіку (padding traffic mechanisms);

- механізми керування маршрутизацією (routing control mechanisms).
- Розглянемо кожен тип механізмів детальніше.

Механізми шифрування припускають використання криптографічних перетворень даних для того, щоб зробити їх нечитабельними або неосмисленими. Шифрування застосовується спільно зі зворотною функцією – розшифрування.

Шифрування використовується для забезпечення послуги конфіденційності, але може також підтримувати інші послуги забезпечення безпеки, наприклад, автентифікації і захисту цілісності даних.

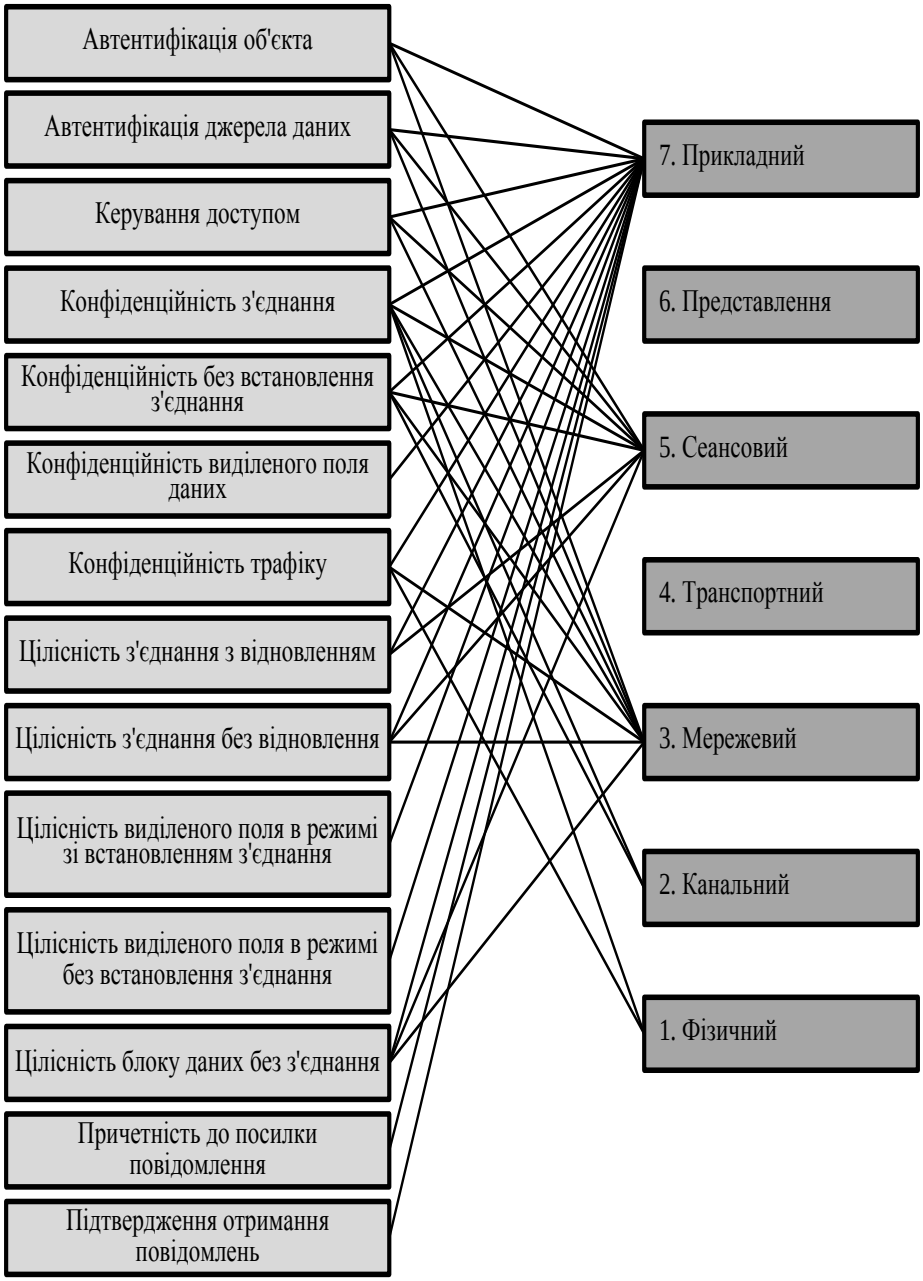


Рисунок 1.8 – Розподіл послуг безпеки по рівнях еталонної моделі ВВС

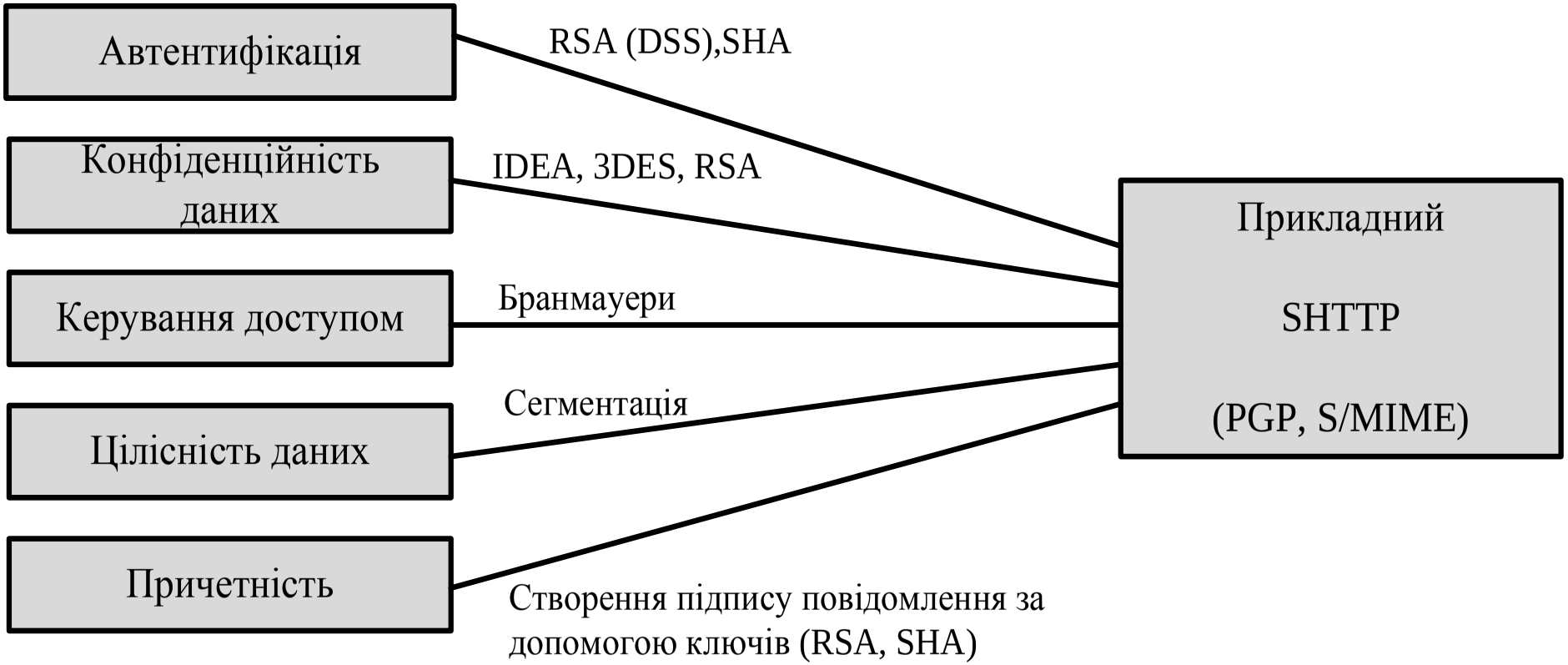


Рисунок 1.9 – Системи, що забезпечують захист інформації на прикладному рівні

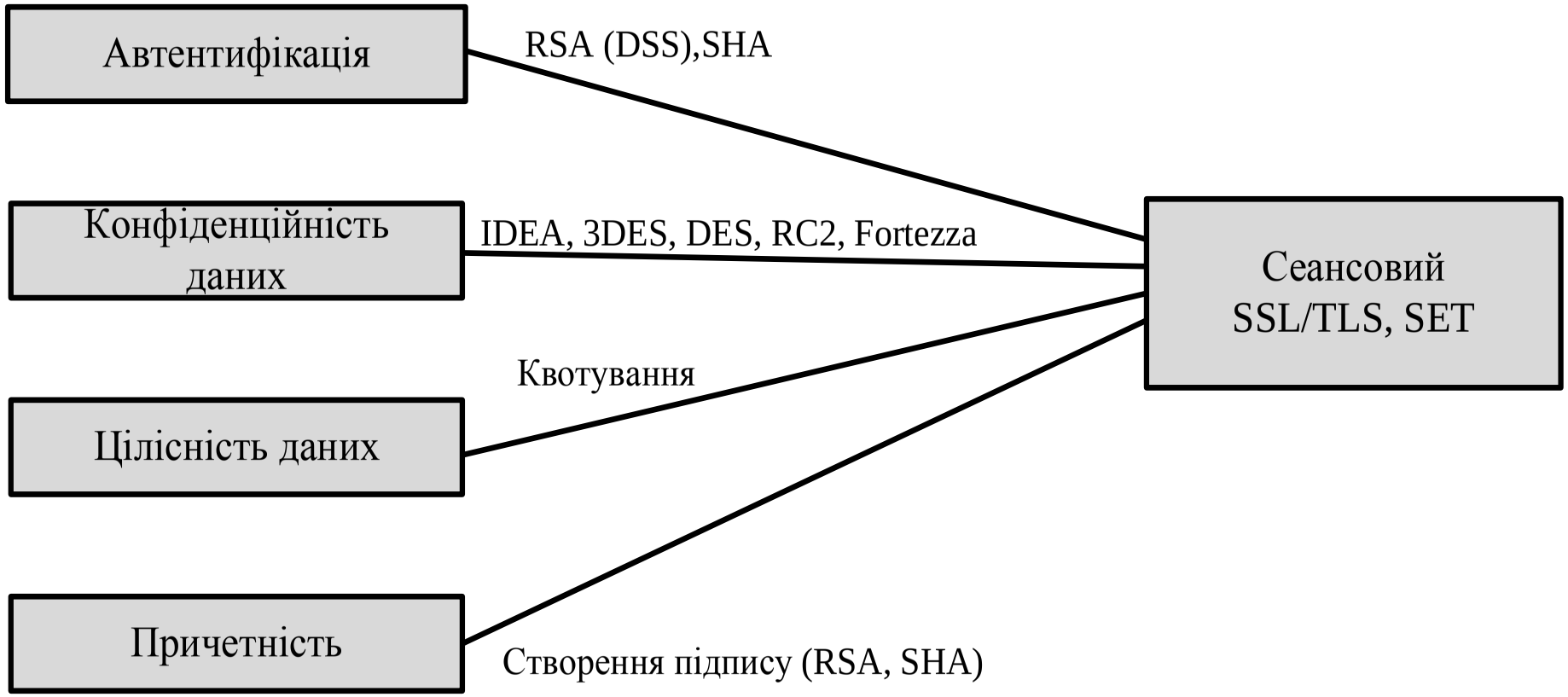


Рисунок 1.10 – Системи, що забезпечують захист інформації на сеансовому рівні

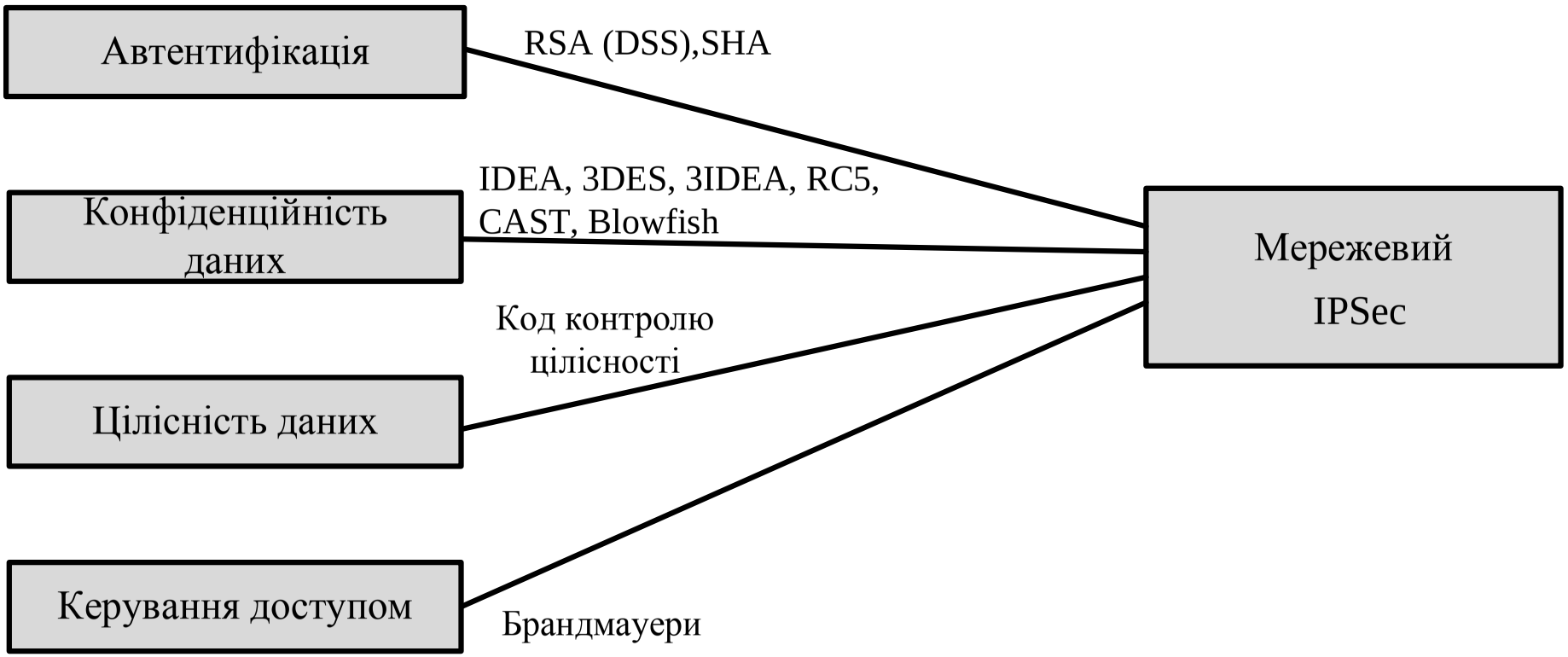


Рисунок 1.11 – Системи, що забезпечують захист інформації на мережевому рівні

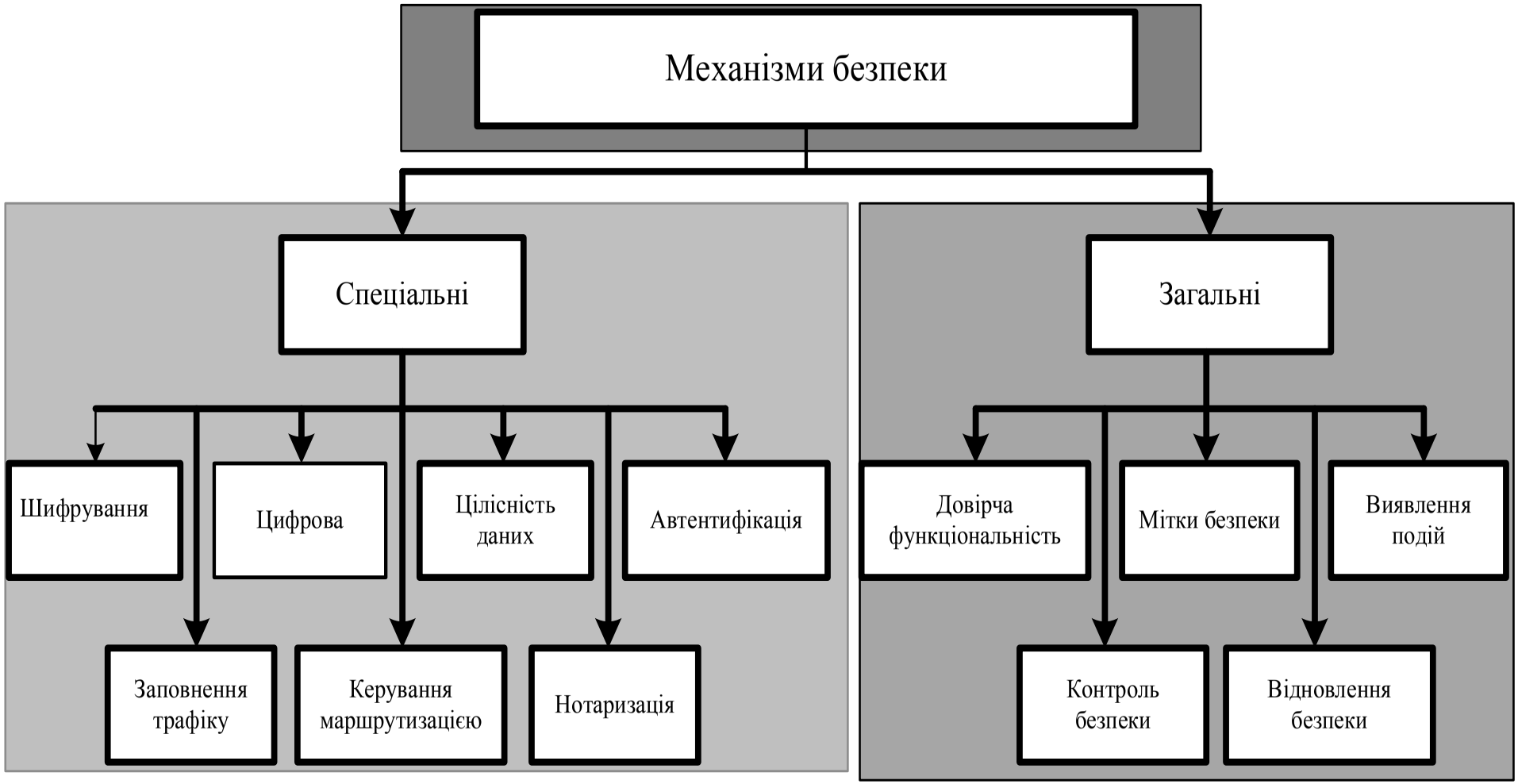


Рисунок 1.12 – Загальна характеристика механізмів

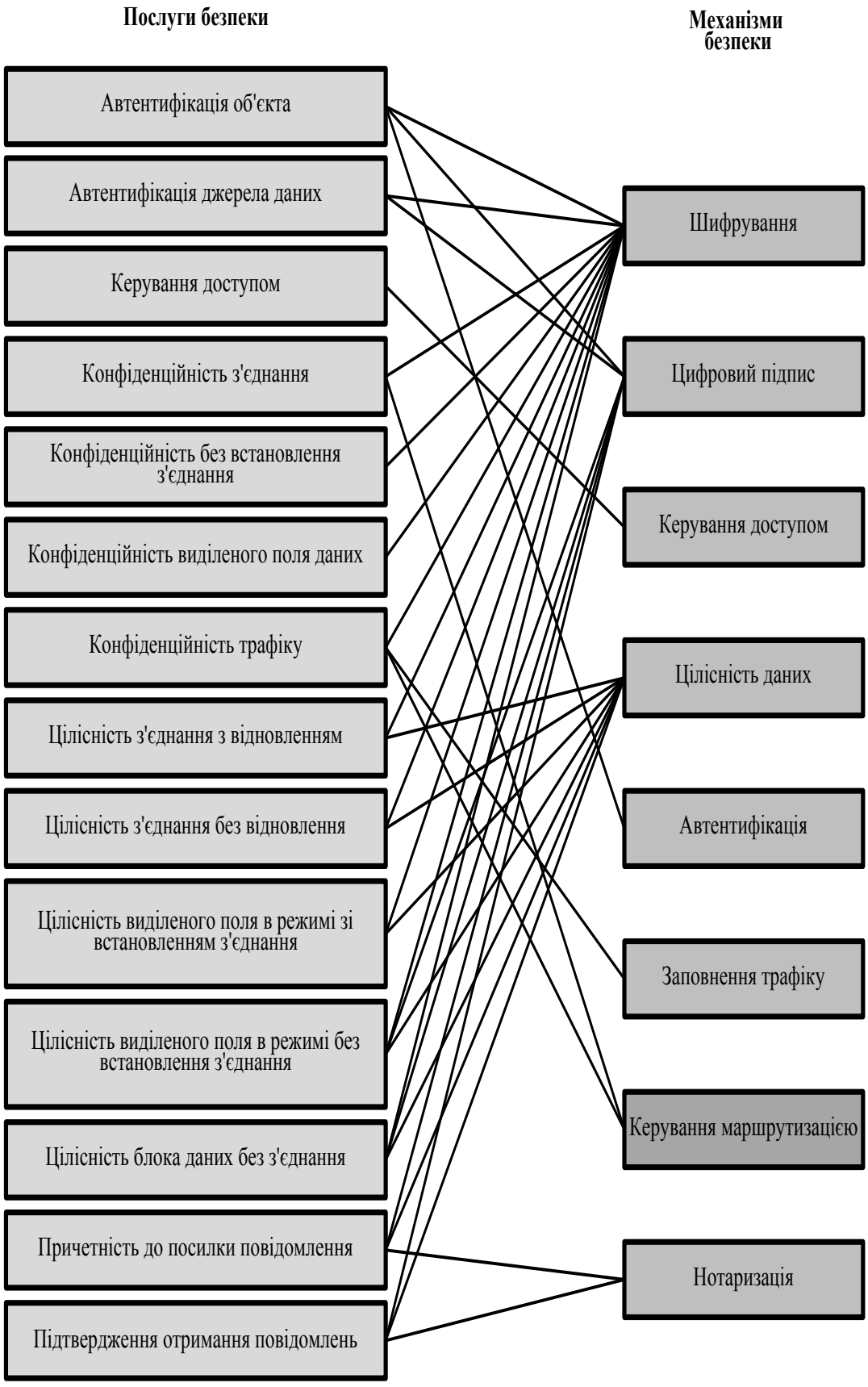


Рисунок 1.13 – Взаємозв'язок послуг і механізмів безпеки

Цифровий підпис є цифровим еквівалентом підпису (друку, штампа та ін..), наявність якого в повідомленні дозволяє з високою точністю визначити джерело повідомлення (документа) і юридично довести, що, з певною ймовірністю, тільки він міг створити і підписати цей документ. Механізми цифрового підпису використовують "відкриті" ключі, які генеруються відправником даних і перевіряються одержувачем. Для шифрування контрольної суми підписуваного повідомлення можуть бути використані методи несиметричного шифрування. Цифровий підпис використовується для забезпечення послуг автентифікації і захисту цілісності, для яких суб'єкт верифікації підписаних даних наперед невідомий.

При певному виборі контрольованого параметра цифровий підпис також може застосовуватися в реалізації послуги підтвердження причетності.

Механізми керування доступом використовуються для забезпечення послуг керування доступом і реалізують політику керування доступом. При ухваленні рішень про надання запрошуваного типу доступу можуть використовуватися такі види й джерела інформації:

- бази даних керування доступом, в яких можуть знаходитися списки керування доступом або структури аналогічного призначення;
- паролі або інша ідентифікаційна інформація;
- ідентифікаційні документи або інші посвідчення, пред'явлення яких свідчить про наявність прав доступу;
- позначки безпеки, асоційовані з суб'єктами й об'єктами доступу;
- час запрошуваного доступу;
- маршрут запрошуваного доступу;
- тривалість запитуваного доступу й інша інформація.
- механізми нотаризації (notarisation mechanisms).

Механізми цілісності даних діляться на два типи механізмів:

- механізми захисту цілісності окремого пакета даних;
- механізми захисту цілісності послідовності пакетів даних.

Механізми другого типу, які зазвичай застосовуються спільно з механізмами захисту цілісності окремого пакета даних, можуть використовуватися для забезпечення послуг цілісності при організації зв'язку в режимі зі встановленням з'єднання. Тут використовуються такі прийоми, як нумерація пакетів, часові штампи, криптографічне скріплення. Ці механізми дозволяють забезпечити захист від крадіжки, переупорядковування, дублювання і вставки повідомлень. У мережах, що функціонують в режимі без встановлення з'єднання, використання тимчасових штампів забезпечує також і обмежену форму захисту від дублювання.

У загальному випадку під автентифікацією розуміють встановлення достовірності повідомлення, джерела даних і приймача даних.

Автентифікація джерела даних часто забезпечується шляхом використання механізму захисту цілісності даних спільно з шифруванням або цифрового підпису. Логічна автентифікація користувача комп'ютерної системи здійснюється на основі пароля.

Автентифікація об'єкта комунікації зазвичай виконується за допомогою подвійного або потрійного підтвердження з'єднання або “рукостискання” аналогічно процедурі синхронізації пакетів у протоколах зі встановленням з'єднання. Односторонній (одноразовий) обмін забезпечує тільки одноразову автентифікацію і не може гарантувати своєчасність обміну. Двосторонній (двократний) обмін забезпечує взаємну автентифікацію джерела і приймача, але не забезпечує своєчасності обміну без застосування спеціальних засобів синхронізації. Тристоронній обмін дозволяє досягти повної взаємної автентифікації систем без додаткової синхронізації. Для забезпечення автентифікації також можуть використовуватися спеціальні механізми керування криптографічними ключами.

Механізм заповнення трафіку застосовується для забезпечення конфіденційності трафіку. Заповнення трафіку може включати генерацію випадкового трафіку, заповнення додатковою інформацією інформативних пакетів, передачу пакетів через проміжні станції в "непотрібному" напрямі. Обидва типи пакетів, як інформативний, так і випадковий, можуть доповнюватися до постійної довжини.

Механізми нотаризації привертають третю сторону, що користується довірою двох суб'єктів, для забезпечення підтвердження комунікаційних характеристик переданих даних. Такими комунікаційними характеристиками є цілісність, час, особи відправників і одержувачів. Найчастіше механізми нотаризації застосовуються для забезпечення послуги підтвердження причетності. Для підтвердження причетності відправника даних нотаризація застосовується спільно з цифровим підписом на основі "відкритого" ключа.

Нотаризація може також застосовуватися для забезпечення надійної тимчасової позначки, що забезпечується "тимчасовим нотаріусом". Така позначка може містити підпис "нотаріуса", ідентифікатор повідомлення, імена відправника і одержувача, а також зареєстровані час і дату отримання повідомлення. При цьому "нотаріус" не має доступу до самого повідомлення, що забезпечує конфіденційність повідомлення.

Механізми керування маршрутизацією застосовуються для забезпечення конфіденційності з метою запобігання контролю за шляхом проходження даних від відправника до одержувача. Вибір шляху може здійснюватися або крайовою системою, реалізуючи маршрутизацію, яка

визначається джерелом (sourse routing), або проміжною системою на основі використання "позначок безпеки", що вводяться в пакет крайовою системою.

Цей механізм вимагає забезпечення надійності (конфіденційності) проміжних систем і може мати істотні варіації при використанні різних систем. Він може також використовуватися і для забезпечення захисту цілісності даних (з функціями відновлення даних або з'єднання) за рахунок введення і використання для передачі даних альтернативних шляхів у разі виникнення атак, що призводять до переривання комунікацій.

До загальних механізмів забезпечення безпеки належать:

- довірча функціональність (trusted functionality);
- позначки безпеки (security lables);
- виявлення подій (event detection)
- контроль безпеки (security audit trail);
- відновлення безпеки (security recovery).

Довірча функціональність використовується разом з іншими механізмами безпеки і є сукупністю рекомендацій і способів, які повинні реалізовуватися для забезпечення гарантії правильної і надійної роботи інших механізмів безпеки. Довірча функціональність припускає широке використання нормативної документації при розробленні програмних або апаратних засобів, що реалізовують механізми безпеки. Розроблення цих засобів повинно вестися при дотриманні відповідних організаційних вимог. Програмні і апаратні засоби повинні розроблятися, тестуватися і сертифікуватися на основі єдиних методик. Саме тут забезпечуються всі необхідні вимоги і рекомендації щодо електромагнітних випромінювань, можливостей фізичного втручання, використання безпечних каналів розповсюдження і багато іншого. Будь-які дані (записані дані, обчислювальні потужності та/або комунікаційні послуги) можуть мати асоційовані **позначки безпеки**, які визначають їх рівень секретності. Позначки безпеки можуть бути явно або побічно зв'язані як окремими пакетами даних, так і з послідовностями пакетів. Зазвичай позначки безпеки використовуються для реалізації методики керування доступом на основі встановлених правил, а також для керування маршрутизацією. Передані дані також можуть мати позначки безпеки, які передаються разом з ними безпечним чином. У цьому випадку для забезпечення захисту позначок застосовуються криптографічні функції, а позначки безпеки використовуються для забезпечення контролю за цілісністю повідомлень.

Механізми виявлення подій в системах захисту інформації служать для виявлення як спроб порушення безпеки, так і для реєстрації легітимної активності користувачів. Виявлення може бути локальним та/або дистанційним і реалізуватися через тривожну сигналізацію про події (event reporting (alarm)), реєстрацію подій (event logging) і відновлювальної дії (recovery actions).

Під **контролем безпеки** розуміють незалежний розгляд і аналіз записів безпеки з метою перевірки достатності керування системою, гарантувати відповідність функціонування системи політиці безпеки і рекомендувати необхідні зміни в управлінні, політиці і процесах безпеки. Зазвичай розглядають дві процедури: протоколювання і аудит. Під протоколюванням розуміють збирання і накопичення інформації про події, що відбуваються в інформаційній системі. Під аудитом розуміють оперативний аналіз накопиченої інформації, що проводиться постійно або періодично.

Механізми протоколювання і аудиту служать для вирішення завдань:

- забезпечення підзвітності користувачів і адміністраторів, що є засобом заборони;
- забезпечення можливості відновлення послідовності подій, що дозволяє виявити слабкості в захисті інформації, виявити винуватця вторгнення, оцінити масштаби заподіяного збитку і повернутися до нормальної роботи;
- надання інформації для виявлення і аналізу проблем через підготовку відповідних звітів і рапортів.

Механізми відновлення безпеки виконують функцію реакції системи на порушення безпеки. Такими діями можуть бути, наприклад, негайне роз'єднання або припинення роботи, відмова суб'єкту в доступі, тимчасове позбавлення суб'єкта прав, занесення суб'єкта в "чорний список" та ін.

У висновку по першому розділу необхідно зазначити, що згадані загрози безпеки комп'ютерних систем і мереж, переличені послуги і механізми безпеки виділено з огляду на потреби існуючого інформаційного світу. Можливо з часом частина їх втратить свою актуальність, але, ймовірніше, з'являться нові завдання, що потребують рішення і, відповідно, список послуг і механізмів безпеки розшириться.

Список джерел інформації

1. ДСТУ 3396.2-97. Захист інформації. Технічний захист інформації. Терміни та визначення. – К.: Укр. НДІССІ, 1997. – 11 с.

2. Спесивцев А.В.Защита информации в персональных ЭВМ / А.В. Спесивцев, В.А. Вегнер, А.Ю. Крутяков и др. – М.: Радио и связь. МП "Веста", 1993. – 192 с.

3. Мафтик С. Механизми защиты в сетях ЭВМ: пер. с англ. – М.: Мир, 1993. – 216 с.

4. Мельников В.В. Защита информации в компьютерных системах / В.В. Мельников. – М.: Финансы и статистика. Электроинформ, 1997. – 367 с.

5. Романец Ю.В. Защита информации в компьютерных системах и сетях / Ю.В. Романец, П.А. Тимофеев, В.Ф. Шаньгин. – М.: Радио и связь, 2001. – 376 с.

6. Столингс В. Криптография и защита сетей. Принципы и практика / В. Столингс. – М.: Вильямс, 2001. – 672 с.

Контрольні запитання

1. Розкрійте терміни: інформація, інформація з обмеженим доступом, секретна інформація, конфіденційна інформація, захист інформації.
2. Дайте визначення національній системі конфіденційного зв'язку.
3. З якою метою було розроблено еталонну модель OSI?
4. Скільки класів функцій було виділено в моделі OSI? Перерахуйте класи.
5. Дайте класифікацію можливих атак на кожний з рівнів моделі OSI.
6. Перерахуйте базові загальноприйняті послуги безпеки Стандартів ISO 7498, ISO/IEC 10181.
7. Назвіть спеціальні механізми забезпечення безпеки зв’язку
8. В чому суть механізму нотаризації?
9. Що таке цифровий підпис?
10. В чому суть механізму управління маршрутизацією?
11. Дайте визначення автентифікації.
12. Що розуміють під конфіденційністю?
13. Що забезпечує та гарантує послуга управління доступом?
14. Наведіть приклади основних послуг безпеки.
15. Наведіть приклади основних систем і протоколів захисту інформації на прикладному, сеансовому і мережевому рівнях моделі ВВС.

2. ОСНОВНІ ЗАГРОЗИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

2.1. Порушення комп'ютерних систем. Методи протидії порушенням

Однією з важливих проблем безпеки мережевого середовища є зловмисні або, принаймні, небажані спроби вторгнення в мережу, що виконуються деякими користувачами або програмним забезпеченням. Такого роду порушення з боку користувачів можуть мати форму спроб несанкціонованого доступу до комп'ютера або спроб легального користувача одержати привілеї або виконати операції, які виходять за рамки наданих йому повноважень. Під порушеннями з боку програмного забезпечення мають на увазі роботу вірусу, "черв'яка" або "троянського коня".

Усі ці порушення належать до питань захисту мереж, оскільки вхід до системи може здійснюватися за допомогою мережі. Проте ці порушення не можна віднести до чисто мережевих. Користувач, що має доступ до локального терміналу, може спробувати проникнути до системи, не використовуючи мережевих засобів. Вірус або "троянський кінь" можуть потрапити до системи з дискети. У цьому сенсі тільки "черв'як" може вважатися чисто мережевим засобом вторгнення в систему. Таким чином, питання вторгнення до системи знаходяться на перетині галузей, що належать до захисту мереж і захисту комп'ютерних систем.

Однією з двох найпоширеніших загроз безпеки є **порушники** (другою загрозою є віруси), яких називають хакерами (hacker) або зломщиками (cracker). Дамо класифікацію порушників.

- **Імітатор (masquerader) – це** особа, що не має права користуватися комп'ютером, але подолала механізм керування доступом і використовує права доступу деякого легального користувача.
- **Правопорушник (misfeasor) – це** легальний користувач, що намагається дістати доступ до даних, програм або ресурсів, до яких він не має відповідних прав доступу, або користувач, який має в своєму розпорядженні відповідні права доступу, але використовує їх в зловмисних цілях.
- **Таємний** користувач **(clandestine** user) – це особа, що заволоділа правами керування системою і використовує ці права для обходу засобів аудиту і керування доступом або для створення перешкод у реєстрації системних подій.

Імітаторами найчастіше бувають зовнішні користувачі, правопорушниками – внутрішні , а таємними користувачами можуть виявитися як ті, так і інші.

За рівнем небезпеки атаки порушників можуть бути як незначними, так і цілком серйозними. До незначних щодо небезпек порушень можна віднести спроби тих, хто намагається отримати доступ до відповідного мережевого середовища просто з особистої цікавості. Серйозними вважаються спроби

читання секретних даних, їх несанкціоновані зміни або дії, що призводять до пошкодження системи.

2.1.1. Методика вторгнення порушників

Метою порушника є отримання доступу до системи або розширення прав, наданих йому системою на законній підставі. Для цього порушнику потрібно дістати інформацію, яка підлягає захисту. В більшості випадків вона подана у формі пароля користувача. Знаючи пароль будь-якого іншого користувача, порушник може увійти до системи під його ім'ям і отримати всі привілеї легітимного користувача.

Зазвичай в системі є файл, що пов'язує паролі з іменами легальних користувачів. Якщо цей файл не захищено, отримати доступ до нього і дізнатися паролі не важко. Файл паролів може захищатися одним з таких способів.

При **односторонньому шифруванні с**истема зберігає пароль користувача тільки в шифрованому вигляді. Коли користувач вводить свій пароль, система шифрує введений пароль і порівнює результат з тим значенням, яке зберігається у файлі паролів. На практиці система, як правило, виконує одностороннє (необоротне) перетворення, в якому пароль служить для генерування ключа функції шифрування.

При керуванні доступом доступ до файлу паролів дозволяється лише одному або дуже невеликому числу користувачів.

Якщо використовуються обидва або хоча би один з цих контрзаходів, потенційному порушнику доведеться докласти певних зусиль, щоб дізнатися паролі.

Можливі такі методи отримання паролів.

1. Перевірка паролів за умовчанням для стандартних облікових записів, що постачаються з системою. Більшість адміністраторів не обтяжують себе роботою зі зміни встановлених за умовчанням значень.
2. Перебирання усіх коротких паролів (від одного до трьох символів).
3. Перевірка слів з системного словника або із списку найбільш ймовірних паролів. Список останніх завжди можна знайти на електронних дошках оголошень хакерів.
4. Збирання інформації про користувачів, включаючи їх повні імена, імена їх дружин, чоловіків, дітей, назви картин і фотографій на робочих місцях і навіть назви їх улюблених книг, за якими можна судити про їх захоплення.
5. Перевірка паролів телефонних номерів, номерів соціальної страховки і номерів кімнат користувачів.
6. Перевірка паролів усіх можливих для даного регіону номерних знаків автомобілів.
7. Використання "троянського коня" для обходу обмежень доступу.
8. Підключення до лінії зв'язку між користувачем і головним вузлом.

Перші шість методів є варіаціями на тему підбору паролів. Якщо порушнику доводиться відгадувати варіанти пароля в процесі реєстрації в системі, то, з одного боку, це досить стомливо, а з іншого – легко виявляється системою. Наприклад, система може просто відкидати будь-які спроби реєструватися під відповідним ім'ям після трьох невдалих спроб введення пароля, що примусить порушника розірвати зв'язок з вузлом і намагатися підключитися знову. У такій ситуації немає сенсу перевіряти велике число паролів. Проте насправді порушник навряд чи стане використовувати такі примітивні методи. Наприклад, якщо порушник може дістати доступ до зашифрованого файлу паролів як непривілейований користувач, то він, швидше за все, спробує скопіювати цей файл і дешифрувати його на основі відомого механізму шифрування для даної конкретної системи, щоб у результаті одержати пароль, що дає вищий рівень привілеїв.

Підбір паролів виявляється ефективним методом і, таким чином, є небезпека тільки в тих випадках, коли він може виконуватися в автоматичному режимі і без загрози виявлення факту підбору пароля.

Під сьомим номером у наведеному вище списку вказана атака, що припускає використання "троянських коней", виявлення якої є достатньо нетривіальним завданням.

Восьмий тип атаки, що полягає в підключенні до лінії зв'язку, належить до сфери питань забезпечення фізичного захисту системи. У разі таких атак можливим методом протидії є шифрування в каналі зв'язку.

2.1.2. Методи протидії порушенням

Тепер ми перейдемо до опису двох основних методів протидії: **попередження** і **виявлення порушень.** Попередження порушень є незмінною метою системи захисту і, в якомусь сенсі, нескінченною битвою з невидимим супротивником. Тяжкість цього завдання пояснюються тим, що сторона захисту повинна намагатися запобігти всім можливим типам атак, тоді як атакуюча сторона має можливість вибрати найслабкішу ланку в загальному ланцюзі захисту і зосередити всі зусилля саме на цьому. Мета виявлення порушень полягає у вивченні типу використаної атаки до її успішного завершення або після.

Першою лінією захисту від порушників є система паролів. Практично всі системи розраховані на багатьох користувачів і вимагають від користувача при вході до системи вказувати не лише свій ідентифікатор (ім'я користувача), але й пароль. Пароль виконує функції автентифікації користувача, що входить до системи. У свою чергу, ідентифікатор користувача забезпечує захист за такими напрямками.

За ідентифікатором визначається право користувача одержувати “користувач-доступ” до системи взагалі. У багатьох системах доступ дозволяється тільки тим, хто вже має відповідний ідентифікатору запис у системі.

За ідентифікатором визначається набір привілеїв, що є у даного користувача. Деякі користувачі можуть мати статус суперкористувача, що дозволяє читати файли і виконувати дії, для яких в операційній системі передбачено особливий захист. У деяких системах передбачені спеціальні ідентифікатори для входу до системи гостей (guest) або анонімних користувачів (anonymous), і користувачі, що реєструються з такими ідентифікаторами, стають більш обмеженими в правах доступу, ніж звичайні.

Ідентифікатори користувачів відіграють важливу роль при використанні так званого розмежувального контролю доступу (discretionary access control). Наприклад, шляхом надання списку ідентифікаторів інших користувачів даний користувач може дозволити їм читати файли, власником яких він є.

Щоб краще зрозуміти природу відповідних атак, розглянемо широко вживану в системах UNIX схему, де паролі ніколи не зберігаються у відкритому вигляді (рис. 2.1). Кожен користувач сам вибирає пароль завдовжки до восьми алфавітно-цифрових символів. Цей пароль перетвориться в 56-бітове значення (на основі 7-бітових кодів ASCII), яке служить значенням, що вводиться, для ключа процедури шифрування. Процедура шифрування, що має назву crypt (3), базується на використанні алгоритму DES. Відповідний алгоритм DES виявляється дещо зміненим за допомогою спеціального 12-бітового модифікатора ("salt" value). На вхід такого модифікованого алгоритму DES спочатку подається 64-бітовий блок вхідних даних, що складається з одних нулів. Вихідне значення подається на вхід наступного кроку шифрування. Процес повторюється до тих пір, поки загальне число кроків не досягне 25. Набуте в результаті 64-бітове вихідне значення перетвориться в 11-символьну послідовність. Шифрований таким чином пароль зберігається разом з відкритим значенням модифікатора у файлі паролів у відповідному даному ідентифікатору запису.

Використання модифікатора вирішує такі задачі:

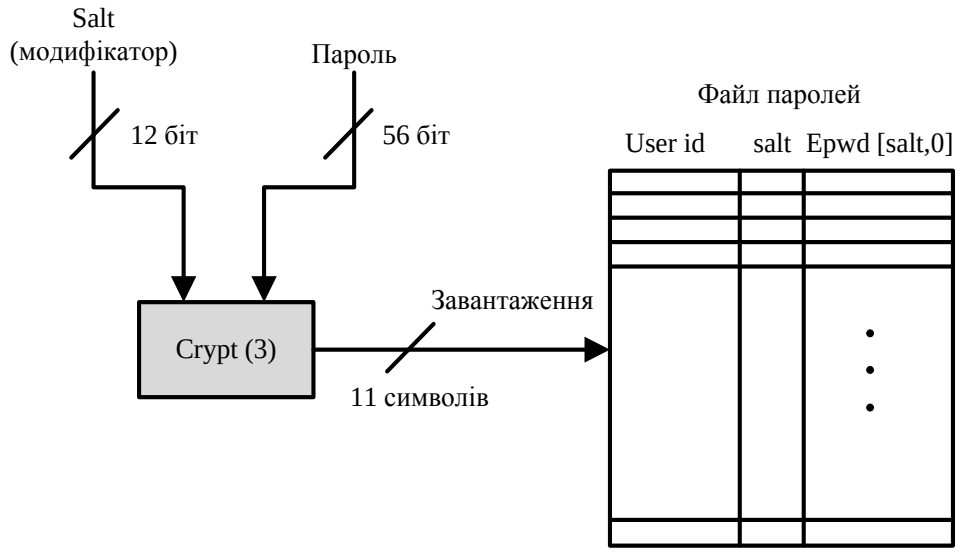
– запобігає можливості виникнення проблеми дублювання паролів у парольному файлі. Навіть якщо два різних користувачі виберуть однакові паролі, їх паролі реєструватимуться у різний час, тому "розширені" подання паролів цих користувачів будуть різними;

– збільшує ефективну довжину пароля на два символи, не примушуючи користувача запам'ятовувати ці два додаткові символи. Отже, число можливих паролів зростає в 4096 разів, що ускладнює завдання підбору пароля;

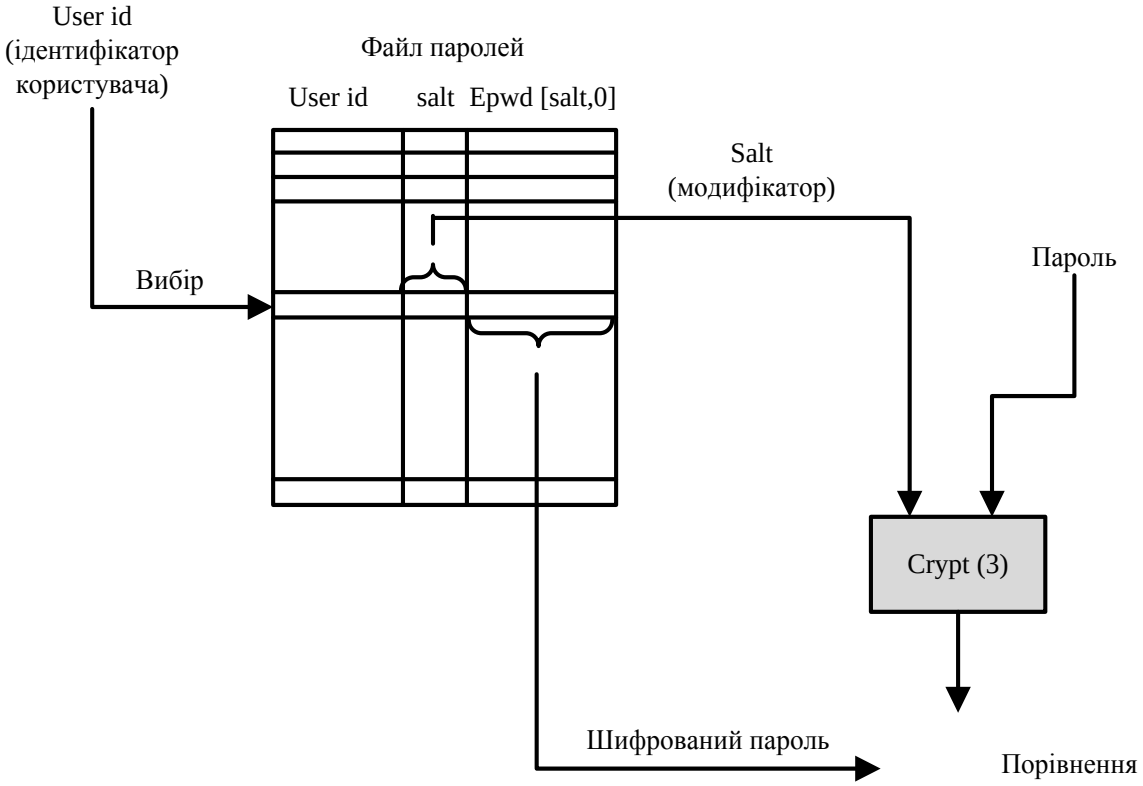
– запобігає можливості застосування апаратної реалізації DES, що спрощує проведення атак з використанням простого перебору всіх паролів.

Коли користувач намагається увійти до системи UNIX, він повинен ввести ідентифікатор і пароль. Операційна система використовує ідентифікатор для знаходження у файлі паролів відкритого значення модифікатора і шифрованого значення пароля. Модифікатор і введений користувачем пароль подаються на вхід процедури шифрування. Якщо в результаті виконання даної процедури

виходить значення, ідентичне збереженому шифрованому значенню пароля, введений пароль приймається системою як правильний.



Завантаження нового пароля



Перевірка пароля

Рисунок 2.1 – Схема використання системи паролів в UNIX

Процедура шифрування призначена для того, щоб виключити можливість вгадування пароля. Програмні реалізації DES повільніші порівняно з апаратними реалізаціями, а необхідність виконання 25 ітерацій ще в 25 разів збільшує час, потрібний для перебору. Проте з часу створення алгоритму ситуація змінилася. По-перше, було розроблено новіші і швидші реалізації алгоритму. Наприклад, в Інтернет з'явилася програма "черв'як", яка за цілком прийнятний час здатна за допомогою вдосконаленого алгоритму перевірити кілька сотень варіантів пароля для входу до системи UNIX, що атакується. По-друге, продуктивність апаратних засобів систем постійно зростає, що означає більш швидке виконання будь-яких алгоритмів.

Таким чином, можна відзначити дві загрози захисту системи паролів, реалізованої в UNIX. По-перше, користувач може дістати доступ до машини, використовуючи гостьовий ідентифікатор або якийсь інший спосіб, а потім запустити на цій машині програму підбору паролів, відому як "password cracker". При цьому порушник може перевірити сотні і навіть тисячі значень пароля, не викликаючи помітної витрати системних ресурсів. Крім того, якщо супротивнику вдасться одержати копію файлу паролів, то програму підбору паролів можна буде запустити на другому комп'ютері в режимі відсутності цейтноту. Це дає супротивнику можливість перевірити багато тисяч варіантів паролів за цілком прийнятний час.

Одним з методів запобігання можливості атаки з підбором паролів є заборона доступу до файлу паролів. Якщо частина файлу, що містить шифровані подання паролів, доступні тільки привілейованим користувачам, порушник не зможе прочитати їх без пароля привілейованого користувача. Проте у цієї стратегії є деякі недоліки.

Багато систем, включаючи більшість систем UNIX, виявляються чутливими до непередбачених вторгнень. Діставши якимсь чином доступ до системи, порушник може намагатися одержати деякий список паролів, аби кожного разу входити до системи під різними іменами, зменшуючи ризик свого власного викриття. Також якийсь легальний користувач системи може використовувати ідентифікатор і пароль іншого користувача, щоб дістати доступ до закритих даних або порушити роботу системи.

Один перебій системи захисту може зробити файл паролів відкритим для читання, внаслідок чого всі облікові записи можуть виявитися скомпрометованими.

Деякі користувачі, що мають доступ до різних машин, вводять для різних облікових записів одні й ті ж паролі. Якщо при цьому хтось зможе дізнатися пароль даного користувача на одній машині, може бути скомпрометовано й іншу машину.

Таким чином, ефективність стратегії захисту доступу до системи нероздільно пов'язана з необхідністю вибору пароля, який нелегко розгадати.

2.1.3. Стратегії вибору пароля

Підсумовуючи результати двох описаних вище досліджень, можливо констатувати, що багато користувачів створюють паролі, які легко вгадуються. В той же час, якщо користувачу надати пароль, що складається з восьми випадково вибраних алфавітно-цифрових символів, зламати такий пароль

виявляється практично неможливим. Проте в цьому випадку для більшості користувачів запам'ятати такий пароль буде так само неможливо. На щастя, якщо навіть обмежити простір таких паролів символьними рядками, які просто запам'ятовуються, то розмір цього простору все одно виявиться достатньо великим для того, щоб запобігти можливості злому. Таким чином, мета полягає у створенні умов, які виключають вибір користувачем легко вгадуваного пароля і в той же час зберігають можливість вибору пароля, що легко запам'ятовується. Рішення цієї задачі досягається за рахунок таких засобів:

- навчання користувачів;
- генерування паролів комп'ютером;
- реактивна перевірка пароля;
- попереджувальна перевірка пароля.

Користувачам необхідно роз'яснити важливість використання важко вгадуваних паролів і надати рекомендації щодо правильного вибору таких паролів. Стратегія навчання користувачів у більшості випадків виявляється малоефективною, особливо, якщо їх число надто велике або є їх постійна міграція. Одні користувачі просто ігноруватимуть рекомендації, інші не зможуть оцінити, наскільки надійними є вибрані ними паролі. Наприклад, багато хто (помилково) вважає, що запис слова в зворотному порядку або зміна останньої букви з малої літери на прописну роблять неможливою розшифрування пароля.

Генерування паролів за допомогою комп'ютера теж призводить до проблем. Якщо генерувати паролі у вигляді випадкових наборів букв і цифр, користувачі не зможуть їх запам'ятати. Навіть якщо паролі є більш менш вимовними словами, користувачі, визнаючи труднощі із запам'ятовуванням, прагнутимуть записати їх. Загалом, схема генерування паролів за допомогою комп'ютера недосконала з точки зору зручності для користувача. Один з найкращих автоматичних генераторів паролів описано в документі FIPS PUB 181. Цей документ містить не лише опис використаного підходу, але й лістинг початкового тексту алгоритму мовою С. Алгоритм генерує вимовні слова за допомогою комбінації вимовних поєднань букв. Потік символів, що формує поєднання букв і слова, забезпечується генератором випадкових чисел.

Стратегія реактивної перевірки паролів полягає в тому, що система періодично запускає власну програму підбору паролів, яка виявляє легко паролі, що вгадуються. Система відміняє всі розгадані паролі і сповіщає про це відповідних користувачів. Даний підхід має ряд недоліків. По-перше, виконання такої перевірки в повному обсязі вимагає від системи великої витрати ресурсів. Оскільки потенційний супротивник, за умови отримання ним копії файлу паролів, може залучити для рішення задачі всі ресурси свого комп'ютера протягом тривалого часу, програма реактивної перевірки паролів втратить свій ефект. Крім того, наявні паролі залишаються уразливими доти, доки програма реактивної перевірки паролів не закінчить роботу і легко паролі, що вгадуються, не будуть виявлені.

Найбільш перспективною є стратегія попереджувальної перевірки паролів. Вона дозволяє вибирати пароль на свій розсуд, але в процесі вибору система перевіряє пароль на відповідність встановленим вимогам і, якщо необхідно, відкидає його. Такий підхід заснований на переконанні, що під керівництвом системи з достатньо широкого простору допустимих паролів користувач вибере пароль, який він зможе легко запам'ятати і який при цьому буде практично неможливо вгадати за допомогою перебору значень із словника.

Проблема попереджувальної перевірки паролів полягає в необхідності досягнення балансу між стійкістю пароля і прийнятністю пароля для користувача. Якщо система відкидає дуже багато паролів, користувачі скаржаться, що вибрати відповідний пароль дуже важко. Якщо ж система використовує для перевірки придатності паролів дуже простий алгоритм, то це тільки дає зломщику можливість удосконалити свою програму підбору паролів. Далі ми обговоримо можливі підходи до реалізації попереджувальної перевірки паролів.

Перший підхід полягає в створенні простої системи контролю за дотриманням певних правил. Наприклад:

- всі паролі повинні містити не менше восьми символів;
- серед перших восьми символів мають бути: одна рядкова буква, одна прописна буква, одна цифра і один знак пунктуації.

Ці правила повинні супроводжуватися відповідними інструкціями для користувача. Хоча даний підхід кращий за просте навчання користувачів, він не завжди може зупинити зломщика паролів. Така схема дає порушнику інформацію про те, які паролі не слід перевіряти, але, зрештою, не виключає можливості успішного проведення відповідної атаки.

Іншою можливістю є просте створення великого словника потенційно "поганих" паролів. При виборі користувачем пароля система перевіряє, чи не потрапляє вибраний пароль у "чорний список". Даний підхід може ускладнюватись двома проблемами.

Перша проблема – це **проблема обсягу.** Аби описаний вище підхід мав ефект, словник повинен бути достатньо великим. Наприклад, файл словника, що використовувався для досліджень в Університеті Педью, мав обсяг більше 30 Мбайт.

Друга проблема – це **проблема часу.** Час пошуку у великому словнику також може стати надмірним. Крім того, для перевірки можливих модифікацій слів необхідно або включити ці модифікації до словника, що зробить його ще величезнішим, або витрачати додатковий час на перевірку модифікацій слів за допомогою спеціальних алгоритмів.

На сьогодні перспективними є дві методики проведення ефективної і достатньо надійної перевірки паролів. Одна з них будується на основі використання марківської моделі генерування паролів, що вгадуються. На рис. 2.2 показана спрощена версія такої моделі. Ця модель задає мову, алфавіт якої складається з трьох символів. Стан системи у будь-який момент ідентифікується останнім вибраним символом. Коефіцієнти переходів з одного стану в інший

задаються ймовірністю проходження одних символів за іншими. Наприклад, імовірність того, що наступною буквою буде *b* за умови, що поточною є буква *a*, в даній ситуації дорівнює 0,5.

У загальному випадку марківська модель задається чотирма значеннями *[m, A, T, k*, де *m* – позначає число станів моделі, *A* – простір станів, *T* – матрицю імовірності переходів, а *k* – порядок моделі. Для моделі порядку *k* імовірність переходу до того або іншого символу алфавіту залежить від значень *k* попередніх символів. На рис. 2.2 показано схему простої моделі першого порядку.

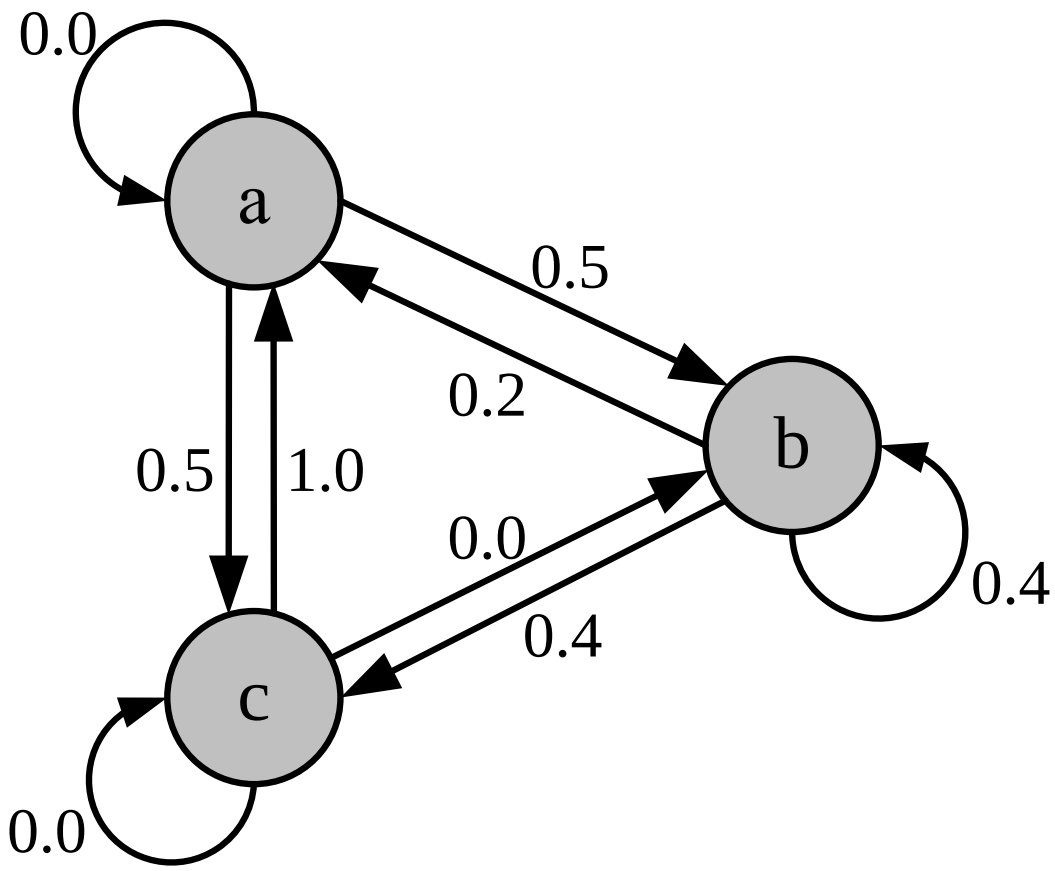


Рисунок 2.2 – Приклад марківської моделі

M=[3,{a,b,c}, E, 1],
де

$$T = \begin{matrix} \mathbf{0.0} & \mathbf{0.5} & \mathbf{0.5} \\ \mathbf{0.2} & \mathbf{0.4} & \mathbf{0.4} \\ \mathbf{1.0} & \mathbf{0.0} & \mathbf{0.0} \end{matrix}$$

Приклад рядка, ймовірного для цієї мови: *abbcacaba.*
Приклад рядка, що не є ймовірним для цієї мови: *aacccbaaa.*

Існує можливість побудови і використання моделі другого порядку. Спочатку створюється словник легко вгадуваних паролів. Після цього обчислюється матриця переходів таким чином.

Визначається матриця частот f, де f(i, j, k) відповідає числу поява тріграм, що складаються з і-го, j-го і к-го символів. Наприклад, пароль *parsnips* містить тріграми *par, ars, rsn, sni, nip* і *ips.*

Для кожної біграми ij обчислюється f (i, j, ∞) як загальне число тріграм, що починається з ij. Наприклад t(a, b, ∞) є загальне число тріграм вигляду *aba, abb, abc* и та ін.

Елементи матриці T обчислюються за формулою (2.1):

$$T(i, j, k) = \frac{f(i, j, k)}{f(i, j, \infty)}. \qquad (2.1)$$

У результаті виходить модель, що відображає структуру слів у словнику. За допомогою цієї моделі питання "чи є даний пароль поганим?" трансформується в питання "чи породжується даний рядок (тобто пароль) за допомогою даної моделі Маркова?". Для введеного пароля можна знайти ймовірність переходів для всіх тріграм. Потім за допомогою стандартних статистичних тестів можна з'ясувати, чи виявляється даний пароль рядком, імовірним або неймовірним для даної моделі. Паролі, які з певною часткою ймовірності породжуються даною моделлю, відкидаються.

Таким чином, для моделі другого порядку виходять добрі результати. Запропонована система знаходить практично всі паролі зі словника і, разом з тим, не відкидає значної кількості потенційно хороших паролів, залишаючись достатньо дружньою по відношенню до користувача.

2.1.4. Виявлення порушників

Навіть краща система запобігання порушенням часто не спрацьовує. Тому іншою лінією захисту повинна стати система виявлення порушень, що останнім часом привертає увагу все більшого числа дослідників. Інтерес до цих питань обумовлений цілим рядом причин, зокрема:

– якщо порушення буде виявлено достатньо швидко, порушника можна буде ідентифікувати і позбавити доступу до системи перш, ніж він встигне пошкодити або скомпрометувати дані. Навіть якщо порушення буде зафіксовано надто пізно, аби перешкодити діям порушника, все ж, чим раніше порушення буде виявлено, тим меншим буде збиток і швидшим відновлення системи;

– ефективна система виявлення порушень сама по собі може служити як засіб для відлякування, частково виконуючи функції системи запобігання вторгненням;

– виявлення порушень дозволяє збирати інформацію про методи вторгнення, яку згодом можна використовувати для посилення системи запобігання вторгненням.

Виявлення порушень ґрунтується на припущенні, що поведінка порушника відрізняється від поведінки легального користувача і відповідні відмінності можна подати в кількісному виразі. Звичайно ж, не можна чекати, що ми зможемо спостерігати абсолютно різне використання ресурсів порушником порівняно з легальним користувачем. Правильніше припустити, що в поведінці того й іншого будуть загальні риси.

На рис. 2.3 в дуже абстрактному вигляді ілюструється природа завдання, що стоїть перед розробником системи виявлення порушень. Хоча типова поведінка порушника й відрізняється від типової поведінки легального користувача, сфери, які характеризують їхню поведінку, все ж таки, перетинаються.

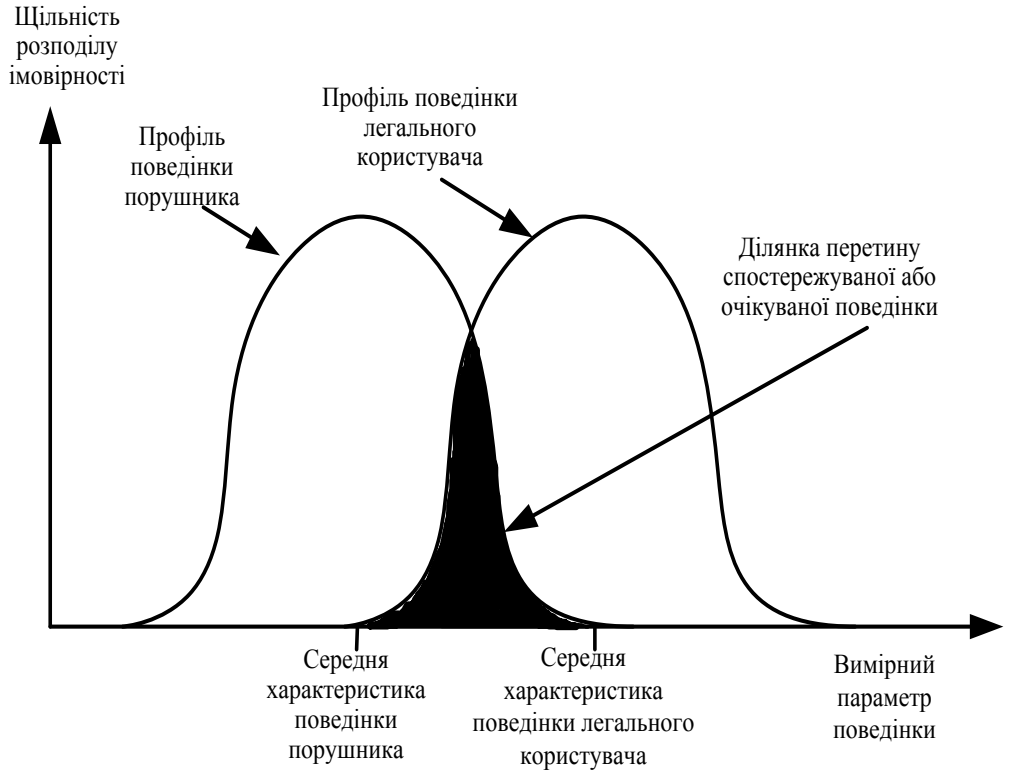


Рисунок 2.3 – Профілі характеристик поведінки порушників і легальних користувачів

Таким чином, чим більш грубо інтерпретуватиметься поведінка порушника, тим вищою буде імовірність "помилкових спрацьовувань", тобто визнання легальних користувачів порушниками. В той же час прагнення позбавитися помилкових спрацьовувань шляхом суворої інтерпретації поведінки порушника веде до ситуацій, коли порушники залишаються невиявленими.

Отже, виявлення правопорушника (легального користувача, що виконує несанкціоновані операції) є складним завданням, оскільки в даному випадку нелегітимна поведінка може майже не відрізнятися від легітимної. Поведінку правопорушника можна виявити, якщо правильно визначити клас умов, за яких відбувається несанкціоноване використання ресурсів. Нарешті, виявлення таємних користувачів, здається, взагалі виходить за рамки одних лише методів автоматичного виявлення.

Пропонуються такі підходи до розв’язання проблеми виявлення порушень.

1. Виявлення на базі статистичних відхилень передбачає збирання даних, що характеризують поведінку легальних користувачів, протягом певного часу. Потім ці дані аналізуються із застосуванням статистичних методів, щоб з високим ступенем точності визначити, відповідає поведінка певного користувача поведінці легального користувача чи ні.

Найбільш поширеними є:

– використання порогових значень. Даний підхід припускає визначення незалежних від конкретного користувача порогових значень, що характеризують частоту виникнення різних подій.

– використання профілю поведінки. Створюється профіль активності користувача і за допомогою цього профілю виявляються відхилення в поведінці користувача, що реєструється в системі під даним ім'ям.

2. Виявлення на базі правил. Припускає розробку набору правил, на основі яких ухвалюється рішення про те, що даний тип поведінки є поведінкою порушника.

Найбільш поширеними є:

– виявлення аномалій**.** Розробляються правила, що дозволяють виявити відхилення від поведінки, які спостерігались раніше .

– ідентифікація вторгнення**.** Підхід на основі використання експертної системи, що виявляє підозрілу поведінку.

По суті, в підході, основаному на статистичних методах, робиться спроба визначити нормальну або очікувану поведінку, тоді як у підході, основаному на правилах, робиться спроба виявити фактичну поведінку.

Щодо типів порушників, визначених вище, виявлення статистичних аномалій є ефективним відносно імітаторів, які навряд чи стануть наслідувати поведінку легального користувача, під ім'ям якого входять в систему. В той же час ця методика практично непридатна для правопорушників. Проти такого типу атак ефективніший підхід, оснований на використанні правил, що дозволяють розпізнати окремі події й їх послідовності, які в певному контексті

означають вторгнення. На практиці в системі реалізується певна комбінація обох підходів, що забезпечує ефективне протистояння широкому спектру порушень.

Основним засобом виявлення порушень є контрольний запис (audit record), який передбачає виконання запису деяких операцій, що здійснюються користувачами для подальшого застосування цих записів як початкових даних системи виявлення порушень. Найчастіше застосовують такі дві стратегії.

Системні контрольні записи (native audit records). Практично всі операційні системи, розраховані на велику кількість користувачів, містять програми для збору інформації про дії, що виконуються користувачами. Перевага використання цієї інформації полягає у відсутності додаткового програмного забезпечення. Недолік пов'язаний з тим, що системні контрольні записи можуть не містити необхідної інформації або можуть містити її в невідповідній формі.

Спеціальні контрольні записи (detection-specific audit records). Можуть бути розроблені і застосовані спеціальні засоби збору інформації, які генерують записи, що містять дані, необхідні тільки для роботи системи виявлення порушень. Перевагою такого підходу є незалежність від постачальника системи і здатність до перенесення на різні платформи. Як недолік слід зазначити додаткове навантаження на систему у зв’язку з необхідністю виконання двох програм ведення контрольних записів.

Кожен контрольний запис містить такі поля:

- **Subject (суб'єкт).** Ініціатор дії. Суб'єктом зазвичай є користувач термінала, але це може бути і процес, що виконується для користувача або групи користувачів. Усі дії виконуються за командами, які дає суб'єкт. Суб'єкти можуть бути згруповані в класи за рівнями доступу, і ці класи можуть перетинатися;
- **Action (дія).** Операція, що виконується суб'єктом по відношенню до об'єкта, наприклад реєстрація входу в систему, читання, введення-виведення даних, виконання програм;
- **Object (об'єкт).** Рецептори дій. Прикладами об'єктів є файли, програми, повідомлення, записи, термінали, принтери, а також структури, створені користувачами або програмами. Коли рецептором дії є суб'єкт (наприклад, при отриманні електронної пошти), цей суб'єкт теж розглядається як об'єкт. Об'єкти можуть бути згруповані за типами і відповідно до типу об'єкта і умов оточення. Наприклад, дії в базі даних можуть реєструватися як на рівні бази даних у цілому, так і на рівні окремих записів;
- **Exception-Condition (виняткова ситуація)**. Зазначає тип виняткової ситуації, якщо вона виникла при виконанні команди повернення;
- **Resource-Usage (використання ресурсу)**. Список кількісних показників, в якому кожен елемент зазначає обсяг використання того або іншого ресурсу (наприклад, число надрукованих або відображених на екрані рядків, число прочитаних або створених записів, час використання процесора, число задіяних каналів введення-виведення, тривалість сеансу зв'язку);

• **Time-stamp (позначка дати-часу**). Унікальна позначка дати і часу, яка вказує на момент виконання дії.

Більшість дій, що виконуються користувачем, складаються з набору елементарних операцій. Наприклад, копіювання файлу означає виконання команди користувача, що припускає отримання права доступу і створення копії плюс читання з одного файлу, плюс запис в інший файл. Розглянемо команду

COPY GAME.EXE TO <Library>GAME.EXE,

ініційовану користувачем Serg з метою копіювання виконаного файлу *GAME* з поточного каталога в каталог <*Library*>. У цій ситуації можуть генеруватися такі контрольні записи.

Serg	execute	<Library>COPY.EXE	0	CPU=00002	11058721678
Serg	read	<Serg>GAME.EXE	0	RECORDS=0	11058721679
Serg	execute	<Library>COPY.EXE	write-viol	RECORDS=0	11058721680

У прикладі процес копіювання був завершений аварійно, оскільки користувач Serg не має права запису в каталозі *<Library>*.

Розкладання дій користувача на елементарні операції має такі переваги.

Оскільки об'єкти в системі є додатками, що захищуються, використання елементарних операцій дозволяє простежити за всіма діями, які виконуються з даним об'єктом. Система може виявляти спроби порушення контролю доступу (за відсутності відхилень від норми виняткових ситуацій) і виявляти успішні випадки таких порушень (за відсутності відхилень від норми в наборі об'єктів, доступних даному суб'єкту).

Принцип створення контрольних записів для кожного об'єкта і кожної дії спрощує модель та її реалізацію.

Зважаючи на просту і одноманітну структуру спеціальних контрольних записів, відносно просто одержати відповідну інформацію або її частину шляхом копіювання з наявних системних контрольних записів у спеціальні контрольні записи.

Як було сказано, **методи виявлення статистичних аномалій** можна розділити на дві категорії: виявлення за **пороговими значеннями** (threshold detection) і виявлення за **профілем поведінки** (profile-based detection). Виявлення за пороговими значеннями припускає ведення обліку частоти виникнення подій певного типу за певний інтервал часу. Якщо частота перевищує значення, яке вважається розумним, система розглядає цей інцидент як вторгнення порушника.

Аналіз порогових значень у чистому вигляді стає дуже грубим і неефективним методом виявлення навіть у разі атак середньої складності. Необхідно правильно визначити і порогові значення, і тимчасові інтервали.

Оскільки користувачі працюють по-різному, просте використання порогових значень призведе або до численних помилкових спрацьовувань (false positive), або до численних помилкових неспрацьовувань (false negative). Проте простий детектор перевищення порогових значень може бути корисним у сукупності з іншими складнішими методами.

Виявлення за профілем поведінки будується на вивченні попередньої поведінки користувача або групи користувачів і порівнянні цієї поведінки з поточною поведінкою на предмет виявлення значних відхилень. Профіль може складатися з набору параметрів, так що відхилення за одним з параметрів може бути недостатнім для того, щоб генерувати системну тривогу.

Даний підхід спирається на аналіз вмісту контрольних записів. Контрольні записи забезпечують введення для функції виявлення порушень таким чином. По-перше, розробник повинен вирішити, яке число параметрів необхідно відстежувати в поведінці користувача. Щоб виявити профіль активності типового користувача, можна провести аналіз контрольних записів впродовж деякого періоду часу. По-друге, вміст поточних записів служить як початкові дані, за якими виявляються порушення. Таким чином, запропонована модель виявлення порушень припускає аналіз контрольних записів поведінки, які надходять, на предмет відхилення цієї поведінки від звичайної.

Прикладами параметрів, які виявляються корисними при виявленні порушень за профілем поведінки, є такі.

Лічильник (counter). Невід'ємне ціле число, значення якого можна збільшувати, але не зменшувати, до тих пір, поки це значення не буде переустановлене в результаті дії програми керування. Звичайно підрахунок числа певних подій, що спостерігалося, ведеться протягом деякого проміжку часу. Прикладами можуть бути число спроб входу в систему, зроблених користувачем протягом однієї години, число викликів певної команди протягом сеансу роботи користувача або число неправильно введених паролів протягом однієї хвилини.

Датчик (gauge) – це невід'ємне ціле число, значення якого може як збільшуватися, так і зменшуватися. Зазвичай датчик призначений для реєстрації поточного значення деякої характеристики об'єкта. Прикладами є число логічних з'єднань, установлених застосуванням користувача, або число вихідних повідомлень, поставлених у чергу призначеним процесом.

Інтервальний таймер (interval timer) – це довжина періоду часу між двома послідовними подіями. Наприклад, довжина періоду часу між двома послідовними спробами реєстрації за одним і тим же обліковим записом.

Показник використання ресурсу (resource utilization) – це обсяг споживання ресурсу за певний проміжок часу. Прикладами є число сторінок, віддрукованих за час сеансу роботи, або загальний час виконання певної програми.

З цими кількісними показниками можна використовувати різні тести для з'ясування правомірності поточної діяльності користувача. При цьому можна використовувати такі методи:

– метод середніх значень і середньоквадратичних відхилень;
– метод багатовимірної моделі;
– метод марківських процесів;
– метод часових рядів;
– операторний метод.

Найпростіший статистичний тест полягає в розгляді середніх значень і середньоквадратичних відхилень параметрів за певний період часу. Результати характеризують середню поведінку і його відхилення від середньої. Використовувати середні значення і середньоквадратичні відхилення можна для широкого спектра лічильників, таймерів і показників використання ресурсів. Але значення цих параметрів самі по собі зазвичай дають дуже приблизну оцінку, щоб використовувати їх безпосередньо в цілях виявлення вторгнення.

Метод багатовимірної моделі ґрунтується на використанні кореляції між двома або декількома змінними. Поведінку порушника можна характеризувати з більшою надійністю, розглядаючи такі кореляції (наприклад, часу використання процесора і ресурсів або частоти входу до системи і тривалість сеансу роботи).

Метод марківських процесів служить для визначення імовірності переходів між різними станами. Наприклад, ця модель дозволяє з'ясувати характер зв'язку між певними командами.

Метод тимчасових рядів оснований на аналізі інтервалів часу з метою виявлення подій, які проходять або дуже швидко, або дуже поволі. При цьому для того щоб охарактеризувати тимчасові аномалії, теж можна застосувати різні статистичні перетворення.

Нарешті, операторна модель базується швидше на визначенні того, що вважається таким, що виходить за рамки звичайної поведінки користувача, а не на автоматичному аналізі вмісту збережених контрольних записів. Зазвичай встановлюються чітко певні межі, і вихід за ці межі розглядається як підозра на вторгнення в систему. Цей підхід краще за все працює тоді, коли поведінка порушника характеризується певними типами його дії. Наприклад, велике число спроб входу до системи і реєстрації за короткий період часу дозволяє зробити висновок про спробу вторгнення.

Як приклад використання розглянутих вище параметрів і моделей розглянемо в табл. 2.1, в якій зібрані повідомлення про різні критерії, що враховуються в системі виявлення порушень IDES, використовуваній в Стенфордському дослідницькому інституті (SRI - Stanford Research Institute).

Таблиця 2.1 – Критерії, використовувані для виявлення порушень

Критерій	Модель	Тип порушень
1	2	3
Вхід до системи і сеанс роботи користувача		
Частота входів до системи по днях і годинах	Середні значення і середньоквадратичні відхилення	Порушники, найімовірніше, намагаються увійти до системи в неробочий час
Час, що пройшов з моменту останнього входу до системи	Операторна модель	Спроба вторгнення до системи за "нічийним" обліковим записом
Тривалість сеансів роботи	Середні значення і середньоквадратичне відхилення	Значні відхилення можуть означати роботу імітатора
Обсяг даних, що пересилаються в певне місце	Середні значення і середньоквадратичне відхилення	Дуже великі обсяги даних, передані на видалені вузли, можуть означати просочування важливої інформації
Показник використання ресурсів під час сеансу	Середні значення і середньоквадратичне відхилення	Показники завантаження процесора або пристроїв введення-виведення, що виходять за рамки звичайних, можуть означати роботу порушника
Число введень неправильних значень пароля при реєстрації	Операторна модель	Спроби проникнення до системи за допомогою вгадування пароля
Невдалі спроби входу до системи з певних терміналів	Операторна модель	Спроби вторгнення до системи

Продовження таблиці 2.1

1	2	3
Виконання команд або програм		
Частота запуску програм	Середні значення і середньоквадратичне відхилення	Може вказувати на присутність порушника, який пробує доступні йому команди, чи на легального користувача, що дістав доступ до привілейованих команд
Використання ресурсів програмами	Середні значення і середньоквадратичне відхилення	Значення, що виходить за рамки звичайного, може означати впровадження вірусу або троянського коня, які у фоновому режимі виконують операції, які збільшують завантаження системи введення-виведення або процесора
Число відмов виконання	Операторна модель	Може означати спроби легального користувача одержати привілеї вищого рівня
Доступ до файлів		
Частота виконання операцій читання, запису, створення, видалення	Середні значення і середньоквадратичне відхилення	Частота, що виходить за рамки звичайної, операцій доступу до файлів для читання і запису може означати присутність імітатора або перегляд ресурсів
Підрахунок невдалих спроб читання, запису, створення	Операторна модель	Можуть виявляти користувачів, які постійно намагаються дістати несанкціонований доступ до файлів

Головна перевага використання статистичних профілів полягає в тому, що для їх застосування не потрібно наперед знати про всі вади в системі захисту. Програма-детектор з'ясовує, яка поведінка є "нормальною", а потім виявляє відхилення. Даний підхід не ґрунтується на характеристиках системи і

відомостях про її уразливість, тому відповідна реалізація легко переноситься з однієї системи на іншу.

Методи виявлення порушень, які основані на використанні правил, передбачають відстеження подій, що відбуваються в системі, і застосування набору правил, за якими можна зробити висновок, чи є дана поведінка підозрілою чи ні. Загалом, усі ці методи можна розділити на два класи: методи, основані на виявленні аномалій, і методи, що ідентифікують подолання захисту.

Виявлення аномалій в даному випадку схоже за підходами і можливостями на методи виявлення статистичних аномалій. При використанні бази правил аналіз збережених контрольних записів проводиться з метою виявлення характеристик типової поведінки і автоматичного генерування правил, що описують таку поведінку. Правила можуть являти собою поведінкові шаблони користувачів, програм, привілеїв, тимчасових інтервалів, терміналів і т.п. Потім спостерігається поточна поведінка, і кожна транзакція перевіряється за набором правил на предмет її відповідності одержаним раніше поведінковим шаблонам.

Як і при виявленні статистичних аномалій, виявлення аномалій, основаних на правилах, не вимагає знання вразливих місць захисту системи. Схема спирається на спостереження за поведінкою користувачів і на припущення, що в майбутньому ця поведінка не повинна істотно змінитися. Щоб цей підхід виявився ефективним, буде потрібна достатньо велика база правил.

Ідентифікація подолання захисту ґрунтується на абсолютно іншому підході, пов'язаному з технологією експертних систем. Основною межею таких систем є використання правил для ідентифікації відомих видів вторгнень або вторгнень, побудованих на відомих недоліках системи захисту. Можна визначити і правила для ідентифікації підозрілої поведінки, навіть якщо поведінка не виходить за рамки типової. Зазвичай правила, вживані в таких системах, залежать від конкретного типу машини і операційної системи. Крім того, такі правила генеруються експертами, а не в результаті автоматичного аналізу контрольних записів. Найчастіше проводиться опитування системних адміністраторів і аналітиків системи захисту для отримання відомих сценаріїв і подій, що становлять загрозу безпеці системи, яка захищається. Таким чином, успіх даного підходу залежить від професіоналізму тих, хто бере участь у виробленні системи правил.

Простий приклад типів правил, які можуть при цьому використовуватися, можна знайти в системі NDIX – одній з перших систем евристичних правил, за допомогою яких можна визначити ступінь підозрілості тієї або іншої діяльності. Приклади таких евристичних правил описані нижче.

Користувачі не повинні читати файли, що знаходяться в особистих каталогах інших користувачів.

Користувачі не мають права записувати інформацію у файли, що належать іншим користувачам.

Користувачі, які постійно працюють з системою, часто при новому вході до системи відкривають ті ж файли, які вони використовували раніше.

Користувачі рідко відкривають дискові пристрої безпосередньо, а використовують для цього утиліти вищого рівня, пропоновані операційною системою.

Користувачі не повинні відкривати декілька сеансів роботи з однією і тією ж системою одночасно.

Користувачі не копіюють системні програми.

Схема ідентифікації вторгнень, реалізована в системі IDES, основана на описаній вище стратегії. Контрольні записи у міру їх появи перевіряються по базі правил. Якщо виявляється збіг, відбувається збільшення рейтингу підозри (suspicion rating) користувача. Якщо збіги спостерігаються для достатньо великого числа правил, рейтинг перевищує заданий поріг і система генерує звіт про виявлену аномалію.

Підхід IDES оснований на перевірці контрольних записів. Слабкою стороною цього підходу є недостатня гнучкість. Наприклад, можна реалізувати такий сценарій вторгнення, коли система генерує ряд послідовностей контрольних записів, які слабо або майже непомітно відрізняються від інших. При цьому укласти такі відхилення в рамки наявних правил зовсім непросто.

2.1.5. Розподілені системи виявлення порушень

До недавнього часу всі роботи зі створення системи виявлення порушень були зосереджені навколо окремо взятої обчислювальної системи. Проте типовій організації потрібно забезпечити захист розподіленого комплексу обчислювальних вузлів, об'єднаних локальною мережею або засобами міжмережевої взаємодії. Звичайно, можна захистити кожен окремий вузол, використовуючи в кожному з них свою систему виявлення порушень, але ефективніший захист досягається шляхом координації і взаємодії систем виявлення порушень у мережі. Таким чином, формулюються наступні головні питання, що виникають при проектуванні розподіленої системи виявлення порушень.

Розподіленій системі виявлення порушень, можливо, доведеться мати справу з різними форматами контрольних записів. У гетерогенному середовищі різні системи використовують різні системи створення контрольних записів, тому для виявлення порушень ці системи можуть мати різні формати для створюваних контрольних записів.

Деякі вузли мережі повинні бути місцем накопичення і аналізу даних, що надходять до них від усіх інших систем у мережі. Тому сирі контрольні або вже оброблені дані повинні передаватися мережею. Тобто необхідно забезпечити цілісність і конфіденційність цих даних. Цілісність не дозволить порушнику маскувати свою діяльність шляхом зміни переданої контрольної інформації. Конфіденційність потрібна тому, що контрольна інформація може виявитися дуже важливою з точки зору підтримки працездатності системи.

Система може мати як централізовану, так і децентралізовану архітектуру. У першому випадку передбачається наявність одного центру, де накопичуються і аналізуються всі контрольні дані. Це спрощує завдання узгодження звітів, що надходять, але створює потенційно "вузьке місце", відмова якого може призвести до виходу з ладу всієї системи. При використанні децентралізованої архітектури є декілька аналітичних центрів, проте при цьому потрібно координувати їх діяльність і організувати обмін інформацією між ними.

Хорошим прикладом розподіленої системи виявлення порушень є система, яка подана на рис. 2.4. На цьому рисунку показана загальна архітектура цієї системи, що складається з таких трьох основних компонентів: модуля агента вузла, модуля агента диспетчера локальної мережі, модуля центрального адміністратора

Модуль агента вузла – це модуль збирання контрольної інформації, що виконується у фоновому режимі в системі, за якою ведеться спостереження. Метою модуля є збір даних про події, що мають відношення до захисту вузла і передача цих даних модулю центрального адміністратора.

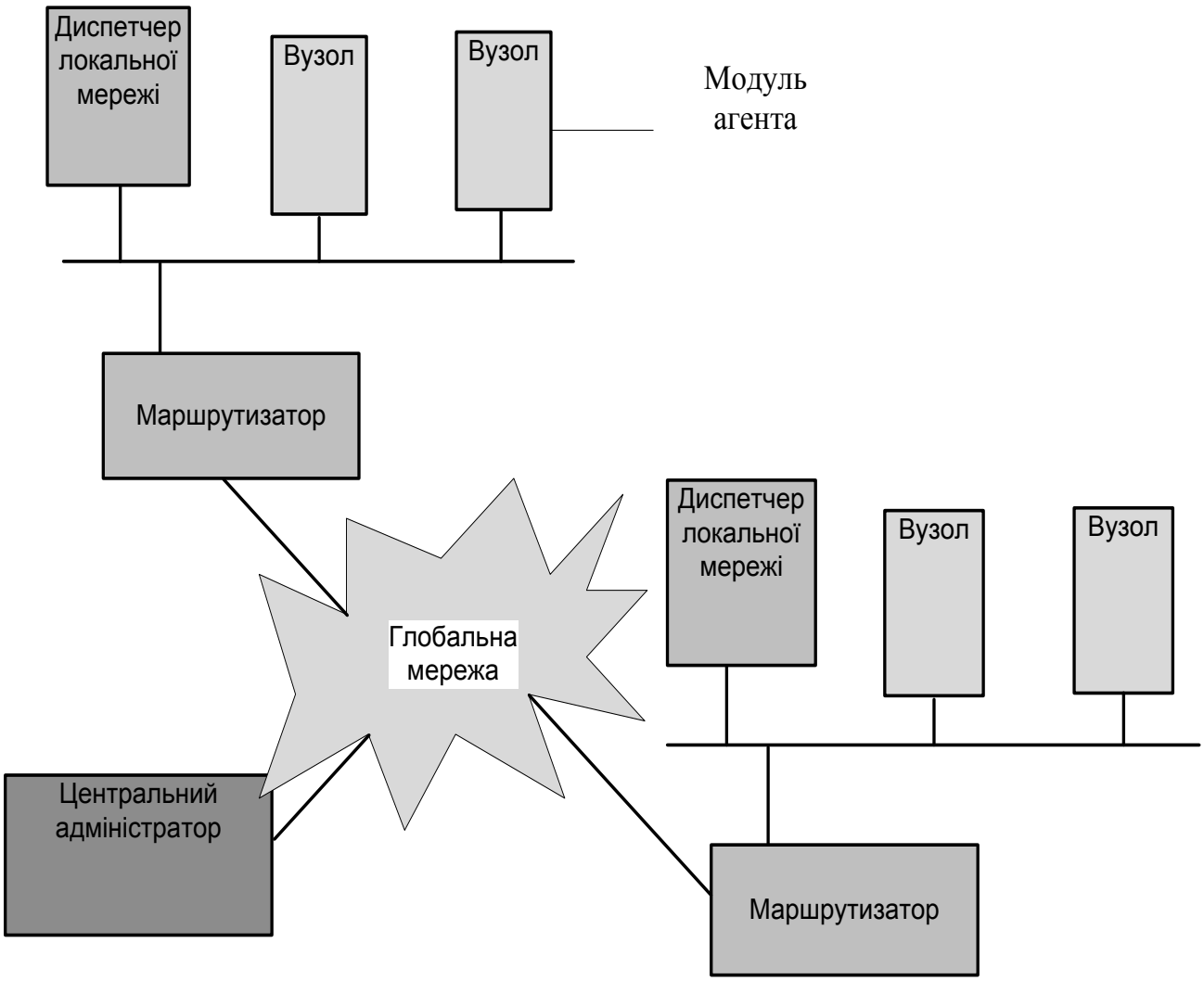


Рисунок 2.4 – Архітектура розподіленої системи виявлення порушень

Робота **модуля** агента диспетчера локальної мережі аналогічна роботі модуля агента вузла, але даний модуль аналізує потік даних локальної мережі, теж передаючи одержані дані модулю центрального адміністратора.

Модуль центрального адміністратора приймає повідомлення, що надходять від агентів вузлів і агента диспетчера локальної мережі, обробляє ці повідомлення, з'ясовуючи їх кореляцію і намагаючись виявити порушення.

Схема розроблялася для того, щоб бути незалежною від операційної системи і конкретних реалізацій системи контролю. На рис. 2.5. показана схема загального підходу, що застосовувався в цьому випадку. Агент переглядає кожен контрольний запис, породжений системою реєстрації контрольних параметрів відповідної системи. Записи, що не мають інтересу з погляду захисту, фільтруються. Записи, що залишилися, перетворяться в стандартний формат контрольного запису головного вузла (HAR – host audit record). Потім набір шаблонів, який використовував логічний модуль, аналізує одержані записи на предмет виявлення підозрілих дій. Перш за все агент проводить пошук у записах подій, що становлять інтерес з точки зору безпеки, незалежно від їх зв'язку з минулими подіями.

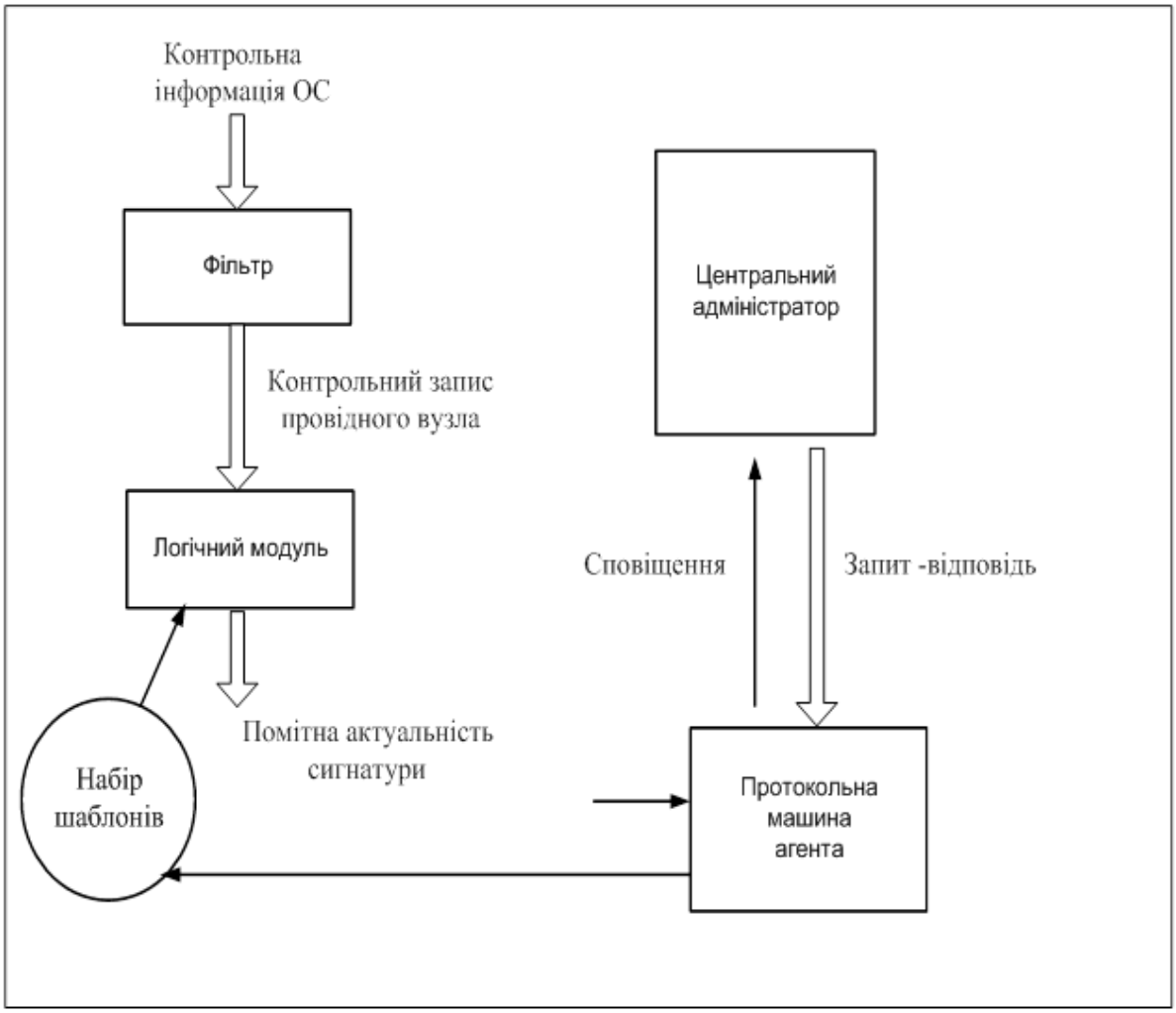


Рисунок 2.5 – Архітектура агента

Прикладами таких подій є відмови в доступі до файлів, доступ до системних файлів і зміна параметрів контролю доступу до файлів.

На наступному рівні пошуку агент шукає послідовності подій, відповідних вже відомим спробам злому (так звані сигнатури). Нарешті, агент намагається знайти аномалії поведінки окремих користувачів, ґрунтуючись на профілі кожного користувача, що містить відомості про число виконуваних програм, число файлів, до яких отримано доступ, і т.п.

Якщо виявляється підозріла активність, центральному адміністратору надсилається повідомлення. Модуль центрального адміністратора включає експертну систему, здатну виявляти взаємозалежність одержаних даних. Модуль центрального адміністратора може також запитати у окремих вузлів копії їх записів HAR, щоб порівняти їх із записами інших агентів.

Модуль агента диспетчера локальної мережі теж надає зібрані їм дані модулю центрального адміністратора. Агент диспетчера локальної мережі контролює з'єднання між вузлами, використані служби і рівень трафіку. Цей агент реєструє всі помітні події, наприклад різка зміна завантаження мережі, використання служб захисту і мережеві операції типу rlogin.

Таким чином, архітектурні рішення, показані на рис. 2.4 і 2.5, виявляються достатньо загальними і гнучкими, щоб лягти в основу машинно-незалежного підходу, що дозволяє створити реалізацію, масштабовану від системи виявлення порушень на рівні окремого вузла до рівня всієї мережі, і на основі порівняння активності окремих вузлів виявляти підозрілі дії, які інакше залишалися б непоміченими.

2.2. Програмні загрози, віруси, антивіруси

Поглядів з приводу народження першого комп'ютерного вірусу дуже багато. Безсумнівно лише одне: на машині Чарльза Беббіджа, який вважається винахідником першого комп'ютера, вірусів не було, а на Univax 1108 і IBM 360/370 у середині 1970-х років вони вже були.

Не дивлячись на це, сама ідея комп'ютерних вірусів з'явилася значно раніше. Відправною точкою можна вважати праці Джона фон Неймана з вивчення математичних автоматів, що самовідтворюються. Ці роботи стали відомими в 1940-х роках. А в 1951 р. знаменитий вчений запропонував метод, який демонстрував можливість створення таких автоматів. Пізніше, в 1959 р., журнал "Scientific American" опублікував статтю Л.С. Пенроуза, яка також була присвячена механічним структурам, що самовідтворюються. На відміну від раніше відомих робіт, тут була описана проста двовимірна модель подібних структур, здатних до активації, розмноження, мутацій, захоплення. Пізніше на базі цієї статті інший учений – Ф.Ж. Шталь – реалізував модель на практиці за допомогою машинного коду на IBM 650.

Необхідно зазначити, що із самого початку ці дослідження були спрямовані зовсім не на створення теоретичної основи для майбутнього

розвитку комп'ютерних вірусів. Навпаки, вчені прагнули удосконалити світ, зробити його більш пристосованим для життя людини. Адже саме ці праці лягли в основу багатьох робіт з робототехніки і штучному інтелекту. І в тому, що подальші покоління зловживали плодами технічного прогресу, немає провини цих чудових вчених.

У 1962 р. інженери з американської компанії Bell Telephone Laboratories – В.А. Висоцькій, Г.Д. Макілрой і Роберт Моріс – створили гру "Дарвін". Гра передбачала присутність у пам'яті обчислювальної машини так званого супервізора, що визначав правила і порядок боротьби між собою програм-суперників, які створювалися гравцями. Програми мали функції дослідження простору, розмноження і знищення. Сенс гри полягав у видаленні всіх копій програми супротивника і захопленні поля битви.

На цьому теоретичні дослідження учених і розробки інженерів пішли в тінь, і незабаром світ дізнався, що теорія структур, здатних до саморозмножування, з не меншим успіхом може бути використана і в інших цілях, наприклад, як комп'ютерний вірус.

Комп'ютерний вірус – це програма, що викликає порушення роботи інших програм, псування інформації, неможливість прочитати файли, уповільнення або нестабільність роботи комп'ютера. Комп'ютерні віруси здатні самостійно розповсюджуватися мережею на інші комп'ютери. Деякі комп'ютерні віруси можуть видозмінюватися. Засобом боротьби з вірусами є створення антивірусних програм. **Троянська програма** – це шкідлива комп’ютерна програма, яка розповсюджується тільки людьми.

Основна маса вірусів і троянських програм у минулому створювалася студентами і школярами, які тільки що вивчили мову програмування, хотіли спробувати свої сили, але не змогли знайти для них більш гідного застосування. Відрадний той факт, що значна частина подібних вірусів їх авторами не розповсюджувалася і віруси через деякий час вмирали разом з дисками, на яких зберігалися. Такі віруси писалися і пишуться до цього дня тільки для самоствердження їх авторів.

Другу групу творців вірусів також становлять молоді люди (частіше – студенти), які ще не повністю оволоділи мистецтвом програмування. Єдина причина, що штовхає їх на написання вірусів, це комплекс неповноцінності, який компенсується комп'ютерним хуліганством. З під пера подібних ”умільців” часто виходять віруси вкрай примітивні і з великим числом помилок (“студентські” віруси). Життя подібних “вірусописьменників“ стало помітно простіше з розвитком Інтернету і появою численних веб-сайтів, орієнтованих на навчання написанню комп'ютерних вірусів. На подібних веб-ресурсах можна знайти докладні рекомендації щодо методів проникнення в систему, прийомів приховування від антивірусних програм, способів подальшого розповсюдження вірусу. Часто тут же можна знайти готові початкові тексти, в які треба всього лише внести мінімальні “авторські” зміни і відкомпілювати рекомендованим способом.

“Хуліганські” віруси останніми роками стають все менш актуальними (за винятком тих випадків, коли такі шкідливі програми викликали глобальні мережеві і поштові епідемії). На даний момент частка подібних вірусів і троянських програм займає не більше 10 % “матеріалу”, що заноситься в антивірусні бази даних. Але ті 90 %, що залишилися, набагато небезпечні, ніж просто віруси.

У подальшому багато з подібних “вірусописьмеників” потрапляють у третю, найбільш небезпечну групу, яка створює і запускає в світ “професійні” віруси. Ці ретельно продумані і відлагоджені програми створюються професійними, часто дуже талановитими програмістами. Такі віруси нерідко використовують достатньо оригінальні алгоритми проникнення в системні сфери даних, помилки в системах безпеки операційних середовищ, соціальний інжиніринг та інші хитрощі.

Окремо стоїть четверта група авторів вірусів – “дослідники“, досить кмітливі програмісти, які займаються винаходом принципово нових методів зараження, утаєння, протидії антивірусам і та ін. Вони ж придумують способи впровадження в нові операційні системи. Ці програмісти пишуть віруси не заради власне вірусів, а швидше заради дослідження потенціалу “комп'ютерної фауни”. Часто автори подібних вірусів не поширюють свої творіння, проте активно пропагують свої ідеї через численні інтернет-ресурси, присвячені створенню вірусів. При цьому небезпека таких “дослідницьких” вірусів теж вельми велика – потрапивши до рук “професіоналів” з попередньої групи ці ідеї дуже швидко з'являються в нових вірусах.

З появою і популяризацією платних інтернет-сервісів (пошта, WWW, хостинг) комп'ютерний андеграунд починає виявляти підвищену цікавість до отримання доступу в мережу за чужий рахунок, тобто за допомогою крадіжки чийогось логіну та пароля (або декількох логінів/паролів з різних уражених комп'ютерів) шляхом застосування спеціально розроблених троянських програм.

На початку 1997 року зафіксовані перші випадки створення і розповсюдження троянських програм, що крадуть паролі доступу до системи AOL. У 1998 році, з розповсюдженням інтернет-послуг в Європі і Україні, аналогічні троянські програми з'являються і для інших інтернет-сервісів. Дотепер трояни, крадучи паролі до dial-up, паролі до AOL, коди доступу до інших сервісів, становлять помітну частину щоденних “надходжень” у лабораторії антивірусних компаній всього світу.

Троянські програми даного типу, як і віруси, зазвичай створюються молодими людьми, у яких немає коштів для оплати інтернет-послуг. Характерний той факт, що у міру здешевлення інтернет-сервісів зменшується і кількість таких троянських програм.

“Дрібними злодіями” також створюються троянські програми інших типів: які крадуть реєстраційні дані і ключові файли різних програмних продуктів (часто – мережевих ігор) і використовують ресурси заражених комп'ютерів на користь свого “господаря” і т.п.

Найбільш небезпечну категорію “вірусописьменників“ становлять хакери-одинаки або групи хакерів, які усвідомлено або не усвідомлено створюють шкідливі програми з єдиною метою: одержати чужі гроші (рекламуючи що-небудь або просто крадучи їх), ресурси зараженого комп'ютера (знов-таки, заради грошей – для обслуговування спам-бізнесу або організації DoS-атак з метою подальшого шантажу).

Обслуговування рекламного і спам-бізнесу – один з основних видів діяльності таких хакерів. Для розсилки спаму ними створюються спеціалізовані троянські proxy-сервери, які потім впроваджуються в десятки тисяч комп'ютерів. Потім така мережа “зомбі-машин” надходить на чорний інтернет-ринок, де отримується спамерами. Для впровадження в операційну систему і подальшого оновлення примусової реклами створюються утиліти, що використовують відверто хакерські методи: непомітну інсталяцію в систему, різноманітні маскування (щоб ускладнити видалення рекламного софтвера), протидію антивірусним програмам.

Другим видом діяльності подібних “вірусописьменників“ є створення, розповсюдження і обслуговування троянських програм-шпигунів, спрямованих на крадіжку грошових коштів з персональних (а якщо повезе – то і з корпоративних) “електронних гаманців” або з обслуговуваних через Інтернет банківських рахунків. Троянські програми даного типу збирають інформацію про коди доступу до рахунків і пересилають її своєму “замовнику”.

Третім видом кримінальної діяльності цієї групи є інтернет-рекет, тобто організація масованої DoS-атаки на один або декілька інтернет-ресурсів з подальшою вимогою грошової винагороди за припинення атаки. Зазвичай під удар потрапляють інтернет-магазини, букмекерські контори – тобто компанії, бізнес яких безпосередньо залежить від працездатності веб-сайту компанії.

Віруси, створені цією категорією “вірусописьменників”, стають причиною численних вірусних епідемій, ініційованих для масового розповсюдження і встановлення описаних вище троянських компонентів.

Системи нав'язування електронної реклами, різні “дзвонилки” на платні телефонні номери, утиліти, що періодично пропонують користувачу відвідати ті або інші платні веб-ресурси, інші типи небажаного програмного забезпечення – вони також вимагають технічної підтримки з боку програмістів-хакерів. Дана підтримка потрібна для реалізації механізмів прихованого впровадження в систему, періодичного оновлення своїх компонентів і протидії антивірусним програмам.

Очевидно, що для вирішення даних завдань у більшості випадків також використовується робота хакерів, оскільки перераховані завдання практично збігаються з функціоналом троянських програм різних типів.

2.2.1. Класифікація програмних загроз

На рис. 2.6 показана загальна схема класифікації програмних загроз або шкідливих програм. Такі програми можна розділити на дві категорії: ті, що

потребують програму-носій, і незалежні програми. До першої категорії належить програмний код, який не може працювати незалежно від деякої реальної прикладної програми, утиліти або системної утиліти. До другої категорії належать самостійні програми, які можуть бути запущені стандартними засобами операційної системи, як будь-яка інша програма.

Хоча класифікація, наведена на рис. 2.6, і дозволяє систематизувати інформацію з цього питання, вона не дає повного опису реальної картини. Зокрема, логічні бомби і "троянські програми" можуть бути частинами вірусу або "черв'яка"

Логічні бомби – ще одним з найстаріших типів шкідливих програм, що виникли ще до появи вірусів і "черв'яків", є логічна бомба. Логічна бомба є програмним кодом, упровадженим в якусь корисну програму, який повинен "вибухнути" при виконанні певних умов. Прикладами умов, які запускають логічну бомбу, можуть бути присутність або відсутність якихось файлів, настання певного дня тижня або певної дати, ім'я конкретного користувача, що ініціював запуск додатка. В одному випадку, що став широко відомим, логічна бомба перевіряла наявність певного табельного номера співробітника (автора бомби) і спрацьовувала тоді, коли цей табельний номер був відсутній в двох підряд відомостях з нарахування зарплати. Після запуску бомба може змінювати або видаляти дані або цілі файли, викликати зависання машини або виконувати якісь інші руйнівні дії.

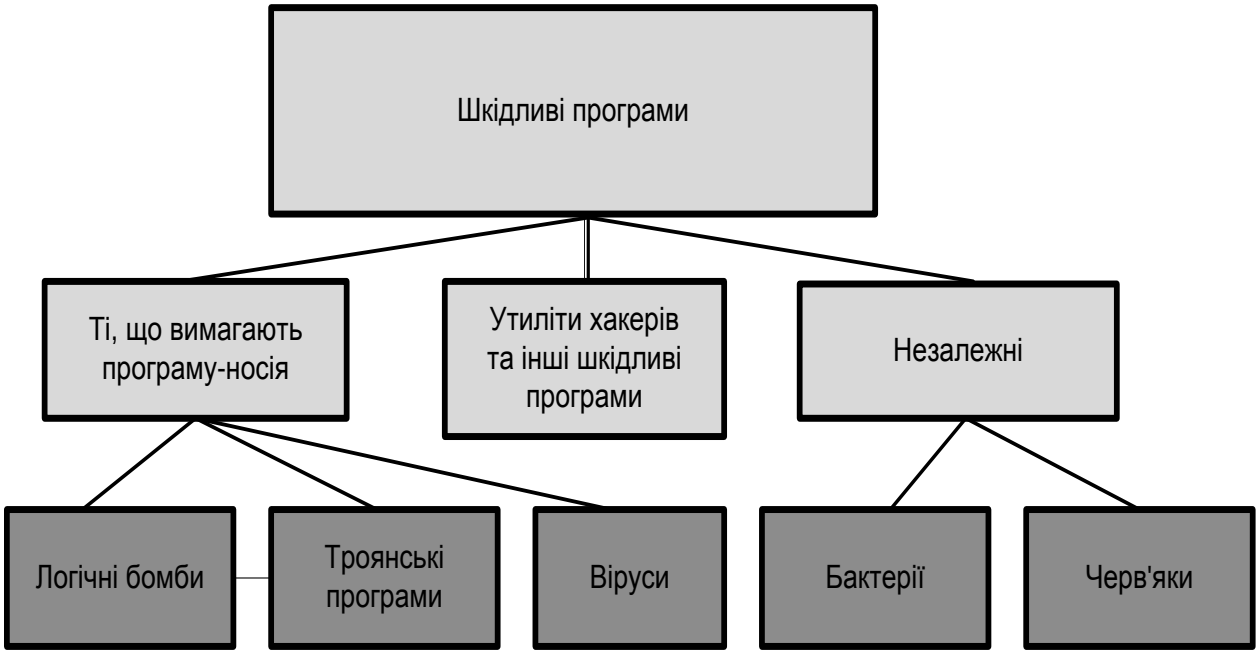


Рисунок 2.6 – Класифікація шкідливих програм

ArcBomb – “бомби” в архівах. Є архіви, спеціально оформлені так, щоб викликати нештатну поведінку архіваторів при спробі розархівувати дані – зависання або істотне уповільнення роботи комп'ютера або заповнення диска великою кількістю “порожніх” даних. Особливо небезпечні “архівні бомби” для файлових і поштових серверів, якщо на сервері використовується яка-небудь система автоматичної обробки вхідної інформації: “архівна бомба” може просто зупинити роботу сервера.

Зустрічаються три типи подібних “бомб”: некоректна назва архіву, дані, що повторюються, і однакові файли в архіві.

Некоректна назва архіву або зіпсовані дані в архіві можуть призвести до перебою в роботі конкретного архіватора або алгоритму розархівування при аналізі вмісту архіву.

Значних розмірів файл, що містить дані, які повторюються, дозволяє заархівувати такий файл в архів невеликого розміру (наприклад, 5ГБ даних упаковуються в 200КБ RAR або в 480КБ ZIP-архіву).

Величезна кількість однакових файлів в архіві також практично не впливає на розміри архіву при використанні спеціальних методів (наприклад, існують прийоми упаковки 10100 однакових файлів в 30КБ RAR або 230КБ ZIP-архів).

Троянські програми здійснюють різні несанкціоновані користувачем дії: збір інформації і її передачу зловмиснику, її руйнування або зловмисну модифікацію, порушення працездатності комп'ютера, використання ресурсів комп'ютера в незаконних цілях.

Окремі категорії троянських програм завдають збитки видаленим комп'ютерам і мережам, не порушуючи працездатності зараженого комп'ютера (наприклад, троянські програми, розроблені для масованих DoS-атак на видалені ресурси мережі).

Троянські програми розрізняються між собою за тими діями, які вони проводять на зараженому комп'ютері.

Backdoor – троянські утиліти видаленого адміністрування. Троянські програми цього класу є утилітами видаленого адміністрування комп'ютерів у мережі. За своєю функціональністю вони багато в чому нагадують різні системи адміністрування, що розробляються і поширюються фірмами-виробниками програмних продуктів.

Єдина особливість цих програм примушує класифікувати їх як шкідливі троянські програми: відсутність попередження про інсталяцію і запуск. При запуску “троян” встановлює себе в системі і потім стежить за нею, при цьому користувачу не видається ніяких повідомлень про дії ”трояна” в системі. Більш того, посилання на ”троян” може бути відсутнім у списку активних додатків. У результаті “користувач” цієї троянської програми може і не знати про її присутність в системі, тоді як його комп'ютер відкритий для видаленого керування.

Утиліти прихованого керування дозволяють робити з комп'ютером усе, що в них заклав автор: приймати або посилати файли, запускати і знищувати їх,

виводити повідомлення, стирати інформацію, перезавантажувати комп'ютер і т. ін. У результаті цей “троян” може бути використаний для виявлення і передачі конфіденційної інформації, для запуску вірусів, знищення даних і т.п. Уражені комп'ютери виявляються відкритими для зловмисних дій хакерів.

Таким чином, троянські програми даного типу є одним з найнебезпечніших видів шкідливого програмного забезпечення, оскільки в них закладена можливість найрізноманітніших зловмисних дій, властивих іншим видам троянських програм.

Окремо слід зазначити групу бекдорів, здатних розповсюджуватися по мережі і впроваджуватися в інші комп'ютери, як це роблять комп'ютерні черв'яки. Відрізняє такий “троян” від черв'яків той факт, що вони розповсюджуються по мережі не мимоволі (як черв'яки), а тільки за спеціальною командою “замовника”, що управляє даною копією троянської програми.

Trojan-PSW – це програма, яка викрадає паролі. Дана множина об'єднує троянські програми, що “крадуть” різну інформацію із зараженого комп'ютера, зазвичай – системні паролі (PSW - Password-Stealing-Ware). При запуску PSW-троян шукає системні файли, що зберігають різну конфіденційну інформацію (зазвичай номери телефонів і паролі доступу до Інтернету) і посилають її за вказаною в коді “трояна” електронною адресою або адресами.

Існує PSW-троян, який повідомляє і іншу інформацію про заражений комп'ютер, наприклад, інформацію про систему (розмір пам'яті і дискового простору, версія операційної системи), тип використовуваного поштового клієнта, IP-адресу і т.п. Деякі трояни даного типу “крадуть” реєстраційну інформацію до різного програмного забезпечення, коди доступу до мережевих ігор і інше.

Trojan-AOL – родина троянських програм, що “крадуть” коди доступу до мережі AOL (America Online). Виділені в особливу групу внаслідок своєї чисельності.

Trojan-Clicker (інтернет-клікери) – родина троянських програм, основна функція яких – організація несанкціонованих звернень до інтернет-ресурсів (зазвичай до веб-сторінок). Досягається це або посилкою відповідних команд браузеру, або заміною системних файлів, в яких вказані “стандартні” адреси інтернет-ресурсів (наприклад, файл hosts в MS Windows).

У зловмисника можуть бути такі цілі для подібних дій:

• збільшення відвідувань яких-небудь сайтів з метою збільшення показів реклами;

• організація DoS-атаки (Denial of Service) на який-небудь сервер;

• залучення потенційних жертв для зараження вірусами або троянськими програмами.

Trojan-Downloader – троянські програми, призначені для завантаження і установки на комп'ютер-жертву нових версій шкідливих програм, установки “трояна” або рекламних систем. Завантажені з Інтернету програми потім або запускаються на виконання, або реєструються “трояном” на автозавантаження

відповідно до можливостей операційної системи. Дані дії при цьому відбуваються без відома користувача.

Інформація про імена і розташування програм, що завантажуються, міститься в коді та даних трояна або викачується трояном з Інтернет-ресурсу, що “керує” (завжди з веб-сторінки).

Trojan-Dropper – це інсталятори інших шкідливих програм. Троянські програми цього класу написані в цілях прихованої інсталяції інших програм і практично завжди використовуються для “підсовування” на комп'ютер-жертву вірусів або інших троянських програм.

Даний троян зазвичай без яких-небудь повідомлень (або з помилковими повідомленнями про помилку в архіві або неправильній версії операційної системи) скидає на диск в який-небудь каталог (у корінь диска C:, у тимчасовий каталог, в каталоги Windows) інші файли і запускає їх на виконання.

Зазвичай структура цих програм така (рис 2.7):

Основний код
Файл 1
Файл 2
...

Рисунок 2.7 – Структура файлу “ троян ”

Компонент “основний код” виділяє з свого файлу решту компонентів (файл 1, файл 2, ...), записує їх на диск і відкриває їх (запускає на виконання).

Зазвичай один (або більше) з компонентів є троянськими програмами і як мінімум один компонент є “обманкою”: програмою-жартом, грою, картинкою або чимось подібним. “Обманка” повинна відвернути увагу користувача та/або продемонструвати те, що файл, який запускається, дійсно робить щось “корисне”, тоді як троянський компонент інсталюється в систему.

У результаті використання програм даного класу хакери досягають двох цілей:

- прихована інсталяція троянських програм та/або вірусів;
- захист від антивірусних програм, оскільки не всі з них в змозі перевірити всі компоненти всередині файлів цього типу.

Trojan-Proxy (троянські проксі-сервери) – це родина троянських програм, що приховано здійснюють анонімний доступ до різних інтернет-ресурсів. Зазвичай використовуються для розсилки спаму.

Trojan-Spy – це шпигунські програми. Даний троян здійснює електронне шпигунство за користувачем зараженого комп'ютера: інформація, що вводиться з клавіатури, знімки екрана, список активних додатків і дії користувача з ними зберігаються в будь-який файл на диску і періодично відправляються зловмиснику.

Троянські програми цього типу часто використовуються для крадіжки інформації користувачів різних платіжних онлайн і банківських систем.

Інші троянські програми. До даних троянів належать ті з них, які виконують інші дії, що потрапляють під визначення троянських програм, тобто руйнування або зловмисна модифікація даних, порушення працездатності комп'ютера та інше.

У даній категорії також присутні "багатоцільові" троянські програми, наприклад, ті з них, які одночасно шпигують за користувачем і надають proxy-сервіс видаленому зловмиснику.

Rootkit – програмний код або техніка, спрямована на приховування присутності в системі заданих об'єктів (процесів, файлів, ключів реєстру і т. ін.). Поняття rootkit пішло з ОС UNIX. Спочатку це поняття використовувалося для позначення набору інструментів, які вживаються для отримання прав root.

Оскільки інструменти типу rootkit на сьогодні застосовуються і на інших ОС (зокрема, на Windows), то слід визнати подібне визначення rootkit морально застарілим і таким, що не відповідає реальному положенню справ.

Для поведінки Rootkit в класифікації "Лабораторії Касперського" діють правила поглинання: Rootkit – наймолодша поведінка серед шкідливих програм. Тобто, якщо Rootkit-програма має троянську складову, то вона детектується як Trojan.

Trojan-Notifier – сповіщення про успішну атаку. Троянська програма даного типу призначена для повідомлення "замовнику" про заражений комп'ютер. При цьому на адресу "замовника" відправляється інформація про комп'ютер, наприклад, IP-адреса комп'ютера, номер відкритого порту, адреса електронної пошти і т.п. Відправка здійснюється різними способами: електронним листом, спеціально оформленим зверненням до веб-сторінки "замовника", ICQ-повідомленням.

Дані троянські програми використовуються в багатокомпонентних троянських наборах для сповіщення свого "замовника" про успішну інсталяцію троянських компонентів до системи, що атакується.

До **класичних комп'ютерних вірусів** належать програми, що поширюють свої копії в ресурсах локального комп'ютера з метою:

- подальшого запуску свого коду при будь-яких діях користувача;
- подальшого впровадження в інші ресурси комп'ютера.

На відміну від черв'яків віруси не використовують мережевих сервісів для проникнення на інші комп'ютери. Копія вірусу потрапляє на видалені комп'ютери тільки в тому випадку, якщо заражений об'єкт з якихось незалежних від функціонала вірусу причин виявляється активізованим на іншому комп'ютері, наприклад:

- при зараженні доступних дисків вірус проник у файли, розташовані на мережевому ресурсі;
- вірус скопіював себе на знімний носій або заразив файли на ньому;
- користувач відіслав електронний лист із зараженим вкладенням.

Деякі віруси містять у собі властивості інших різновидів шкідливого програмного забезпечення, наприклад бекдор-процедуру або троянський компонент для знищення інформації на диску.

Типи комп'ютерних вірусів розрізняються за такими основними ознаками:

- середовище проживання;
- спосіб зараження.

Під “середовищем проживання” розуміють системні галузі комп'ютера, операційні системи або додатки, у компоненти (файли) яких впроваджується код вірусу. Під “способом зараження” розуміють різні методи впровадження вірусного коду до об'єктів, що заражаються.

За “середовищем проживання” віруси можна розділити:

- на файлові;
- на завантажувальні;
- на макровіруси;
- на скриптові.

Файлові віруси при своєму розмноженні тим або іншим способом використовують файлову систему будь-якої (або будь-яких) ОС та:

- різними способами впроваджуються у виконувані файли (найбільш поширений тип вірусів);
- створюють файли-двійники (віруси компаньйони);
- створюють свої копії в різних каталогах;
- використовують особливості організації файлової системи (link-віруси).

Завантажувальні віруси записують себе або до завантажувального сектора диска (boot-сектор), або до сектора, що містить системний завантажувач вінчестера (Master Boot Record), або міняють вказівник на активний boot-сектор. Даний тип вірусів був достатньо поширений в 1990-тих роках, але практично зник з переходом на 32-бітові операційні системи і відмовою від використання дискет як основного способу обміну інформацією. Теоретично можлива поява завантажувальних вірусів, що заражають CD-диски і USB-флешки, але на даний момент такі віруси не виявлені.

Багато табличних і графічних редакторів, системи проектування, текстові процесори мають свої макромови для автоматизації виконання дій, що повторюються. Ці макромови часто мають складну структуру і розвинений набір команд. Макровіруси є програмами на макромовах, вбудованих у такі системи обробки даних. Для свого розмноження віруси цього класу використовують можливості макромов і за їх допомогою переносять себе з одного зараженого файлу (документа або таблиці) до іншого.

За способом зараження файлів віруси поділяються:

- на ті, що перезаписують (overwriting);
- на паразитичні (parasitic);
- на віруси-компаньйони (companion);
- на віруси-посилання (link);

- на віруси, що заражають об'єктні модулі (OBJ);
- на віруси, що заражають бібліотеки компіляторів (LIB);
- на віруси, що заражають початкові тексти програм.

Overwriting – цей метод зараження є найбільш простим: вірус записує свій код замість коду файлу, що заражається, знищуючи його вміст. Природно, що при цьому файл перестає працювати і не відновлюється. Такі віруси дуже швидко виявляють себе, оскільки операційна система і додатки досить швидко перестають працювати.

Parasitic (паразитичні файлові віруси) – це віруси, які при розповсюдженні своїх копій обов'язково змінюють вміст файлів, залишаючи самі файли при цьому повністю або частково працездатними.

Основними типами таких вірусів є віруси, що записуються на початку файлів (prepending), наприкінці файлів (appending) і в середині файлів (inserting). У свою чергу, впровадження вірусів в середину файлів відбувається різними методами – шляхом перенесення частини файлу до його кінця або копіювання свого коду до свідомо невживаних даних файлу (cavity-віруси).

Впровадження вірусу на початок файлу – це впровадження паразитичного файлового вірусу на початок файлу. Перший спосіб полягає в тому, що вірус переписує початок файлу, що заражається, до його кінця, а сам копіюється в місце, що звільнилося. Другий спосіб – вірус дописує файл, що заражається до свого тіла.

Таким чином, при запуску зараженого файлу керування першим одержує код вірусу. При цьому віруси, щоб зберегти працездатність програми, лікують заражений файл, повторно запускають його, чекають закінчення його роботи і знову записуються на його початок (іноді для цього використовується тимчасовий файл, в який записується знешкоджуваний файл), або відновлюють код програми в пам'яті комп'ютера і настроюють необхідні адреси в її “тілі” (тобто дублюють роботу ОС).

Найбільш поширеним способом впровадження вірусу у файл є **впровадження вірусу в кінець файлу**. При цьому вірус змінює початок файлу таким чином, що першими виконуваними командами програми, які містяться у файлі, є команди вірусу.

Для того щоб отримати керування при старті файлу, вірус коректує стартову адресу програми (адреса точки входу). Для цього вірус проводить необхідні зміни в заголовку файлу.

Існує декілька методів **впровадження вірусу в середину файлу**. У найбільш простому з них вірус переносить частину файлу в його кінець або “розсовує” файл і записує свій код у простір, що звільнився. Цей спосіб багато в чому аналогічний методам, перерахованим вище. Деякі віруси при цьому стискають блок файлу, що переноситься, так, що довжина файлу при зараженні не змінюється.

Другим є метод “cavity”, при якому вірус записується в свідомо невживані ділянки файлу. Вірус може бути скопійований в незадіяні ділянці

назви EXE-файлу, в “дірки” між секціями EXE-файлів або в ділянку текстових повідомлень популярних компіляторів.

Існують віруси, що заражають тільки ті файли, які містять блоки, заповнені будь-яким постійним байтом, при цьому вірус записує свій код замість такого блока.

Крім того, копіювання вірусу всередину файлу може відбутися в результаті помилки вірусу, в цьому випадку файл може бути безповоротно зіпсований.

Окремо слід зазначити досить незначну групу вірусів, що не мають “точки входу” (EPO-віруси - Entry Point Obscuring viruses) – в**іруси без точки входу**. До них належать віруси, що не змінюють адресу точки старту в заголовку EXE-файлів. Такі віруси записують команду переходу на свій код в будь-яке місце всередину файлу і одержують керування не безпосередньо при запуску зараженого файлу, а при виклику процедури, що містить код передачі керування на тіло вірусу. Причому виконуватися ця процедура може вкрай рідко (наприклад, при виведенні повідомлення про яку-небудь специфічну помилку). В результаті вірус може довгі роки “спати” усередині файлу і вискочити на свободу тільки за деяких обмежених умов.

Перед тим як записати всередину файлу команду переходу на свій код, вірусу необхідно вибрати “правильну” адресу у файлі – інакше заражений файл може виявитися зіпсованим. Відомі декілька способів, за допомогою яких віруси визначають такі адреси усередині файлів, наприклад, пошук у файлі послідовності стандартного коду заголовків процедур мов програмування (C/Pascal), дизасемблювання коду файлу або заміна адрес функцій, що імпортуються.

До категорії “**companion**” належать віруси, що не змінюють файлів, які заражаються. Алгоритм роботи цих вірусів полягає в тому, що для файлу, який заражається, створюється файл-двійник, причому при запуску зараженого файлу керування одержує саме цей двійник, тобто вірус.

До вірусів даного типу належать ті з них, які при зараженні перейменовують файл в будь-яке інше ім'я, запам'ятовують його (для подальшого запуску файлу-носія) і записують свій код на диск під ім'ям файлу, що заражається. Наприклад, файл NOTEPAD.EXE перейменовується в NOTEPAD.EXD, а вірус записується під ім'ям NOTEPAD.EXE. При запуску керування одержує код вірусу, який потім запускає оригінальний NOTEPAD.

Можливе існування й інших типів вірусів-компаньйонів, що використовують інші оригінальні ідеї або особливості інших операційних систем. Наприклад, PATH-компаньйони, які розміщують свої копії в основному каталозі Windows, розраховують, що і файли для запуску Windows в першу чергу шукатимуть саме в ньому. Даним способом самозапуску користуються також багато комп'ютерних черв'яків і троянські програми.

Існують віруси, які жодним чином не пов'язують свою присутність з виконуваним файлом. При розмножуванні вони всього лише копіюють свій код в будь-які каталоги дисків, аби ці нові копії коли-небудь запустив користувач.

Іноді ці віруси дають своїм копіям “спеціальні” імена, щоб підштовхнути користувача на запуск своєї копії, наприклад, INSTALL.EXE або WINSTART.BAT.

Деякі віруси записують свої копії до архівів (ARJ, ZIP, RAR). Інші записують команду запуску зараженого файлу у BAT-файли.

Link-віруси також не змінюють фізичного вмісту файлів, проте при запуску зараженого файлу “примушують” ОС виконати свій код. Цієї мети вони досягають модифікацією необхідних полів файлової системи.

Відомі на даний момент **завантажувальні віруси** заражають завантажувальний (boot) сектор гнучкого диска і boot-сектор або Master Boot Record (MBR) вінчестера. Принцип дії завантажувальних вірусів оснований на алгоритмах запуску операційної системи при включенні або перезавантаженні комп'ютера – після необхідних тестів установленого устаткування (пам'яті, дисків і та ін.) програма системного завантаження прочитує перший фізичний сектор завантажувального диска (A:, C: або CD-ROM залежно від параметрів, встановлених в BIOS Setup) і передає на нього керування.

При зараженні дисків завантажувальні віруси ”підставляють” свій код замість будь-якої програми і отримують керування при завантаженні системи. Отже, принцип зараження однаковий у всіх описаних вище способах: вірус “примушує” систему при її перезапуску зчитати в пам'ять і віддати керування не оригінальному коду завантажувача, а коду вірусу.

Зараження дискет відбувається єдиним способом – вірус записує свій код замість оригінального коду boot-сектора дискети. Вінчестер може заразитися трьома способами: вірус записується замість коду MBR або замість коду boot-сектора завантажувального диска (зазвичай диска C:), або модифікує адресу активного boot-сектора в таблиці розділів диска (Disk Partition Table), розташованій в MBR вінчестера.

При інфікуванні диска вірус, в більшості випадків, переносить оригінальний boot-сектор (або MBR) в будь-який інший сектор диска (наприклад, в перший вільний). Якщо довжина вірусу більше довжини сектора, то в сектор, що заражується, поміщається перша частина вірусу, решта частин розміщується в інших секторах (наприклад, у перших вільних).

Найбільшого поширення набули **макровіруси** для Microsoft Office (Word, Excel і PowerPoint), що зберігають інформацію у форматі OLE2 (Object Linking and Embedding). Віруси в інших додатках досить рідкісні.

Фізичне розташування вірусу всередині файлу MS Office залежить від його формату, який серед продуктів Microsoft є надзвичайно складним – кожен файл-документ Word Office97 або таблиця Excel являє собою послідовність блоків даних (кожний з яких також має свій формат), об'єднаних між собою за допомогою великої кількості службових даних. Унаслідок такої складності форматів файлів Word, Excel і Office97 подати розташування макровірусу у файлі можна лише таким чином (табл. 2.2).

Таблиця 2.2 – Розташування макровірусу у файлі

<table>
<tr><th>Незаражений файл-документ або таблиця</th><th>Вірус у файлі-документі або таблиці</th></tr>
<tr><td>Назва файлу</td><td>Назва файлу</td></tr>
<tr><td>Службові дані (каталоги, FAT)</td><td>Службові дані (каталоги, FAT)</td></tr>
<tr><td>Текст</td><td>Текст</td></tr>
<tr><td>Шрифти</td><td>Шрифти</td></tr>
<tr><td>Макроси (якщо є)</td><td>Макроси (якщо є)</td></tr>
<tr><td rowspan="3">Інші дані</td><td></td></tr>
<tr><td>Макроси вірусу</td></tr>
<tr><td>Інші дані</td></tr>
</table>

При роботі з документами і таблицями MS Office виконує різні дії: відкриває документ, зберігає, друкує, закриває та ін. При цьому MS Word, наприклад, шукає і виконує відповідні “вбудовані макроси” – при збереженні файлу за командою File/Save викликається макрос FileSave, при збереженні за командою File/SaveAs – FileSaveAs, при друкуванні документів – FilePrint і т. ін., якщо, звичайно, такі макроси визначені.

Існує також декілька “автомакросів”, що автоматично викликаються за різних умов. Наприклад, при відкритті документа MS Word перевіряє його на наявність макросу AutoOpen. Якщо такий макрос присутній, то Word виконує його. При закритті документа Word виконує макрос AutoClose, при запуску Word викликається макрос AutoExec, при завершенні роботи – AutoExit, при створенні нового документа – AutoNew. Автоматично (тобто без участі користувача) виконуються також макроси/функції, що асоціюються з будь-якою клавішею або моментом часу або датою, тобто MS Word/Excel викликають макрос/функцію при натисненні на будь-яку конкретну клавішу (або комбінацію клавіш) або досягши будь-якого моменту часу.

Макровіруси, що уражають файли MS Office, як правило, користуються одним з перерахованих вище прийомів – у вірусі або присутній автомакрос (автофункція), або переозначений один із стандартних системних макросів (асоційований з будь-яким пунктом меню), або макрос вірусу викликається автоматично при натисненні на будь-яку клавішу або комбінацію клавіш. Одержавши керування, макровірус переносить свій код в інші файли, зазвичай у файли, які редагуються в даний момент. Рідко макровіруси самостійно шукають інші файли на диску.

Скрипт-віруси (віруси скрипта) − це підгрупа файлових вірусів, які написані на різних мовах скрипта (VBS, JS, BAT, PHP і та ін.). Вони або заражають інші програми (командні і службові файли MS Windows або Linux) скрипта, або є частинами багатокомпонентних вірусів. Також дані віруси

можуть заражати файли інших форматів (наприклад, HTML), якщо в них можливе виконання скриптів.

Бактерії є програмами, які не ушкоджують, самі по собі, жодних файлів. Єдиною метою "бактерії" є відтворення собі подібних. Типова програма-бактерія може просто запустити дві власні копії в багатозадачному середовищі або створити два нових файли, що містять по копії оригінальної програми-"бактерії". Потім кожна з копій може створити ще дві копії і т. ін. Швидкість розмноження "бактерій" зростає експоненціально, що врешті-решт призводить до швидкого захоплення всіх ресурсів процесора, пам'яті або дискового простору, внаслідок чого відбувається відмова користувачам у доступі до цих ресурсів.

Мережевий черв'як – це різновидність шкідливої програми, яка самостійно розповсюджується комп'ютерною мережею**.** Основною ознакою, за якою типи черв'яків розрізняються між собою, є спосіб розповсюдження черв'яка – яким способом він передає свою копію на видалені комп'ютери. Іншими ознаками відмінності комп'ютерних черв'яків між собою є способи запуску копії черв'яка на комп'ютері, що заражується, методи впровадження в систему, а також поліморфізм, "стелс" та інші характеристики, властиві і іншим типам шкідливого програмного забезпечення (вірусам і троянським програмам).

Email-Worm поштові черв'яки для свого розповсюдження використовують електронну пошту. При цьому черв'як посилає або свою копію у вигляді вкладення в електронний лист, або посилання на свій файл, розташований на якому-небудь мережевому ресурсі (наприклад, URL на заражений файл, розташований на зламаному або хакерському веб-сайті).

У першому випадку код черв'яка активізується при відкритті (запуску) зараженого вкладення, в другому – при відкритті посилання на заражений файл. В обох випадках ефект однаковий – активізується код черв'яка.

Для відправки заражених повідомлень поштові черв'яки використовують різні способи. Найбільш поширені:

- пряме підключення до SMTP-сервера з використанням вбудованої до коду черв'яка поштової бібліотеки;
- використання сервісів MS Outlook;
- використання функцій Windows MAPI.

Різні методи використовуються поштовими черв'яками для пошуку поштових адрес, на які розсилатимуться заражені листи. Поштові черв'яки:

- розсилають себе за всіма адресами, виявленими в адресній книзі MS Outlook;
- прочитують адреси з адресної бази WAB;
- сканують "відповідні" файли на диску і виділяють в них рядки, що є адресами електронної пошти;
- посилають себе за всіма адресами, виявленими в листах поштової скриньки (при цьому деякі поштові черв'яки "відповідають" на знайдені в ящику листи).

Багато черв'яків використовують відразу декілька перерахованих методів. Зустрічаються також інші способи пошуку адрес електронної пошти.

IM-Worm – це черв'яки, що використовують інтернет-пейджери. Відомі комп'ютерні черв'яки даного типу використовують єдиний спосіб розповсюдження – розсилку на виявлені контакти (з листа контакту) повідомлень, що містять URL на файл, розташований на будь-якому веб-сервері. Даний прийом практично повністю повторює аналогічний спосіб розсилки, що використовується поштовими черв'яками.

Net-Worm інші мережеві черв'яки. Існують інші способи зараження віддалених комп'ютерів, наприклад:

– копіювання черв'яка на мережеві ресурси;
– проникнення черв'яка на комп'ютер через уразливість операційних систем і додатків;
– проникнення до мережевих ресурсів публічного використання;
– паразитування на інших шкідливих програмах.

Перший спосіб такий: черв'як шукає віддалені комп'ютери і копіює себе до каталогів, відкритих для читання і запису (якщо такі виявлені). При цьому черв'яки даного типу або перебирають доступні мережеві каталоги, використовуючи функції операційної системи, та/або випадково шукають комп'ютери в глобальній мережі, підключаються до них і намагаються відкрити їх диски для повного доступу.

Для проникнення іншим способом черв'яки шукають у мережі комп'ютери, на яких використовується програмне забезпечення, що містить критичні уразливості. Для зараження уразливих комп'ютерів черв'як посилає спеціально оформлений мережевий пакет або запит (експлойт уразливості), внаслідок чого код (або частина коду) черв'яка проникає на комп'ютер-жертву. Якщо мережевий пакет містить тільки частину коду черв'яка, він потім викачує основний файл і запускає його на виконання.

Окрему категорію становлять черв'яки, що використовують для свого розповсюдження веб- і FTP-сервера. Зараження відбувається в два етапи. Спочатку черв'як проникає до комп'ютера-сервера і необхідним чином модифікує службові файли сервера (наприклад, статичні веб-сторінки). Потім черв'як "чекає" відвідувачів, які запрошують інформацію із зараженого сервера (наприклад, відкривають заражену веб-сторінку), і таким чином проникає на інші комп'ютери в мережі.

Існують мережеві черв'яки, які паразитують на інших черв'яках та/або троянських програмах віддаленого адміністрування (бекдорах). Дані черв'яки використовують той факт, що багато бекдорів дозволяють за певною командою викачувати вказаний файл і запускати його на локальному диску. Те саме можливо з деякими черв'яками, що містять бекдор-процедури. Для зараження віддалених комп'ютерів дані черв'яки шукають інші комп'ютери в мережі і посилають на них команду викачування і запуску своєї копії. Якщо комп'ютер, що атакується, виявляється вже зараженим "відповідною" троянською програмою, черв'як проникає в нього і активізує свою копію.

Слід зазначити: чимало комп'ютерних черв'яків використовують більше одного способу розповсюдження своїх копій по мережах, що використовують два і більш методів атаки на віддалені комп'ютері.

Механізм роботи більшості подібних **P2P-Worm черв'яків для файлообмінних мереж** достатньо простий. Для впровадження до P2P-мережі черв'яку досить скопіювати себе до каталогу обміну файлами, який зазвичай розташовано на локальній машині. Решту роботи за розповсюдження вірусу P2P-мережа бере на себе. При пошуку файлів у мережі вона повідомить віддалених користувачів про даний файл і надасть увесь необхідний сервіс для викачування файлу із зараженого комп'ютера.

Існують складніші P2P-черв'яки, які імітують мережевий протокол конкретної файлообмінної системи і на пошукові запити відповідають позитивно. При цьому черв'як пропонує для викачування свою копію.

До категорії **утиліт хакерів та інші шкідливі програми** належать:

- утиліти автоматизації створення вірусів, черв'яків і троянських програм (конструктори);
- програмні бібліотеки, розроблені для створення шкідливого ПЗ;
- утиліти приховання коду заражених файлів від антивірусної перевірки (шифрувальники файлів);
- "злі жарти", що ускладнюють роботу з комп'ютером;
- програми, що свідомо повідомляють користувачу помилкову інформацію про свої дії в системі;
- інші програми, які тим або іншим чином навмисно завдають прямого або непрямого збитку даному або віддаленим комп'ютерам.

До **інших шкідливих програм** належать різноманітні програми, які не становлять загрози безпосередньо комп'ютеру, на якому виконуються, а розроблені для створення інших вірусів або троянських програм, організації віддалених DoS-атак сервера, зламу інших комп'ютерів і тому подібні.

DoS, DDoS мережеві атаки – це програми , які реалізують віддалені атаки на сервери, посилаючи на них численні запити, що призводить до відмови в обслуговуванні, якщо ресурси сервера, що атакується, недостатні для обробки всіх запитів, що надходять (DoS = Denial of Service).

DoS-програми реалізують атаку з одного комп'ютера з відома користувача. DDoS-програми (Distributed DoS) реалізують розподілені атаки з різних комп'ютерів, причому без відома користувача зараженого комп'ютера. Для цього DDoS-програма засилається будь-яким способом на комп'ютер "жертв-посередників" і після запуску залежно від поточної дати або за командою від "замовника" починає DoS-атаку на вказаний сервер у мережі.

Деякі комп'ютерні черв'яки містять у собі DoS-процедури, що атакують сайти, які з будь-яких причин "не полюбив" автор черв'яка. Так, 20 серпня 2001 року черв'як Codered організував успішну атаку на офіційний сайт президента США <http://www.whitehouse.gov>, а черв'як Mydoom.a 1 лютого 2004 року "вимкнув" сайт SCO <http://www.sco.com> виробника дистрибутивів UNIX.

Exploit, HackTool – це зломники віддалених комп'ютерів. Утиліти хакерів даного класу, призначені для проникнення на віддалені комп'ютери з метою подальшого керування ними (використовуючи методи троянських програм типу "backdoor") або для впровадження в зламану систему інших шкідливих програм.

Утиліти хакерів типу "exploit" при цьому використовують уразливості операційних систем або додатків, установлених на комп'ютері, що атакується.

Flooder ("засмічення" мережі) − це утиліти хакерів, які використовуються для "забивання сміттям" (непотрібними повідомленнями) каналів Інтернету – IRC-каналів, комп'ютерних пейджингових мереж, електронної пошти і та ін.

Конструктори вірусів і троянських програм – це утиліти, призначені для виготовлення нових комп'ютерних вірусів і троянів. Відомі конструктори вірусів для DOS, Windows і макровірусів. Вони дозволяють генерувати початкові тексти вірусів, об'єктні модулі та/або безпосередньо заражені файли.

Деякі конструктори забезпечені стандартним віконним інтерфейсом, де за допомогою системи меню можна вибрати тип вірусу, об'єкти, що уражаються, наявність або відсутність самошифрування, протидію відладчику, внутрішні текстові рядки, вибрати ефекти, які супроводжують роботу вірусу і тому подібне. Інші конструктори не мають інтерфейсу і прочитують інформацію про тип вірусу з конфігураційного файлу.

Nuker фатальні мережеві атаки – це утиліти, що відправляють у мережі спеціально оформлені запити на комп'ютери, що атакуються, внаслідок чого система, яка атакується, припиняє роботу. Використовують уразливості в програмному забезпеченні і операційних системах, внаслідок чого мережевий запит спеціального вигляду викликає критичну помилку в додатку, що атакується.

Bad-Joke, Hoax (злі жарти, введення користувача в оману**) –** це програми, які не завдають комп'ютеру жодної прямої шкоди, однак виводять повідомлення про те, що таку шкоду вже завдано або буде завдано за певних умов, або попереджають користувача про неіснуючу небезпеку. До "злих жартів" належать, наприклад, програми, які "лякають" користувача повідомленнями про форматування диска (хоча ніякого форматування насправді не відбувається), детектують віруси в незаражених файлах, виводять дивні вірусоподібні повідомлення і та ін., залежно від почуття гумору автора такої програми.

FileCryptor, PolyCryptor (приховання від антивірусних програм**) –** це **у**тиліти хакерів, що використовуються для шифрування інших шкідливих програм з метою приховання їх вмісту від антивірусної перевірки.

PolyEngine (поліморфні генератори**)** не є вірусами у прямому розумінні цього слова, оскільки до їх алгоритму не закладаються функції розмноження, тобто відкриття, закриття і записи у файли, читання і записи секторів і та ін. Головною функцією подібного роду програм є шифрування тіла вірусу і генерація відповідного розшифрувальника.

Зазвичай поліморфні генератори розповсюджуються їх авторами без обмежень у вигляді файлу-архіву. Основним файлом в архіві будь-якого генератора є об'єктний модуль, що містить даний генератор. У всіх генераторах, що зустрічалися, цей модуль містить зовнішню (external) функцію – виклик програми генератора.

2.2.2. Природа вірусів

Віруси здатні робити все, що можуть і звичайні програми. Єдина відмінність полягає в тому, що вірус приєднується до іншої програми і виконується потай в процесі роботи програми-носія. Під час свого впровадження вірус може виконати будь-яку операцію, наприклад, стерти файли документів і програми.

Життєвий цикл типового вірусу складається з чотирьох етапів.

1. Інкубаційний період. Вірус ніяк не виявляється. Врешті-решт вірус буде активізовано певною подією, наприклад, настанням певної дати, присутністю іншої програми або файлу, появою достатнього місця на диску. Інкубаційний період мають не всі віруси.
2. Фаза розповсюдження. Вірус поміщає свою копію до інших програм або в певні системні ділянки на диску. Тепер усі інфіковані програми міститимуть копію вірусу, кожна з яких теж повинна буде коли-небудь пройти свою фазу розповсюдження.
3. Фаза активізації. Вірус активізується для виконання функції, заради якої він створювався. Фаза активізації може бути ініційована самими різними системними подіями, наприклад, наявністю певного числа копій даного вірусу в системі.
4. Фаза виконання. Виконується функція, що міститься у вірусі. Ця функція може бути як цілком нешкідливою (наприклад, виведення повідомлення на екран), так і деструктивною (наприклад, знищення програм і файлів з даними).

Роботу більшості вірусів побудовано відповідно до архітектурних принципів конкретної операційної системи, а в деяких випадках навіть до конкретних апаратних засобів. Таким чином, в їх основі лежить використання недоліків і нюансів тих чи інших систем.

2.2.3. Структура вірусу

Вірус може бути розміщений на початку або в кінці виконуваної програми, а також вбудовуватися іншим чином. Головна ідея роботи вірусу полягає в тому, щоб при запуску програми-носія спочатку виконався код вірусу і лише потім – код самої програми. У найзагальніших рисах структуру вірусу показано на рис. 2.8.

У даному випадку код вірусу V додається на початку програми, яка інфікується, з припущенням, що точкою входу до програми є перший рядок.

Виконання інфікованої програми починається з виконання коду вірусу у такий спосіб. У першому рядку коду програми вказано команду переходу на початок основного коду вірусу. Другий рядок є спеціальним маркером, який використовується вірусом для перевірки програми на наявність зараження цим вірусом. При запуску інфікованої програми керування відразу ж передається в основне тіло вірусу. Програмний код вірусу спочатку шукає незаражені файли і заражує їх. Після цього вірус може виконати якісь інші дії, зазвичай ворожі по відношенню до системи.

Ці дії можуть мати місце при кожному запуску інфікованої програми або лише при виконанні певних умов. Закінчивши роботу, вірус передає керування програмі-носію. Якщо інфікування здійснюється достатньо швидко, то користувач навряд чи зможе помітити при запуску програми, що вона інфікована.

```
program V :=
{goto main;
1234567;
subroutine infect-executable :=
{loop:
file :5 get-random-executable-file;
if (first-line-of-file 5 1231567)
then goto loop
else prepend V to File;}
subroutine do-damage :=
{whatever damage is to be done}
subroutine trigger-pulled :=
{return true if some condition holds}
main;          main-program :=
{infect-executable;
if trigger-pulled then do-damage;
goto next;}
next:
}
```

Рисунок 2.8 – Приклад простого вірусу

Віруси, побудовані за тільки що описаним принципом, легко виявити, оскільки в результаті інфікування збільшується довжина файлу програми, що інфікується. Щоб обійти такі прості методи виявлення, вірус повинен стиснути виконуваний файл, аби інфікована та неінфікована версії файлу були однакової довжини. На рис. 2.9. показана загальна логіка роботи такого вірусу.

```
program CV :=
{goto main;
01234567;
subroutine infect-executable :=
{loop:
file := get-random-executable-file;
if (first-line-of-file = 01234567) then goto loop;
compress file;
prepend CV to file;
        }
main: main-program :=
{if ask-permission then infect-executable;
uncompress rest-of-file;
run uncompressed file;)
            }
```

Рисунок 2.9 – Логіка вірусу, що використовує стиснення

Найбільш важливі рядки вірусу пронумеровано, а відповідні операції ілюструються на рис. 2.10. Ми припускаємо, що програму P1 інфіковано вірусом CV. Коли така програма запускається на виконання, керування передається вірусу, який робить наступне.

Виявивши неінфікований файл P2, вірус спочатку стискає його, в результаті одержуючи файл P2', довжина якого менша початкової на довжину коду вірусу. Копія вірусу приєднується до початку стисненої програми. Стисла версія інфікованої оригінальної програми P1 розпаковується. Розпакована оригінальна програма виконується.

У розглянутому вище прикладі вірус тільки розмножується. Але він може містити й логічну бомбу, як описано в попередньому прикладі.

Початкове інфікування. Потрапивши до системи шляхом зараження певної програми, вірус за допомогою цієї програми може заразити декілька або всі файли даного комп'ютера. Втім розповсюдження вірусної інфекції можна не допустити, якщо не дозволити вірусу перший раз потрапити до системи. Запобігання вірусному зараженню є дуже складним завданням, оскільки вірус може опинитися в будь-якій програмі, що потрапляє до системи ззовні. Таким чином, ризику піддається кожен, хто не створив власну операційну систему і прикладне програмне забезпечення за допомогою власних апаратних засобів.

У більшості випадків початкове інфікування відбувається через носії інформації, з яких заражені програми копіюються в комп'ютер. Багато таких програм належать до розряду ігрових чи є простими, але зручними утилітами, якими співробітники користуються на своїх домашніх комп'ютерах, а потім приносять на роботу.

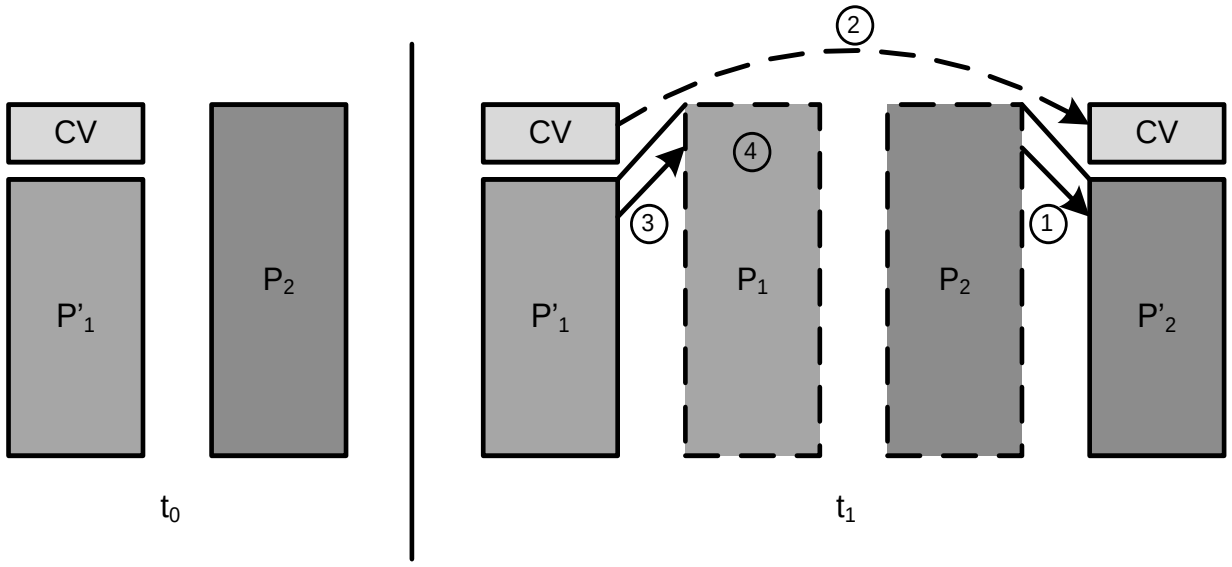


Рисунок 2.10 – Вірус, що використовує стиснення

Іноді заражені програми містяться навіть на упакованих дистрибутивних дисках розробників програмного забезпечення. Лише невелика частина випадків інфікування відбувається через мережеві з'єднання. Найчастіше джерелом інфекції є електронні дошки оголошень. Знову ж таки, користувач при цьому завантажує з мережі якусь гру або корисну утиліту, які, як з'ясовується згодом, містять віруси.

2.2.4. Антивірусний захист

Ідеальним розв'язанням проблеми вірусів є запобігання інфікуванню: не слід допускати початкового проникнення вірусу до комп'ютерної системи. На практиці цієї мети досягти важко, хоча превентивні заходи можуть знизити кількість успішно завершених вірусами атак. Для цього потрібно виконати такі вимоги:

1. Своєчасне виявлення. Якщо зараження відбулося, воно повинне бути негайно виявлене зі встановленням місцезнаходження вірусу.

2. Ідентифікація. Як тільки зараження виявлене, необхідно ідентифікувати тип вірусу, що інфікував програму.

3. Видалення. Як тільки вірус ідентифіковано, слід видалити всі сліди вірусу з інфікованих програм і відновити в їх початковому вигляді. Важливо видалити вірус з усіх інфікованих систем, щоб хвороба не розповсюджувалася далі.

Якщо вірус виявлено, але його не вдається ідентифікувати чи видалити з системи, треба видалити інфіковану програму з подальшим завантаженням її з резервної копії.

Технології розробки вірусів і антивірусів розвиваються одночасно. Перші віруси були порівняно простими фрагментами коду і могли видалятися за

допомогою простих антивірусних програм. У міру ускладнення вірусів антивірусне програмне забезпечення теж ускладнювалось.

Антивірусні програми розділяються на чотири покоління.

1. Перше покоління: звичайні сканери.
2. Друге покоління: евристичні аналізатори.
3. Третє покоління: монітори.
4. Четверте покоління: повнофункціональні системи захисту.

Антивірусні програми-сканери першого покоління для ідентифікації вірусів використовували характерні для відповідних вірусів сигнатури. Віруси могли містити "групові символи", але всі копії вірусу мали в основному одну і ту ж структуру і незмінний код. Такі програми-сканери, що використовують сигнатури, могли виявляти тільки відомі віруси. Інший тип сканерів першого покоління припускав пошук невідповідностей поточних значень довжин файлів із значеннями, збереженими в спеціальній базі даних.

Сканери другого покоління були вже не орієнтовані на конкретні сигнатури. Натомість в них почали застосовувати евристичний аналіз, за допомогою якого можна було зробити висновок про можливу наявність у програмі вірусу. Один з різновидів таких сканерів припускав пошук у програмі фрагментів коду, характерного для вірусів. Наприклад, сканер міг шукати початок циклу шифрування, використовуваного поліморфним вірусом, і намагатися відкрити ключ шифрування. Одержавши ключ, сканер міг розшифрувати тіло вірусу, ідентифікувати вірус, видалити його з програми і повернути програму в робочий стан.

Іншим підходом, що застосовувався в антивірусних програмах другого покоління, була перевірка цілісності. З кожною програмою можна пов'язати контрольну суму. Якщо вірус інфікує програму, не змінюючи при цьому контрольної суми, то перевірка цілісності обов'язково це виявить. Щоб протистояти вірусам, які при зараженні можуть міняти відповідну контрольну суму, потрібно використовувати певну функцію хешування з шифруванням. Ключ шифру зберігається окремо від програми, щоб вірус не міг згенерувати новий хеш-код і зашифрувати його. Використання функції хешування з шифруванням замість звичайної контрольної не дає вірусу можливості модифікувати програму так, щоб результат хешування після інфікування не змінювався.

Програми третього покоління були резидентними програмами, тобто виявляли віруси за виконаними діями, а не за їх структурою в інфікованій програмі. Перевагою таких програм було те, що для них не потрібно було постійно оновлювати базу даних сигнатур і евристик усе більшого числа вірусів. Натомість досить було визначити відносно невеликий набір дій, що характеризують можливі прояви вірусу.

Продукти четвертого покоління – це пакети, які об'єднують в єдине ціле всі існуючі антивірусні технології. Такий підхід, крім виконання сканування і наявності компонентів, що дозволяють реєструвати певні дії вірусів, припускає наявність засобів керування доступом, за допомогою яких можна обмежити

можливості вірусів проникати до системи і вносити зміни у файли з метою розповсюдження інфекції під виглядом оновлення.

Вірусна "гонка озброєнь" продовжується. З появою пакетів четвертого покоління з'явилася можливість побудови всеосяжної стратегії антивірусного захисту, що є органічною частиною загальних заходів забезпечення захисту комп'ютерної системи.

2.3. Спам. Методи боротьби зі спамом

Останнім часом почастішали скарги користувачів на збільшення небажаної електронної кореспонденції рекламного характеру. Такі листи називаються в мережі спамом.

2.3.1. Визначення спаму, історія його виникнення,

Термін "спам" походить від старого скетчу британської комік-групи Monty Python Flying Circus, в якому відвідувачі ресторану, що намагаються зробити замовлення, вимушені слухати хор вікінгів, який оспівує м'ясні консерви SPAM.

Стосовно нав'язливої мережевої реклами термін "спам" став вживатися кілька років тому, коли рекламні компанії почали публікувати в конференціях новин Usenet свої рекламні оголошення. На щастя передплатників таких груп новин це недовго продовжувалося, оскільки технологія Usenet передбачала будь-яку фільтрацію повідомлень, отже, адміністратори конференцій просто видаляли спам раніше, ніж він досягав великого числа адресатів. Зазнавши поразки, спамери переключилися на розсилку реклами по групах адресатів.

Спам у сучасному Інтернеті є незаконним заняттям. У законодавстві ряду країн передбачено відповідальність за діяльність подібного роду. Наприклад, у США один з найбільших провайдерів Інтернет America Online (AOL) кожного місяця порушує по декілька судових позовів до спамерів, які займаються систематичною розсилкою реклами на адреси її клієнтів.

Чим же поганий спам? Часто користувачі не звертають уваги на мережеву рекламу, видаляючи такі повідомлення з своїх поштових скриньок. Насправді шкода від таких розсилок дорого обходиться одержувачу спаму і його провайдеру. Велика кількість рекламної кореспонденції може призвести до зайвого навантаження на канали і поштові сервери провайдера, через що звичайна пошта, на яку чекають одержувачі, проходитиме значно повільніше. Спамер практично нічого не платить за те, що передає пошту. За все розплачується одержувач спаму, який оплачує своєму провайдеру час у мережі, що витрачається на отримання незапитаної кореспонденції з поштового сервера.

Для того щоб ефективно боротися зі спамом, необхідно чітко визначити, що саме мають на увазі під поняттям “спам”. Нерідко провайдери і власники мереж керуються “презумпцією винності”, відносячи до спаму практично всю

пошту, яку не запрошував одержувач. За останні півтора року експерти вивчили всі існуючі види і категорії спаму і дійшли висновку, що при віднесенні до спаму всякого небажаного або рекламного листа може спровокувати втрату ділової пошти.

“Побутові” визначення спаму як “небажаної пошти” або “незапитаної рекламної розсилки”, які можна почути від користувачів, провайдерів або власників комп'ютерних мереж, не є ефективними.

Оскільки при фільтрації спаму головне – не нашкодити одержувачу пошти, необхідно дати більш зважене визначення.

Ось найбільш точне визначення спаму: **спам – це анонімна, масова, незапитана розсилка.**

Це визначення досить добре співвідноситься зі світовою практикою і визначеннями спаму, покладеними в основу американського і європейського законодавства про спам. Крім того, це визначення можна ефективно використовувати на практиці. Пояснимо його значення.

Анонімна розсилка: ми всі страждаємо в основному саме від автоматичних розсилок з прихованою або фальсифікованою зворотною адресою. В наш час не існує спамерів, які не приховували б своєї адреси і місця розсилки.

Масова розсилка: саме масові розсилки є справжнім бізнесом для спамерів і справжньою проблемою для користувачів. Невелика розсилка, зроблена помилково людиною, що не є професійним спамером, може бути небажаною поштою, але не спамом.

Непрошена розсилка: очевидно, підписні розсилки і конференції не повинні потрапляти в категорію “спаму” (хоча умова анонімності і так значною мірою це гарантує).

Важливі також і категорії, які ми свідомо не включили у визначення спаму. Наприклад, у визначення спаму часто включають словосполучення “рекламна розсилка” або “комерційна пропозиція”. На наш погляд, це неправильно.

Річ у тому, що значна частина спаму не має на меті своєї користі від реклами. Існують розсилки політичного і агітаційного спаму, є також “добродійні” спамерські листи (що закликають допомогти нещасним). Окрему категорію становлять шахрайські листи (так звані нігерійські листи з пропозиціями отримати готівкою велику суму грошей або ті, що залучають до фінансових пірамід), а також листи, спрямовані на крадіжку паролів і номерів кредитних карт (“фішінг”). Ще бувають так звані “ланцюжкові листи”, тобто листи з проханням переслати їх знайомим (“листи щастя”) і т.п. Є також вірусні листи, що містять привабливий текст і віруси під виглядом іграшок, картинок, програм (“справжня історія Білосніжки”, “З Новим роком!” і т.п.). Всі ці листи, як правило, не можна віднести до реклами, хоча вони є очевидним спамом.

Спам і цільові комерційні пропозиції. З даного визначення випливає, що комерційна пропозиція, явно спрямована на адресу одержувача з реальною зворотною адресою, – це не спам.

Отже, не вважається спамом небажаний рекламний лист, наприклад, запрошення на семінар, відправлений особисто директору фірми. Або пропозицію гірськолижного туру зі справжньою зворотною адресою турфірми.

Комусь це може здатися дивним і нелогічним, але ми вважаємо, що це необхідно розрізнювати – і теоретично, і практично.

Такий лист також може бути небажаним і викликати роздратування. Зазначимо, що такі листи у багатьох випадках теж можна розпізнати і відфільтрувати технічно, разом із спамом. Наприклад, Kaspersky Anti-Spam має рубрики "Семінари/Конференції", "Туризм" і тому подібні.

Проте перш ніж видаляти листи даних категорій, системному адміністратору варто погоджувати політику обробки спаму з відділом маркетингу і PR. Цілком можливо, що їм потрібні подібні листи. Наприклад, працівники туристичних фірм часто з цікавістю читають туристичні пропозиції і навіть спам, а організатори семінарів і співробітники кадрових відділів хотіли б одержувати всі запрошення на семінари.

Окрім спаму і цільових комерційних пропозицій, існує ще один вид поштових повідомлень, який часто плутають зі спамом. Це **небажана пошта**. В деяких випадках незапитане і непотрібне повідомлення не є спамом.

Ось деякі приклади небажаної пошти, яку одержувач не замовляв та/або не бажає одержувати:

- Різного роду помилки: автоматичних розсильників – технічний перебій служби розсилки, запити на підтвердження підписки на розсилку або якийсь сервіс; помилки людей, наприклад, людина шукає однокурсника, а одержувач має схоже прізвище і адресу.
- Різноманітна технічна кореспонденція: повідомлення про невідправку листа й інші помилки; автоматичні повідомлення від антивірусних програм про віруси у відправленому з вашої адреси листі; екстраординарні або рутинні повідомлення від адміністраторів сервісів (наприклад, про те, що поштовий сервіс буде недоступний, або про появу вірусу і ін.). Такі листи для одержувача часто виглядають як незапитані.
- Нові можливості спілкування і бізнесу: діловий лист від приватної особи (фірми) приватній особі (фірмі). Такий лист часто може служити початком нового контракту, справи, бізнесу. Прямий лист менеджеру корпорації від рекрутингового агентства – адресу, звичайно, одержано неофіційно, сам лист справедливо трактується компанією як загроза бізнесу, в той же час такі листи дуже корисні для ринку праці і капіталу.

І, природно, особисті листи від тих, з ким одержувач ніколи раніше не переписувався: листи від старих знайомих, друзів, агітаторів (наприклад, агітація жителів району проти забруднення парку і т. п.).

Будь-який з цих листів є незапитаним, бо приймаюча сторона його явно не запрошувала. З іншого боку, викидати подібну пошту без прочитання не можна. З цього виходить, що ознаки масовості і анонімності є необхідними для розпізнавання тих, хто робить бізнес на спамі.

Політика поводження зі спамом і небажаною поштою зводиться до розподілу всіх незапитаних повідомлень, що потрапили у поштову скриньку, на такі категорії:

- спам, що має всі ознаки анонімної масової розсилки;
- цільові комерційні пропозиції;
- небажана пошта.

Спам, поза сумнівом, потрібно фільтрувати, а потім зберігати в особливих теках або поміщати в карантин, а іноді відразу видаляти – згідно з політикою компанії.

Другу і третю категорії листів також можливо розпізнавати і фільтрувати, але з ними потрібно поводитися обережніше. У компанії можуть бути різні відділи, які хотіли б одержувати різні категорії непрошеної пошти (адміністраторам потрібні повідомлення від сервісів і антивірусів, кадровикам – запрошення на семінари).

Таким чином, системний адміністратор повинен вести ретельно продуману політику обробки пошти, що включає не тільки знищення спаму, але маршрутизацію і зберігання незапитаної і навіть небажаної пошти.

2.3.2. Методи боротьби із спамом

Можна визначити **активні і пасивні методи захисту.** Пасивні методи захисту включають якісь профілактичні заходи, що дозволяють не допустити попадання вашої поштової адреси до спамера.

Проблеми з рекламними розсилками (спамом) у приватного користувача починаються в той момент, коли його email-адреса потрапляє до бази даних спамерів. Спамери знаходять email-адреси своїх жертв різними способами:

- скануючи веб-сайти;
- скануючи дошки оголошень, форуми, чати, Usenet News і так далі;
- підбираючи “легкі” адреси (jonh@, mary@, alex@, info@, sales@, support@) за словником імен і часто вживаних слів;
- підбираючи “короткі” адреси (aa@, an@, bb@, abc@) простим перебором.

Виходячи з цього, приватному користувачу можна порекомендувати такі заходи.

1. Заведіть собі дві адреси – приватну, для листування (маловідому, яку ви ніколи не публікуєте в загальнодоступних джерелах), і публічну – для публічної діяльності (форумів, чатів і так далі).

2. Адреса для листування ніколи не повинна публікуватися у відкритому доступі.

3. Адреса для листування не повинна бути легкою в запам'ятовуванні або “красивою”. Ваше ім'я або красиве слово – не підходять. Vasily.M.Pupkin-IV – підходить цілком. Чим довша адреса і чим менш вона легка для читання, тим краще.

4. Якщо потрібно повідомити свою приватну адресу (у конференції, на сайті), робіть це способом, непридатним для автоматичного прочитання складальником адрес. “Ivan-точка-Susanin-собака-mail-точка-ру” – хороший спосіб. “Ivan.Susanin at mail.ru” –набагато гірше, Ivan.Susanin@mail.ru – нікуди не годиться. Якщо йдеться про публікацію на сайті, можна опублікувати адресу у вигляді картинки.

5. Адресу для публікації потрібно наперед вважати тимчасовою. Не варто її жаліти – ви завжди можете завести нову. Як правило, спам починає приходити на неї через декілька днів після публікації. Оскільки цю адресу можуть використовувати не тільки спамери (туди приходитиме і нормальна пошта), слід його періодично переглядати. Ви можете читати пошту, що приходить на неї, раз на тиждень або раз в місяць.

Деякі інтернет-магазини, конференції, форуми і т.п. вимагають реєстрації з вказівкою працюючої електронної пошти. Іноді передані таким чином адреси потрапляють до спамерів. Далеко не завжди це злий намір, але користувачам від цього не легше.

6. При реєстраціях завжди указуйте публічну адресу. Вона все одно може вважатися втраченою. Можна на кожну реєстрацію заводити нову адресу на безкоштовній пошті – тоді ви знатимете, хто з магазинів і форумів “продав” вашу адресу спамерам.

7. Якщо спаму приходить небагато і з ним ще можна миритися, то слід дотримуватися простих правил:

- Ніколи не відповідайте спамеру. Можливо, нічого поганого не відбудеться. Але може трапитися і так, що вашу відповідь прочитає “робот” і помітить вашу адресу як “живу” – в результаті спаму приходитиме ще більше.
- Не намагайтеся скористатися посиланням “відписатися”, якщо ви не упевнені, що вона спрацює. Можливо, вас дійсно відпише даний конкретний розсильник. Але при цьому вашу адресу можуть помітити як діючу... і спаму стане більше. Дізнатися, що трапиться, можна, тільки спробувавши. Але чи хочете ви цього?

8. Якщо миритися зі спамом вже ніяк не можна, то змініть свою приватну адресу. На деякий час це допоможе

9. Якщо ви хочете все-таки мати загальновідому і загальнодоступну адресу, приготуйтеся одержувати туди сотні спам-повідомлень на добу. Якщо не пощастить – то тисячі на добу.

10. Якщо від такої адреси ви не хочете відмовлятися, то залишається остання порада:

- Використовуйте антиспам-фільтр – або на сервері, вибравши провайдера з послугою фільтрації спаму, або у себе на комп'ютері, вибравши засіб, відповідний для вашого поштового клієнта. Сучасні фільтри мають достатньо високою якістю (відсоток фільтрованого спаму у добре настроєних фільтрів досягає 95–99 %), і їх використання різко знизить гостроту проблеми.

Якщо ж пасивні методи захисту не приносять належного успіху, час займати активну позицію. Щоб ви могли вести ефективну боротьбу проти спамера, необхідно з'ясувати такі складові:

- що рекламує спамер;
- через якого провайдера йде розсилка спаму;
- справжня електронна скринька спамера.

Лінію оборони можна організувати на базі вашого ж провайдера. У практиці веба існують так звані "black lists" – списки чорних адрес, куди провайдери, а також антиспамерські організації заносять спамерів, їх повідомлення знищуються ще до надходження у вашу поштову скриньку на сервері.

2.3.3. Сучасні технології спамерів

На сьогодні розсилка спаму набула виняткових масштабів – щодоби в світі розсилаються десятки мільярдів спам-повідомлень (від 40 до 70 відсотків всієї електронної пошти). Такі масштаби вимагають істотних вкладень у технологію розсилок.

Склалися цілком стійкі технологічні ланцюжки дій спамерів:

1. Збирання і верифікація email-адрес одержувачів. Класифікація адрес за типами.
2. Підготовка “точок розсилки” – комп'ютерів, через які розсилатиметься спам.
3. Створення програмного забезпечення для розсилки.
4. Пошук клієнтів.
5. Створення рекламних оголошень для конкретної розсилки.
6. Створення розсилки.

Кожен окремий крок у цьому ланцюжку може виконуватися, незалежно від іншого.

Для розсилки спаму необхідно мати **список адрес електронної пошти** потенційних одержувачів (“спам-базу”, email database). Адреси в таких списках можуть мати додаткову інформацію:

- регіон;
- вид діяльності компанії (або інтереси користувачів);
- список адрес користувачів конкретної поштової служби (Yandex, AOL, Hotmail і т. п.) або конкретного сервісу (eBay, Paypal).

Збирання адрес здійснюється такими методами:

- підбір за словниками імен власних, “красивих слів”, частих поєднань “слово-цифра” (наприклад, jonh@, destroyer@, alex-2@);
- метод аналогій – якщо існує адреса Serg.User@hotmail.com, то цілком резонно пошукати Serg.User у доменах yahoo.com, aol.com, Paypal;
- сканування всіх доступних джерел інформації – веб-сайтів, форумів, чатів, дошок оголошень, Usenet, баз даних Whois на поєднання

слово1@слово2.слово3 (при цьому на кінці такого поєднання повинен бути домен верхнього рівня – com, ru, info і т. д.);

- крадіжка баз даних сервісів, провайдерів і т. п.;
- крадіжка персональних даних користувачів за допомогою комп'ютерних вірусів і інших шкідливих програм.

При скануванні доступних джерел інформації (спосіб 3) можна намагатися визначити "коло інтересів" користувачів даного джерела, що дає можливість одержати тематичні бази даних. У разі крадіжки даних у провайдерів достатньо часто є додаткова інформація про користувача, що теж дозволяє провести персоналізацію.

Крадіжка персональних даних користувачів – адресних книг поштових клієнтів (більшість адрес в яких – діючі) і інших персональних даних − набула поширення порівняно недавно. На жаль, масові вірусні епідемії останніх років показують, що поширеність антивірусних засобів недостатня, отже, частота використання даного способу збирання персональних даних зростатиме.

Одержані адреси потрібно верифіковувати, що здійснюється такими способами:

1. Пробна посилка повідомлення. Як правило, це повідомлення з випадковим текстом, які проходять через спам-фільтри. Аналізуючи відповідь поштового сервера (пошту прийнято або не прийнято), можна з'ясувати, чи діє кожна конкретна адреса зі списку.

2. Розміщення в текст спам-повідомлення унікального посилання на картинку, розташовану на WWW-сервері. При прочитанні листа картинку буде завантажено (у багатьох сучасних поштових програмах цю функцію блоковано), а власник сайту дізнається про доступність адреси. Метод верифікує не валідність адреси, а факт прочитання листа.

3. Посилання "відписатися" в спам-повідомленні. Якщо одержувач натискає на це гіперпосилання, то ніякої відписки не відбувається, а його адреса позначається як валідна. Метод верифікує активність одержувача.

Усі три способи верифікації не дуже хороші, відповідно, в базах даних адрес електронної пошти буде достатньо багато "мертвих" адрес.

Підготовка "точок розсилки" спаму здійснюється трьома основними способами:

- пряма розсилка з орендованих серверів;
- використання "відкритих релеїв" і "відкритих proxy" – сервісів, помилково конфігурованих їх власниками таким чином, що через них можна розсилати спам;
- прихована установка на призначених для користувача комп'ютерах програмного забезпечення, що дозволяє несанкціонований доступ до ресурсів даного комп'ютера (бекдорів).

Для розсилки спаму з орендованих серверів необхідно мати постійно поповнюваний набір цих серверів. Вони достатньо швидко потрапляють у чорні списки IP-адрес, отже, розсилати спам у такий спосіб можна тільки на тих одержувачів, поштові сервіси яких не використовують чорні списки.

Для використання відкритих сервісів необхідно постійно вести пошук таких сервісів – для цього пишуться і використовуються спеціальні програми, які швидко сканують великі ділянки адресного простору Інтернету.

Найбільшу популярність на сьогодні має установка бекдорів на комп'ютерах звичайних користувачів. Це здійснюється одним з таких способів:

• Включення троянських програм в піратське програмне забезпечення: модифікація поширюваних програм, включення троянської програми в "генератори ключів", "програми для обману провайдерів" і т.п. Достатньо часто такі програми розповсюджуються через файлообмінні мережі (eDonkey, Kazaa) або через сайти з "варезом" (warez, піратські копії програм).

• Використання слабких місць в інтернет-браузерах (в першу чергу, Microsoft Internet Explorer) – ряд версій таких програм містить помилки в перевірці прав доступу, що дозволяє розмістити на веб-сайті компоненти, які будуть непомітно для користувача викачані і виконані на його комп'ютері, після чого на комп'ютер користувача буде відкрито віддалений доступ для зловмисників. Такі програми розповсюджуються в основному через часто відвідувані сайти (перш за все, порнографічного змісту). Проте влітку 2008 року було помічено двоступінчату схему – масовий злам сайтів, що працюють під керуванням MS IIS, і модифікація сторінок на цих сайтах з включенням у них шкідливого коду, що призвело до зараження комп'ютерів користувачів, що відвідували ці сайти (звичайного змісту).

• Використання комп'ютерних вірусів, поширюваних по каналах електронної пошти і використовуючих уразливості в мережевих сервісах Microsoft Windows:

– усі великі вірусні епідемії, що сталися 2008 року, були проведені вірусами, які могли бути використані для віддаленого доступу до призначеного для користувача комп'ютера;

– інтенсивність спроб використання вразливостей Windows на сьогодні просто жахлива – підключена до Інтернету машина під керуванням стандартної Windows XP без включеного міжмережевого екрана і встановлених сервісних паків виявляється зараженою протягом декількох десятків хвилин.

Сучасні шкідливі програми є достатньо розвиненими в технічному значенні – їх автори докладають значних зусиль для ускладнення їх виявлення ззовні (наприклад, провайдером, клієнти якого розсилають спам непомітно для себе). Троянські компоненти можуть прикидатися інтернет-браузером, звертаючись на веб-сайти за інструкціями, що їм робити – займатися DoS-атакою, розсилати спам і т.п. (більш того, інструкції можуть містити вказівку про час і "місце" наступного отримання інструкцій). Інший спосіб замаскованого отримання команд полягає у використанні IRC.

З іншого боку, одне із застосувань заражених машин – це здача їх в оренду (наприклад, для розсилки спаму). Вимога "популярності" списку призводить до того, що шкідливі програми працюють за стандартними протоколами (HTTP або SOCKS proxy) з номерами портів з невеликого списку,

що дає можливість їх використання третіми особами і одночасно полегшує пошук заражених машин системними адміністраторами.

Для розсилки спаму необхідне **програмне забезпечення.** Середня спам-розсилка має сьогодні обсяг у кілька мільйонів повідомлень. Ці повідомлення потрібно розіслати за невеликий час, щоб встигнути провести розсилку до перенастроювання (або оновлення бази даних) антиспам-фільтрів.

Швидка розсилка великої кількості email-повідомлень є технологічною проблемою, розв'язання якої вимагає досить великих ресурсів. Як наслідок, на ринку є відносно невелика кількість програм, що задовольняють вимоги спамерів-професіоналів. Ці програми:

– вміють розсилати як через "відкриті сервіси" (поштові релеї, proxy), так і через заражен, призначені для користувача машини;

– можуть формувати динамічний текст листа (див. нижче розділ про формування текстів);

– достатньо точно підроблюють заголовки повідомлень − розпізнавання спаму за заголовками стає нетривіальним завданням;

– можуть відстежувати валідність баз даних email-адрес;

– можуть відстежувати статус повідомлення на кожну окрему адресу і перепосилати його через іншу "точку розсилки" у разі використання на приймальній стороні чорних списків.

Такі програми оформлені або у вигляді сервісу, доступного за передплатою, або як відчужувана (що купується) програма.

Судячи з досвіду, основний спосіб **пошуку клієнтів** – це, власне, рекламні розсилки (спам). Вони становлять основну частину всього спаму. Таким чином рекламуються, зокрема, легальні сервіси, наприклад, програми для розсилки і бази даних email-адрес.

Сьогодні проста розсилка ідентичних (або майже ідентичних) спам-повідомлень не є ефективною. Такі листи будуть виявлені фільтрами за частотністю і змістом. Тому **формування спам-повідомлень** стали індивідуальними, кожне наступне тепер відрізняється від попередніх. Основні технології "індивідуалізації" повідомлень такі: внесення випадкових текстів, "шуму", невидимих текстів. На початку або в кінці листа спамер може помістити уривок з класичного твору або просто випадковий набір слів. У HTML-повідомлення можна внести "непомітний" текст (дуже дрібним шрифтом або кольором, що збігається з кольором фону). Ці додавання ускладнюють роботу нечітких сигнатур і статистичних методів. Як міра у відповідь з'явився пошук цитат, стійкий до доповнень текстів, детальний розбір HTML й інші методи поглибленого аналізу змісту листа. У багатьох випадках можна визначити сам факт використання "спамерського трюка" і відкласифікувати повідомлення як спам, не аналізуючи його текст у деталях.

Рекламне повідомлення можна надіслати користувачу у вигляді **графічного файлу**, що вкрай ускладнить автоматичний аналіз. Як міра у відповідь з'являються способи аналізу зображень, що виділяють на них текст.

У графічне повідомлення можна внести ”шум” і отримати **графічні листи, що змінюються.** Це ускладнить аналіз листа фільтром.

Одне і те ж рекламне повідомлення складається з безлічі варіантів одного і того ж тексту. Кожен окремий лист виглядає як звичайний зв'язний текст, і лише маючи багато копій повідомлення, можна встановити факт **перефразовування текстів**. Таким чином, ефективно набудувати фільтри можна тільки після отримання істотної частини розсилки.

Ці методи підтримуються безпосередньо в програмах для розсилки, тому використання конкретного методу індивідуалізації повідомлень залежить від використовуваного програмного забезпечення.

З огляду на викладене всі основні технологічні складові бізнесу спамерів можуть бути використані незалежно один від одного, тобто має місце **розподіл праці**. В наш час працюють окремі “виробники” вірусів і троянських компонентів, автори програм для розсилки і ті, хто створює адреси. Спамери (а саме ті, хто збирає з клієнтів гроші і проводить розсилку) можуть просто орендувати необхідні їм сервіси, купувати бази даних, списки машин для розсилання. Таким чином, вхід на даний ринок є технічно доступним і відносно дешевим.

У той же час очевидним є поділ ринку на професіоналів (власників баз даних адрес, програм для розсилки або власних вірусів), які мають регулярний прибуток, і аматорів, що заробляють час від часу.

Перспективи розсилки спамів є вражаючими**.** Знаючи вартість спам-розсилки (близько 100 USD за мільйон повідомлень) і кількість повідомлень, що розсилаються в світі (десятки мільярдів у день), нескладно оцінити грошовий обіг на цьому ринку: він становить сотні мільйонів доларів на рік. Зрозуміло, що в індустрії з таким капіталом “компанії повного циклу”, які здійснюють увесь комплекс послуг на високому професійному рівні. Єдина проблема всього бізнесу є його незаконність. Збір персональних даних без відома користувача теж карається. З іншого боку, інтегрованість дає масу незаперечних технологічних переваг.

Якщо подібні вертикальні компанії ще не з'явилися, то поява їх – справа найближчого часу. Потерпілими будуть, природно, рядові отримувачі електронної пошти.

Розсилки спаму (небажаної реклами) з'явилися у середині 90-х років минулого століття – з появою великої кількості Інтернет-користувачів, що зацікавило рекламодавців. 1997 року вже почали говорити про “проблему спаму”, тоді ж з'явився перший чорний список IP-адрес спам-машин.

Розвиток методів розсилки спаму визначався удосконаленням засобів фільтрації. Як тільки один з методів розсилки починає переважати, знаходяться ефективні засоби боротьби з ним, і спамерам доводиться змінювати технологію. При цьому чим більшою проблемою є спам, тим активніше розроблюються шляхи протидії, а відтак, швидше змінюються технології спамерів, їх бізнес росте і дозволяє вкладати більше коштів в розробку.

Спам починався з **прямих розсилок** – спамери розсилали повідомлення від власного імені з власних поштових серверів. Такий спам блокується достатньо просто (за адресою поштового сервера або адресою відправника). Як тільки такі блокування стали поширеними, спамери були вимушені підробляти адреси відправників й іншу технічну інформацію.

Далі почали використовуватися **розсилки через "відкриті релеї".** Відкритий релей (open relay) – це поштовий сервер, який дозволяє пересічному користувачу відправити довільний електронний лист на будь-яку адресу. У середині 90-х років всі поштові сервери були "відкритими релеями", тому знадобилося змінювати і перенастроювати програмне забезпечення всіх поштових серверів світу. Не всі адміністратори поштових систем робили це достатньо швидко, тому з'явилися сервіси пошуку "відкритих релеїв", а потім їх списки (зокрема, основані на технології DNS списки реального часу - RBL, realtime blackhole list). Сьогодні цей метод розсилки все ще застосовується, оскільки відкриті релеї існують дотепер .

Як тільки розсилки через відкриті релеї перестали бути ефективними, спамери стали застосовувати розсилку з **розсилки з модемних пулів** (dialup-підключень), використовуючи такі можливості:

– як правило, поштовий сервер провайдера приймає пошту від своїх клієнтів і пересилає її далі;

– dialup-підключення одержує динамічну (змінну після кожного нового з'єднання) IP-адресу, таким чином, спамер може розсилати пошту з множини IP-адрес.

У відповідь провайдери стали вводити ліміти на число листів, посланих від одного користувача, з'явилися списки dialup-адрес і блокування прийому пошти з "чужих" модемних пулів.

На початку 2000-х років одночасно з розповсюдженням високошвидкісних підключень (ADSL, Cable) спамери стали використовувати слабкі місця в клієнтському устаткуванні і виконувати **розсилку з proxy-серверів**. Багато ADSL-модемів мали вбудований SOCKS-сервер або HTTP proxy (програмне забезпечення, що дозволяє здійснювати розділення Інтернет-каналу між багатьма комп'ютерами), причому доступ до них був зі всього світу без паролів і контролю доступу (для спрощення настройки кінцевим користувачем). Таким чином, можна було провести будь-яку дію (у тому числі і розсилку спаму) з IP-адреси ADSL-користувача. Оскільки таких користувачів по всьому світу – мільйони, то проблему було частково розв'язано тільки зусиллями виробників устаткування – відкриті всьому світу "посередники" впродовж останніх років до складу устаткування не входять.

У наш час основна маса розсилок проводиться з призначених для користувача комп'ютерів, на які тим або іншим способом встановлено "троянське" програмне забезпечення, що дозволяє спамерам (й іншим недобросовісним людям) здійснювати доступ до призначених для користувача машин без відома і контролю користувача. Для зламу призначених для користувача машин використовуються такі методи:

– троянські програми, які поширюються разом з піратським програмним забезпеченням по файлообмінних мережах (Kazaa, eDonkey та ін.);

– використання слабких місць в різних версіях Windows і широко розповсюдженого програмного забезпечення (в першу чергу, MSIE і MS Outlook) для установки бекдорів на призначених для користувача комп'ютерах;

– email-черв'яки останніх поколінь, які також використовуються для установки бекдорів.

За найскромнішими оцінками троянські програми встановлено на кількох мільйонах машин по всьому світу. На сьогодні ці програми достатньо хитромудрі – вони можуть оновлювати свої версії, одержувати інструкції з наперед підготовлених сайтів або каналів IRC, розсилати спам, здійснювати DDoS-атаки і т.п.

Поява засобів виявлення спаму, основаних на аналізі змісту листа (контентний аналіз), привела до еволюції **змісту спамерських листів** – їх готують так, щоб автоматичний аналіз був ускладнений. Як і у разі зміни методів розсилки, спамери вимушені боротися з антиспам-засобами.

Перші спам-повідомлення були **простими текстовими і HTML-листами.** Всім одержувачам розсилався один і той же текст. Такі повідомлення тривіально фільтруються (наприклад, за частотою повторення однакових листів).

Наступним кроком було додавання **персоналізації** (наприклад, Hello, serg! – на початку листа на адресу serg@ukr.net), що зробило всі повідомлення різними. Тепер для їх фільтрації потрібно було вишукувати рядок, що не змінюється, і заносити його до списку правил фільтра. Як метод боротьби було запропоновано нечіткі сигнатури – стійкі до невеликих змін тексту і статистичні адаптивні методи фільтрації (Байесовська фільтрація та ін.).

На початку або в кінці листа спамер може помістити уривок з класичного твору або просто **випадковий набір слів**. У HTML-повідомлення можна внести “**невидимий” текст** (дуже дрібним шрифтом або кольором, який збігається з кольором фону). Ці додатки ускладнюють роботу нечітких сигнатур і статистичних методів. У відповідь з'явився пошук цитат, стійкий до доповнень текстів, детальний розбір HTML й інші методи поглибленого аналізу змісту листа. У багатьох випадках можна визначити сам факт використання “спамерського трюку” і відкласифікувати повідомлення як спам, не аналізуючи його текст у деталях.

Рекламне повідомлення можна прислати користувачу у вигляді **графічного файлу**, що украй ускладнить автоматичний аналіз. У відповідь з'являються способи аналізу зображень, що виділяють з них текст.

Одне й те ж рекламне повідомлення складається з множини варіантів одного і того ж тексту. Кожен окремий лист виглядає як звичайний зв'язний текст, і лише маючи багато копій повідомлення, можна встановити факт **перефразовування**. Таким чином, ефективно набудувати фільтри можна тільки після отримання істотної частини розсилки.

На сьогодні широко використовуються три останні методи – далеко не всі антиспам-засоби можуть з ними нормально боротися, що дає можливість доставляти спам тим користувачам, які використовують недостатньо ефективні засоби фільтрації.

2.3.4. Тематики спаму

Слово “спам” зараз знайоме практично кожному комп'ютерному користувачу. Тенденція зростання обсягів незапитаних масових комерційних розсилок перевершила найсміливіші прогнози. За даними лінгвістичних лабораторій, у першому півріччі 2008 року рівень спаму становить 75–90 % від загального обсягу всієї вхідної кореспонденції в публічних поштових службах. Це означає, що на сьогодні спам є великомасштабною загрозою нормальному функціонуванню електронної пошти.

Спам не має мовних меж. В Інтернеті зустрічаються спамерські повідомлення практично на всіх мовах. Лідирує англійська, але частка листів у “східних кодуваннях”, тобто китайською, корейською та ін., теж достатньо велика.

Спам дуже різноманітний. В більшості випадків спам містить рекламні пропозиції, що стосуються сфери товарів і послуг.

При цьому деякі сегменти товарного ринку виявляються для спамерів важливішими, ніж інші. Це відбувається з різних причин. Наприклад, пропозиції, пов'язані з комп'ютерними технологіями (дешевий софтвер, пільговий хостінг і т. п.), швидше знайдуть відгук споживачів у середовищі користувачів електронної пошти (усі вони, як мінімум, мають комп'ютери в особистому користуванні або на роботі), ніж в інших груп споживачів. Деякі групи товарів, традиційно пропонованих у спамі, наприклад, ліки або тютюнові вироби, виготовлені або розповсюджуються з порушенням правил торгівлі і законодавчих актів (немає ліцензій, не сплачені мита і т. п.). Тому розповсюдження їх через реальні торгові мережі стає проблематичним.

Частина спаму є відвертим шахрайством. Наприклад, це може бути спробою змусити користувача переслати за невідомою адресою логін та пароль від поштової скриньки або навіть від банківського акаунта (подібний спам відомий як “фішинг”). Один з яскравих прикладів подібних листів – так звані “нігерійські листи”.

“Нігерійськими листами” у всьому світі називають повідомлення, написані від імені реальних або вигаданих осіб, найчастіше, громадян країн з нестабільною економічною ситуацією. Автор такого листа зазвичай стверджує, що він має в своєму розпорядженні мільйони доларів (наприклад, вкрадені іноземні інвестиції або гранти ООН), але вони придбані не зовсім законними способами чи зберігаються в обхід закону. Далі автор листа пояснює, що з цієї причини він не може розмістити гроші на рахунку в “своєму“ банку і що йому терміново потрібен рахунок у зарубіжному банку, куди можна перерахувати “брудні” гроші. Як винагорода за допомогу пропонується від 10 % до 30 % від

заявленої в листі суми. Ідея шахрайства полягає в тому, що довірливий користувач надасть автору листа доступ до свого рахунку. Неважко передбачити результат – всі гроші з цього рахунку будуть зняті, і користувач їх втратить.

В англомовній традиції спам, автори якого очевидно порушують законодавство, має самостійну назву – scam (на відміну від звичайного – spam).

П'ять основних тематик покривають близько 50 % усього потоку спаму в Інтернеті. Склад тематичних "лідерів" практично не змінювався за останні роки і не залежить від регіону розповсюдження спаму. Спам підкорюється законам ринку реклами, тому зміст спамерських пропозицій залежить від сезону – так, взимку спамери пропонують купити автопідігрів для сидінь автомобіля, а влітку – кондиціонери.

Лідируючі тематики спаму:

- спам "для дорослих";
- здоровий спосіб життя і медикаменти;
- комп'ютери й Інтернет;
- особисті фінанси;
- освіта.

Спам "для дорослих". До цієї тематики відносять рекламу засобів для підвищення потенції (віагра та ін.), а також для поліпшення фізичних можливостей у сексі; пропозиції відвідати порносайти, подивитися/купити порнографічні матеріали (відео та ін.). Традиційна мова таких повідомлень – англійська.

Приклади.

Істотну частку спаму "для дорослих" становлять пропозиції відвідати порносайти (рис.2.11), що містять порнографічні матеріали (графіка та ін.). З етичних міркувань, приклади такого спаму тут не наведено.

Subject: very cheap erection tool :-)
Good day! We would like to offer cheapest Viagra in the world! You can get it at: {LINK} Sincerely, Liza Stokes
Subject: i think you're gonna like watching me get off :-)
Hi...I'm Brooke..and I just got a webcam...lets have a little chat.. while you watch me get dirty .. haha;-) {LINK}

Рисунок 2.11 – Приклад спаму "Для дорослих".

Це пропозиції скинути зайву вагу, поліпшити стан шкіри, волосся; набути правильної постави, купити біологічні добавки, ліки online і т.п. є прикладом спаму "**здорового способу життя і медикаментів**" (рис. 2.12). Мова таких повідомлень, як правило, англійська.

Subject: Lose up to 19% weight. A new weightloss is here.
Hello, I have a special offer for you... WANT TO LOSE WEIGHT? The most powerful weightloss is now available without prescription. All natural Adipren720 100% Money Back Guarantée! - Lose up to 19% Total Body Weight. - Up to 300% more Weight Loss while dieting. - Loss of 20-35% abdominal Fat. - Reduction of 40-70% overall Fat under skin. - Increase metabolic rate by 76.9% without Exercise. - Burns calorized fat. - Suppresses appetite for sugar. - Boost your Confidence level and Self Esteem. Get the facts about all-natural Adipren720: {LINK}
Subject: Legal Low prices for Valium (Diazepam) (Caffeine FREE)
Rx Shopping Service Brings You our Newest Product: Your personal shopping service that legally provides Over the Counter (OTC) approved drugs from Canada and around the world. Order Valium (Diazepam) and it will be guaranteed Delivery within 7 DAYS! Do not miss out *Limited Quantity! Visit Here: {LINK}

Рисунок 2.12 – Приклад спаму "Здоровий спосіб життя і медикаменти"

До тематики "**комп'ютери і Інтернет** " належать пропозиції придбати програмне забезпечення, комп'ютерну техніку, витратні матеріали, а також пропозиції для власників сайтів (хостинг, обмін банерами і т. ін.) (рис. 2.13). Традиційні мови таких спамерських повідомлень: англійська і російська.

Subject: Huge savings on OEM Software. All brand names available now stewardess
Looking for not expensive high-quality software? We might have just what you need. Windows XP Professional 2002 ..... $50 Adobe Photoshop 7.0 ..... $60 Microsoft Office XP Professional 2002 .... $60 Corel Draw Graphics Suite 11 ..... $60 and lots more...

Рисунок 2.13 – Приклад спаму “ Комп'ютери і Інтернет”

Особисті фінанси – це пропозиції щодо страхування, зменшення кредитної заборгованості, вигідних умов позик і та ін. (рис.2.14). Переважна мова таких спамерських повідомлень – англійська.

Приклади:

Subject: Lenders Compete--You Win
Reduce your mortgage payments Interest Rates are Going Up! Give Your Family The Financial Freedom They Deserve Refinance Today & SAVE *Quick & EASY *CONFIDENTIAL *100's Of Lenders *100% FREE *Get The Lowest Rate Apply Today! {LINK} All credit will be accepted To clear your name from our database please {LINK}or use one of the optins below. Thank You Call 1-800-279-7310 Or please mail us at: 1700 E. Elliot Rd. STE3-C4 Tempe, AZ. 85283

Рисунок 2.14 – Приклад спаму “ Особисті фінанси ”

Приклад спаму **“ Освіта ” –** це пропозиції купити дипломи/атестати; реклама семінарів, тренінгів, курсів (рис. 2.15). Переважні мови таких спамерських повідомлень – англійська і російська.

Subject: get a degree from home, Mas#ters, Bachelors or PHD
Call {Phone Num.} to inquire about our degree programs. Whether you are seeking a Bachelors, Masters, Ph.D. or MBA We can provide you with the fully verifiable credentials to get your career BACK ON TRACK! No testing or coursework required Call: {Phone Num.} we are sorry if you did not want to receive this mail. To be removed from our list please call {Phone Num.}

Рисунок 2.15 – Приклад спаму “ Освіта ”

Політичний спам – це повідомлення, що формують громадську думку в період виборів й інших політичних кампаній (рис. 2.16). В Інтернеті політичний спам частіше використовувався як засіб “чорного PR”, тобто розсилки виявлялися “фальшивими”, сфабрикованими анонімно і проводилися з метою скомпрометувати ту особу або той політичний рух, від імені якого нібито йшла розсилка. В основному користувачів обурює спосіб реклами – несанкціонована анонімна розсилка (спам), після чого обурення переноситься на тих, від імені кого проведена розсилка. У результаті страждає імідж партії або політичного діяча.

Subject: <без темы>
голосуй за Иванова, ЦИК продалась США

Рисунок 2.16 – Приклад спаму “ Політичний спам ”

До пропозиції антиспамерського програмного забезпечення належать пропозиції купити антиспамерське програмне забезпечення. Достатньо часто на рекламованому сайті намагаються продати “зламане” антиспамерське програмне забезпечення відомих фірм (рис. 2.17). Буває й так, що, стежачи за посиланням, користувач знаходить не антиспамерську програму, а вірус, який намагається атакувати його комп'ютер.

Subject: Join the thousands who are now sp@m-free
FORGET SPAM BLOCKERS! Get SMART Spam Control That Always Delivers The Email You Want! Finally, we discovered the ultimate solution that is guaranteed to stop all spam without losing any of your important email! This revolutionary advanced technology also protects you 100% against ALL email-born viruses both known and unknown. We didn't believe it either until we actually tried it. So you be the judge and see for yourself. {LINK}

Рисунок 2.17 – Приклад спаму “ Пропозиція антиспамерського програмного забезпечення ”

Спамери постійно розширюють спектр своїх пропозицій і шукають нові способи залучення довірливих користувачів. Набір спамерських тематик поступово розширюється. Хоча частка “нових” тематик у потоках спаму поки незначна, деякі з них заслуговують окремої згадки.

Чому спам – загроза для бізнесу? Причина вибухового зростання спам-розсилок у тому, що спамер, нав’язуючи інтернет-аудиторії певну інформацію, практично нічого не платить, проте це дорого коштує одержувачу спаму та його провайдеру. Спам-повідомлення намагаються бути максимально схожими на звичайну пошту – аби їх читали.

Чим шкідливий спам:

- зниженням продуктивності компанії;
- неконтрольованою втратою важливих повідомлень при ручному чищенні електронної пошти;
- загрозою стабільності роботи поштових серверів;
- небезпечним змістом: вірусами, троянами, забороненими законом матеріалами. В наш час 75–90 % вхідних повідомлень є спамом. Це означає, що три чверті свого дискового простору і процесорної потужності поштовий сервіс витрачає зараз на обслуговування бізнесу спамерів!;
- паразитним трафіком;
- для провайдерів – витратами на службу підтримки, конфліктами з клієнтами.

Таким чином, спам несе значні ризики для бізнесу компаній та Інтернет-провайдерів.

Список джерел інформації

1. Величко В.В. Передача данных в сетях мобильной связи третьего поколения / В.В.Величко. – М.: Радио и связь, Горячая линия-Телеком, 2005. – 332 с.
2. ДСТУ 3396.2-97. Захист інформації. Технічний захист інформації. Терміни та визначення. – К.:УкрНДІССІ, 1997. – 11 с.
3. Спесивцев А.В. Защита информации в персональных ЭВМ / А.В. Спесивцев, В.А. Вегнер, А.Ю. Крутяков и др. – М.: Радио и связь, 1993. – 192 с.
4. Левин М. Библиотека хакера 2. Книга 1 / М.Левин. – М.: Майор, 2003. – 640 с.
5. Левин М. Библиотека хакера 2. Книга 2 / М.Левин. – М.: Майор, 2003. – 688 с.
6. Мафтик С. Механизмы защиты в сетях ЭВМ: пер. с англ./ С. Мафтик – М.: Мир, 1993. – 216 с.
7. Мельников В.В. Защита информации в компьютерных системах / В.В. Мельников. – М.: Финансы и статистика. Электроинформ, 1997. – 367 с.
8. Романец Ю.В. Защита информации в компьютерных системах и сетях / П.А. Тимофеев, В.Ф. Шаньгин. – М.: Радио и связь, 2001. – 376 с.
9. Столингс. В. Криптография и защита сетей. Принципы и практика / В. Столингс. – М.: «Вильямс», 2001. – 672 с.
10. Толковый словарь по вычислительным системам / под ред. И. Иллингуорта. – М.: Машиностроение, 1989. – 568 с.
11. Якубайтис. Э.А. Открытые информационные сети / Э.А. Якубайтис. – М.: Радио и связь, 1991.– 208 с.
12. Якубайтис. Э.А. Информатика – электроника – сети / Э.А. Якубайтис. – М.: Финансы и статистика, 1989. – 198 с.

Контрольні запитання

1. Дайте класифікацію порушників.
2. Перерахуйте методи отримання паролів.
3. Які методи протидії порушникам існують?
4. Проілюструйте схему використання паролів у системі UNIX.
5. Які стратегії вибору паролів?
6. Перерахуйте причини необхідності виявлення порушників.
7. Проілюструйте профілі поведінки порушників.
8. Які підходи до розв’язання проблеми виявлення порушників?
9. Проілюструйте розподілену систему виявлення порушень.

10. Наведіть класифікацію програмних загроз.

11. Наведіть приклади троянських програм, логічних бомб та класичних вірусів.

12. Який життєвий цикл типового вірусу?

13. Опишіть просту процедуру роботи вірусу, що використовує стиснення.

14. Які антивірусні програми ви знаєте?

15. Дайте визначення спаму.

16. У чому відмінність спаму від цільових комерційних пропозицій і небажаної пошти?

17. Перерахуйте методи боротьби зі спамом.

3. СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ В КОМП'ЮТЕРНІЙ МЕРЕЖІ ІНТЕРНЕТ

3.1. Безпека інформації в мережі ІНТЕРНЕТ

Всесвітню інформаційну мережу Інтернет було створено для поліпшення взаємодії між науковими організаціями, що виконували роботи на користь уряду США. Протягом 80-х років минулого століття до Інтернету підключилися освітні установи, державні організації, різні американські та іноземні фірми. У теперішній час доступність цієї мережі практично в усіх країнах світу дозволила отримувати інформацію мільйонам користувачів.

Проте проблема інформаційної безпеки при підключенні до мережі Інтернет залишається актуальною. Організації, які ігнорують цю проблему, піддають себе ризику бути атакованими зловмисниками і стати стартовим майданчиком при атаках на інші мережі. Навіть ті організації, які піклуються про безпеку, мають бути пильними через ймовірну появу нових вразливих місць у мережевому програмному забезпеченні.

Деякі проблеми безпеки в Інтернеті – це результат наявності вразливих місць через помилки при проектуванні або конфігурації систем і засобів керування доступом. Нижче розглянемо основні загрози безпеки комп'ютерним мережам і причини, що призводять до їх виникнення, а також системи, засоби і протоколи захисту інформації в мережі Інтернет.

3.1.1. Найбільш поширені сервіси, що забезпечуються мережею Інтернет

Існує ряд сервісів, що забезпечуються комп'ютерними системами і мережею Інтернет. Найбільш поширеним сервісом є електронна пошта, реалізована на базі протоколу SMTP (Простий Протокол Передачі Листів). Також широко використовуються TELNET (емуляція віддаленого термінала) і FTP (протокол передачі файлів). Крім них, існує ряд сервісів і протоколів для віддаленого друку, надання віддаленого доступу до файлів і дисків, роботи з розподіленими базами даних і організації інших інформаційних сервісів. Короткий список найбільш поширених сервісів наведено у табл. 3.1 та 3.2.

На рис. 3.1 проілюстровано типову мережеву архітектуру організації, що використовує Інтернет для прийому і передачі корпоративної інформації. Більшість комерційних й інших організацій використовують Інтернет-сервіси для того, щоб забезпечити поліпшену взаємодію між підрозділами організації або між організацією та її клієнтами, або для скорочення витрат на автоматизацію комерційної діяльності. У цих випадках дуже важливо

враховувати вимоги до безпеки інформації, яка циркулює в мережі, оскільки один інцидент з безпекою може перекреслити будь-які фінансові зиски, що їх надає з'єднання з мережею Інтернет.

Таблиця 3.1 – Короткий список найбільш поширених сервісів

Сервіс	Протокол	Послуги захисту інформації	Системи захисту
Електронна пошта	**SMTP –** основний протокол передачі пошти, використовується для прийому і передачі електронної пошти	Автентифікація, конфіденційність даних. Керування доступом. Цілісність даних. Причетність	PGP, S/MIME Брандмауер
Система віддаленого доступу	**TELNET** – використовується для підключення до віддалених систем, приєднаних до мережі, застосовує базові можливості по емуляції термінала	Автентифікація. Конфіденційність даних. Керування доступом. Цілісність даних. Причетність	SOCS, Брандмауер
Міжсистемний файловий обмін	**FTP** – протокол передачі файлів, використовується для прийому або передачі файлів між системами в мережі	Автентифікація, конфіденційність даних. Керування доступом. Цілісність даних. Причетність	SSL/TLS, SOCS, Брандмауер
Служба мережевих імен	**DNS** – використовується TELNET, FTP і іншими сервісами для трансляції імен хостів на IP адреси	Автентифікація. Конфіденційність даних. Керування доступом. Цілісність даних. Причетність	Брандмауер

Таблиця 3.2 – Короткий список найбільш поширених інформаційних сервісів

gopher	Засіб пошуку і перегляду інформації за допомогою системи меню, яке може забезпечити дружній інтерфейс до інших інформаційних сервісів
WAIS	Глобальний інформаційний сервіс, використовується для індексування і пошуку в базах даних файлів
WWW/http	Всесвітня павутина, об'єднання FTP, gopher, WAIS і інших інформаційних сервісів, що використовує протокол передачі гіпертексту (http), і програми Netscape, Microsoft Internet Explorer і Mosaic як клієнтські програми
NFS	Мережева файлова система, дозволяє системам спільно використовувати директорії й диски, при цьому видалена директорія або диск здаються такими, що знаходяться на локальній машині
NIS	Мережеві інформаційні сервіси, дозволяють декільком системам спільно використовувати бази даних, наприклад файл паролів, для централізованого керування ними
Система X Windows	Графічна віконна середа і набір прикладних бібліотек, використовуваних на робочих станціях
rlogin, rsh та інші r-сервіси	Реалізують концепцію хостів, які довіряють один одному, дозволяючи виконувати команди на інших комп'ютерах, не вводячи пароль

Стисло розглянемо основні послуги, що забезпечуються зв'язком з Інтернетом, і засоби безпеки, з допомогою яких здійснюється захист інформаційних сервісів.

У табл. 3.3 подано відповідність між наявними засобами безпеки і Інтернет-сервісами, які використовують організації. Деякі із засобів (комерційне усунення наслідків інцидентів) забезпечують безпеку для всіх сервісів. У таких випадках знак стоїть напроти тих сервісів, для яких даний засіб необхідний.

У теперішній час комерційна діяльність все більше вимагає **віддаленого доступу** до своїх інформаційних систем. Це може пояснюватися необхідністю доступу співробітників у відрядженнях до своєї електронної поштової скриньки або необхідністю для продавців віддаленого введення замовлень на продукцію. За своєю природою віддалений доступ до комп'ютерних систем призводить до появи нових вразливих місць через введення додаткових точок доступу в мережі.

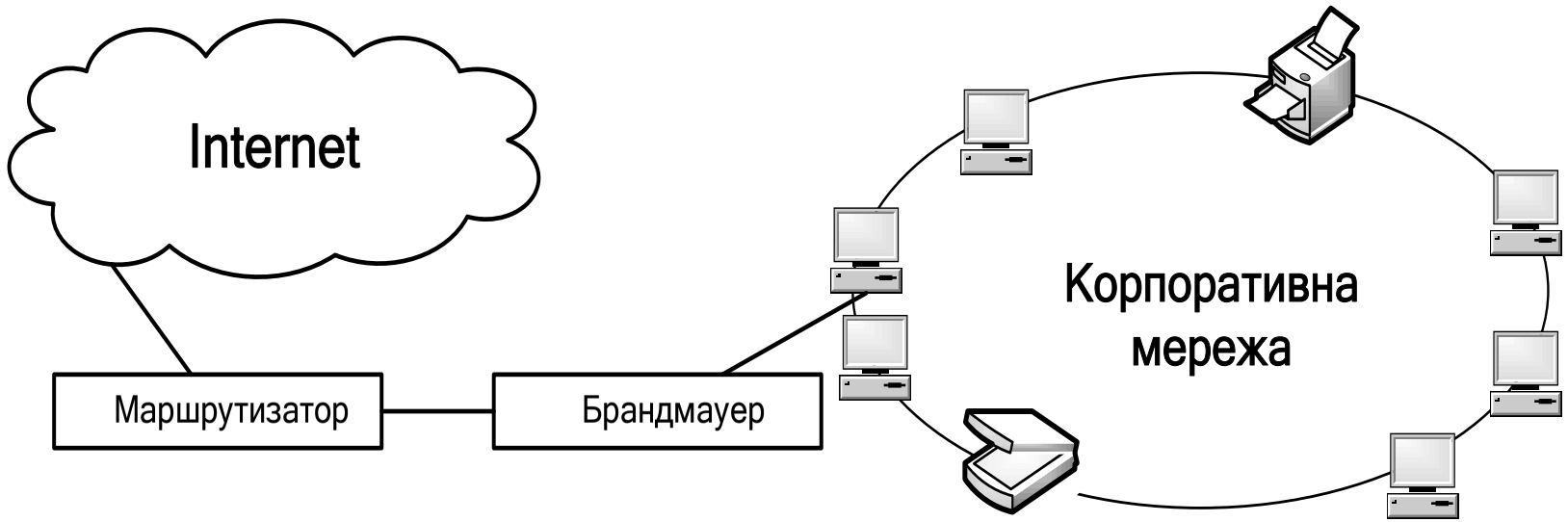


Рисунок 3.1 – Приклад архітектури організації, підключеної до мережі Інтернет

Таблиця 3.3 – Використання засобів безпеки для захисту сервісів

Сервіс	Ідентифікація і автентифікація	Керування доступом	Брандмауер	Засоби контролю програм, що імпортуються	Шифрування	Архітектура	Комерційне усунення наслідків інцидентів	Організаційні заходи
Віддалений доступ	X	X	X		X			X
Електронна пошта	X			X	X			X
Публікація інформації		X	X			X		X
Дослідження		X	X	X		X		X
Електронна комерція	X	X	X	X	X	X	X	X
Постійна доступність						X		X
Легкість використання						X		X

Існує три основні режими віддаленого доступу:

Віддалений доступ до сервісу – при цьому режимі доступ зазвичай обмежується віддаленим доступом до одного сервісу, як правило, пошти. Такі продукти як Lotus Notes і cc:Mail, підтримують віддалений доступ до цих продуктів без надання доступу до будь-яких інших мережевих сервісів. Цей режим найбезпечніший – бо кількість уразливих місць обмежена.

Віддалене керування дозволяє віддаленому користувачу керувати персональним комп'ютером, фізично розташованим у корпоративній мережі організації. Це може бути спеціальна комп'ютерна система або звичайний комп'ютер, що стоїть на робочому місці користувача. Віддалений комп'ютер використовується лише як клавіатура або дисплей. Віддалене керування обмежує віддалених користувачів доступом до програм, запущених на корпоративному комп'ютері, що є плюсом з погляду безпеки. Деякі продукти спільного віддаленого доступу декількох користувачів підтримують також добрий аудит і протоколювання дій користувачів.

При роботі в режимі ***віддаленого вузла мережі*** віддалений комп'ютер з'єднується з сервером віддаленого доступу, який призначає віддаленому комп'ютеру мережеву адресу.

Всі працюючі програми знаходяться на віддаленому комп'ютері разом з локальною пам'яттю. Режим віддаленого вузла надає віддаленим користувачам доступ до всіх мережевих сервісів, якщо тільки не використовуються програми керування доступом. Режим віддаленого вузла став найпопулярнішою формою віддаленого доступу, але його використання призводить до появи найвищого рівня уразливості корпоративних систем. Ці форми віддаленого доступу можуть бути реалізовані за допомогою комутованого з'єднання, сеансів ***telneta*** або використання програмних продуктів, що забезпечують віддалений доступ.

Віддалений доступ по телефонних каналах став найпопулярнішою формою віддаленого доступу. При цьому віддалений комп'ютер використовує аналоговий модем для дозвону до модему в режимі автовідповіді. Методи забезпечення безпеки цього з'єднання включають:

- **обмеження кола осіб, які знають номери телефонів, до яких підключені модеми,** – цей підхід уразливий до автоматизованих атак, простих програм, що використовують модеми з автодозвоном для сканування блоків телефонних номерів і виявлення номерів з модемами;
- **використання пар ім'я-пароль**, оскільки атакуючому потрібно підключитися до телефонної лінії, щоб дізнатися ім'я і пароль. Комутовані з'єднання менш уразливі до атак за допомогою перехоплювачів паролів, які роблять часто використовувані паролі практично безглуздими в глобальних мережах. Проте використання перехоплювачів паролів на внутрішніх мережах, вибір паролем легко вгадуваних слів, соціальна інженерія роблять отримання паролів легким. Також часто зловмисники представляються співробітниками відділів технічної підтримки, для того щоб дізнатися у законних користувачів їх паролі.

Існує багато методів **посиленої автентифікації**, які можуть бути використані для забезпечення або заміни звичайних паролів. Ці методи включають:

- **модеми із зворотним дзвінком** – ці пристрої вимагають від користувача ввести ім'я і пароль при встановленні з'єднання. Потім корпоративний модем розриває з'єднання і шукає авторизований номер телефону для даного користувача. Після цього він сам дзвонить за цим номером і встановлює з'єднання. Користувач знову вводить ім'я і пароль для встановлення з'єднання. Цей підхід уразливий до атак перепризначення дзвінка і не забезпечує гнучкості, потрібної для встановлення з'єднання ,приміром, з готелями і аеропортами.

•**одноразові паролі** – системи запит-відповідь на основі криптографії, такі, як S/Key Bellcore, і SecurID Security Dynamics. Вони вимагають, щоб користувач використовував програмний або апаратний генератор паролів. Ці пристрої створюють унікальний пароль для кожного сеансу і вимагають, щоб користувач знав ім'я і пароль, а також володів генератором паролів. Хоча цей метод є уразливим до атак повтору сеансу, проте саме він забезпечує мінімально допустимий рівень безпеки для більшості з'єднань з віддаленим доступом.

•**автентифікація на основі місцезнаходження віддаленого користувача** – нові технології автентифікації використовують системи глобального позиціонування для реалізації автентифікації на основі місцезнаходження. Якщо користувач здійснює з'єднання не з авторизованого місця, то доступ для нього заборонять. Ця технологія все ще ненадійна, дорога і складна у використанні. Але вона може виявитися доречною для багатьох додатків. Слабкі місця пов'язані з ймовірністю того, що порушник може дати фальшиву інформацію про своє місцезнаходження. Більшість підходів використовують криптографію для захисту від такої форми атаки.

Електронна пошта. Хоча мультимедійна форма WWW привертає основну увагу, саме електронна пошта сприяла розширенню Інтернету. Використання електронної пошти для здійснення важливих ділових взаємодій росте швидкими темпами. Хоча електронна пошта є найдоступнішим способом взаємодії з клієнтами, діловими партнерами, з її використанням пов'язано ряд проблем безпеки:

•адреси електронної пошти в Інтернеті легко підробити. Не можна сказати напевно, хто написав і послав електронний лист, спираючись лише на його адресу. Електронні листи можуть бути легко модифіковані. Стандартний SMTP-лист не містить засобів перевірки цілісності;

•існує ряд місць, де зміст листа може бути прочитаний тими, кому він не призначений. Електронний лист швидше схожий на листівку – його можуть прочитати на кожній проміжній станції.

Звичайно, немає гарантій доставки електронного листа. Хоча деякі поштові системи надають можливість одержати повідомлення про доставку, часто такі повідомлення означають лише те, що поштовий сервер одержувача (не обов'язково сам користувач) отримав повідомлення.

Ці вразливі місця визначають політику певної організації щодо використання електронної пошти в комерційних цілях.

Інтернет значно спрощує надання інформації громадянам, клієнтам організації і діловим партнерам, принаймні, тим, хто має комп'ютер, підключений до Інтернету. Проте будь-яке використання засобів електронної публікації інформації, яке зменшує число запитів інформації по телефону або

поштою, може допомогти організації скоротити витрати на цю статтю і принести додаткові прибутки. Існують два види публікації інформації– примусова та ініційована читачем. Передплата на журнали – приклад примусової публікації – інформація регулярно надсилається передплатникам. Газетні кіоски – приклад публікації за ініціативою читача – читачі мають захотіти одержати інформацію.

Електронний еквівалент примусової публікації – створення списку розсилки, в якому інформація посилається всім передплатникам цього списку. Зазвичай для відправки повідомлень до списку розсилки, а також для включення до списку розсилки або видалення з нього використовується спеціальна програма – сервер списку розсилки. Сервери списків розсилки відносно безпечні в тому відношенні, що користувачам не потрібно мати з'єднання з мережею організації, яка публікує інформацію для отримання даних. Проте вони мають декілька вразливих місць:

Програма-сервер розсилки обробляє дані від користувачів для включення їх до списку, для видалення зі списку або отримання інформації про сам список. Існує багато безкоштовних програм-серверів розсилки, причому ряд з них не перевіряє до кінця введені користувачем дані. Зловмисники можуть надіслати команди Unix або дуже великі рядки для того, щоб викликати непередбачені режими роботи або зробити проникнення шляхом переповнювання буфера.

При неправильній конфігурації сервер розсилки може зробити видимим список передплатників для кожного передплатника. Це може дати інформацію для проведення подальшої атаки "відмова в обслуговуванні" або "соціальна інженерія".

Існує два електронних еквіваленти публікації за ініціативою читача, що використовуються в Інтернеті, – FTP-сервери і WWW-сервери.

Для надання FTP-сервісу в Інтернеті практично все, що потрібно, – це комп'ютер і підключення до Інтернету. FTP-сервери можуть бути встановлені на будь-який комп'ютер, що працює під керуванням Unix (Linux), а також на ті, що працюють під керуванням Microsoft Windows.

Існує багато комерційних і безкоштовних версій програм для FTP, часто як частина стека TCP/IP, що забезпечує драйвери для підключення до сервісів Інтернет. Вони можуть дозволяти здійснювати повністю анонімний доступ, де не потрібні паролі або вони можуть бути налаштовані так, що вимагатимуть для отримання доступу до сервісу пари ім'я-пароль. FTP-сервери забезпечують простий інтерфейс, що нагадує стандартний інтерфейс Unix для роботи з файлами. Користувачі можуть одержати файли, а потім переглянути їх або виконати, якщо у них є відповідні програми.

Якщо FTP-сервер неправильно налаштовано, його можливо додавати до *будь-якого* файлу на комп'ютері-сервері або навіть в мережі, приєднаній до

цього комп'ютера. FTP-сервери повинні обмежувати доступ окремим деревом піддиректорій і, за необхідності, вимагати ім'я та пароль .

Впродовж попередніх років спростерегається колосальне зростання Всесвітньої павутини (WWW). Веб-сервери надають дешевий спосіб публікації інформації, що містить текст, графіку, або навіть аудіо- і відео- записів. Використання стандартів Гіпертекстової мови розпізнавання документів (HTML) і Протоколу передачі гіпертекстової інформації (HTTP) дозволяє користувачам легко копіювати і переглядати Web-документи, незважаючи на різноманітність клієнтських платформ.

Хоча розробка професійного веб-сайту складна і не дешева, будь-який комп'ютер, підключений до Інтернету, може виступити в ролі веб-сервера. Існує велике число як комерційних, так і безкоштовних програм WWW-сервера для різних операційних систем. Останні версії операційних систем включають програми, необхідні для організації веб-сервера, а також корисні програми-майстри, що дозволяють автоматизувати установку і конфігурацію.

Як і FTP-сервери, WWW-сервери можуть призводити до появи серйозних уразливих місць у корпоративних мережах при неправильній конфігурації. Дивіться розділ WWW для докладнішої інформації про політику безпеки для WWW- і FTP-серверів.

Проведення досліджень за допомогою Інтернету включає використання клієнтських програм для пошуку і читання інформації з віддалених серверів. Клієнтські програми можуть бути таких типів:

FTP-програми дозволяють підключатися до віддалених систем, переглядати файлові структури на них і завантажувати звідти файли;

Gopher-програми розроблені в університеті Міннесоти і надають графічний інтерфейс для перегляду і завантаження файлів у стилі FTP;

World Wide Web – веб-браузери набагато зручніші для читання інформації в Інтернеті. Програма-клієнт для проглядання інформації у WWW зазвичай має можливості FTP-клієнта і Gopher-клієнта, крім розширених можливостей мультимедіа.

Існує ряд інформаційних систем на основі Інтернету, які вимагають використання спеціальної програми-клієнта, а не веб-браузера. Як правило, вони надають доступ до інформації, захищеної авторськими правами, або інформації, що зберігається в реляційних базах даних.

Основний ризик, пов'язаний з використанням Інтернету для досліджень, – це можливість занесення вірусів. З появою "макро-вірусів", які містяться в стандартних документах текстових процесорів, завантаження документів стало таким же ризикованим, як і завантаження виконуваних файлів. Крім того, доступність "додатків-помічників" і завантажуваних "аплетів" для забезпечення

відображення файлів спеціальних форматів (таких, як PostScript) збільшила ризик троянських коней.

Іншим ризиком є сліди, що програми-клієнти залишають при перегляданні вмісту інформаційних серверів в Інтернеті. Більшість серверів мають можливість записувати, як мінімум, IP-адресу клієнта, а веб-сервери можуть одержати частину інформації про тип використовуваного браузера, останній відвіданий сайт і адресу електронної пошти, що використовувалась у браузері, а також іншу критичну інформацію. Крім цього, програма веб-сервера може зберігати файл "візиток" (cookie) на комп'ютері, де знаходиться браузер, що дозволяє серверу відстежувати візити клієнта на сервер і відвідувані ним сегменти.

Розглянемо найбільш поширені на сьогодні види **електронної комерції.**

Традиційною послугою у галузі електронної торгівлі є продаж інформації, наприклад передплата на бази даних, що функціонують у режимі on-line.

За кордоном останнім часом стає все більш популярною концепція "електронних магазинів". Зазвичай електронним магазином є Web-site, в якому подано оперативний каталог товарів, віртуальний "кошик" покупця, на який "збираються" товари, а також засоби оплати за поданням номера кредитної картки мережею Інтернет або телефоном. Оперативні каталоги товарів можуть оновлюватися у міру зміни пропозицій продукції або для віддзеркалення сезонних акцій стимулювання попиту. Відправка товарів покупцям здійснюється поштою або, у разі покупки електронних товарів (наприклад, програмного забезпечення), по каналах електронної пошти чи безпосередньо через Web-site мережею Інтернет.

Розвивається такий вид електронної комерції, як електронні банки. Серед основних переваг електронних банків можна виділити відносно низьку собівартість організації (не потрібно орендувати престижні будівлі, не потрібні сховища цінностей і т.д.) і широке охоплення клієнтів (потенційним клієнтом електронного банку може стати практично будь-який користувач Інтернет). Тому електронний банк може надавати клієнтам вигідніші, ніж у звичайного банку, відсотки, а також ширший спектр банківських послуг за нижчу платню. Природно, що електронний банк має власні системи безпеки і захисту електронної інформації, наприклад спеціальні карти-генератори випадкових паролів, що синхронізуються з паролем на банківському сервері (це дозволяє створювати унікальний пароль при кожному зверненні клієнта до банківського сервера). Інший, менш дорогий підхід пов'язаний з використанням персональних смарт-карт, які також дозволяють генерувати сесійні (сеансові) ключі.

Певну затримку в розвитку електронної торгівлі було обумовлено відсутністю надійної системи захисту. Поки платіжна інформація передається по відкритих мережах з мінімальними обережностями або зовсім без них. Це сприяє автоматизованому шахрайству (наприклад, використанню фільтрів для всіх повідомлень, що проходять через будь-яку мережу, з метою добування номерів рахунків кредитних карток з потоку даних), а також шахрайства "заради пустощів", що характерно для деяких хакерів.

Електронний обмін даними (EDI) – це термін, що не потребує пояснень. Простою його формою є обмін інформацією між двома суб'єктами (званими в EDI торговими партнерами) бізнесу в стандартизованому форматі. Базовою одиницею обміну є набір транзакцій, який, загалом, відповідає стандартному документу бізнесу, такому, як платіжне доручення або накладна на товар. За допомогою стандартів, основу яких становлять X.9 і UN/EDIFACT, ділове співтовариство розробило групу стандартних наборів транзакцій.

Кожен набір транзакцій складається з великого числа елементів даних, потрібних для даного документа бізнесу, кожен з яких має свій формат і місце серед інших елементів даних. Якщо транзакція містить більше ніж одну транзакцію (декілька платіжних доручень до однієї фірми), то групі транзакцій передаватиме заголовок функціональної групи, а за групою йтиме закінчення функціональної групи.

Компанії стали використовувати EDI, щоб зменшити час і витрати на контакти з постачальниками. Так, в автомобільній промисловості великі компанії вимагали від постачальників використовувати EDI для всіх транзакцій, що дозволило зберегти величезну кількість паперу, значно прискорити процес постачання і скоротити зусилля на підтримку актуальності баз даних.

Інтернет може забезпечити можливості взаємодії, необхідні для EDI, за низькими цінами. Але Інтернет не забезпечує сервісів безпеки (цілісності, конфіденційності, контролю учасників взаємодії), потрібних для EDI. Як і електронна пошта в Інтернеті, транзакції EDI уразливі до модифікації, компрометації або знищення при посилці через Інтернет. Використання криптографії для забезпечення необхідних сервісів безпеки змінило положення – багато компаній і урядових агентств перейшли на EDI в Інтернеті.

Інформаційні транзакції або забезпечення інформацією – основний елемент комерції, який високо цінується. Інформація в комерції може мати декілька форм:

1. Статичні дані: історична інформація, карти і т. ін.
2. Корпоративна інформація: телефонні номери, адреси, структура організації і т. ін.
3. Інформація про продукцію або послуги.

4. Платна інформація: новини, періодичні видання, доступ до баз даних і т. ін.

Використання Інтернету для надання цих сервісів набагато дешевше, ніж використання факсу, телефону або звичайної пошти. Потенційні клієнти можуть шукати і одержувати інформацію в потрібному їм темпі, і це не вимагатиме додаткових витрат на службу технічного супроводу.

Зазвичай такі інформаційні сервіси використовують WWW як базовий механізм для надання інформації. Цілісність і доступність інформації, що надається, є головною проблемою забезпечення безпеки, що вимагає застосування засобів безпеки і створення політики безпеки.

Комп'ютери і мережі протягом тривалого часу використовуються для обробки **фінансових транзакцій**. Переказ грошей з рахунку на рахунок в електронному вигляді використовується для транзакцій банк-банк, а банкомати використовуються для операцій клієнт-банк. Авторизація покупця за допомогою кредитних карток виконується за допомогою телефонних ліній і мереж передачі даних.

Для підтримки безпеки цих транзакцій вони виконуються за допомогою приватних мереж або шифруються. Використання приватних глобальних мереж (як і для EDI) обмежувало можливості взаємодії. І лише Інтернет забезпечив доступність фінансових транзакцій.

Існує три основних класи фінансових транзакцій і п'ять основних типів механізмів платежу (табл. 3.4).

Таблиця 3.4 – Платежі і фінансові транзакції

Класи транзакцій	Типи механізмів платежу				
	Готівка	Чек	Дебіт	Кредит	Електронний переказ фондів
Компанія-компанія		Основний			Допоміжний
Компанія-клієнт	Основний	Допоміжний	Допоміжний	Допоміжний	
Клієнт-клієнт	Основний	Допоміжний			

Використання Інтернету для виконання цих типів транзакцій дозволяє замінити уявлення або показ готівки, чеків, кредитних карт їх електронними еквівалентами.

Зараз існує ряд конкуруючих підходів для реалізації *електронних грошей*, реалізація яких ще знаходиться на стадії розробки. Всі ці методи використовують криптографію для створення безпечних цифрових "гаманців", в яких зберігається цифрова готівка. Передача електронних грошей необов'язково вимагає участі фінансових установ як проміжної стадії.

Банківська індустрія розробляє стандарт для *електронних чеків*, що означає: інформація, яка міститься у фізичних чеках, повинна подаватися в електронному повідомленні. Електронні чеки завжди вимагають участі фінансових установ при їх передачі.

Дебітові карти – смарт-карти і карти з пам'яттю можуть зберігати електронні гроші за допомогою кількох способів. Кожна транзакція дебетує певну кількість, доки карта не пустіє. Карти з пам'яттю не вимагають використання фінансових установ.

Кредитні карти – основні гравці в індустрії кредитних карт (Visa, Master Card і American Express) розробили стандарт для виконання транзакцій з кредитними картами по глобальних мережах. Відомий під назвою Безпечні електронні транзакції (Secure Electronic Transactions), цей стандарт визначає триетапні транзакції між клієнтом, продавцем і власником дебіту кредитної карти, зазвичай, банком. Транзакції електронних кредитних карт, що використовують SET, завжди вимагають участі фінансової установи.

Електронний переклад фондів (EFT) використовує криптографію для забезпечення безпеки перекладу фондів між банками і іншими фінансовими установами. Клієнти можуть авторизувати банки на посилку і прийом платежів за допомогою EFT для клієнта.

Кожна з цих форм електронних фінансових транзакцій включає використання криптографії для забезпечення цілісності, конфіденційності, автентифікації і контролю учасників взаємодії.

У міру того як Інтернет стає дедалі важливішим у виконанні ділової повсякденної діяльності, до засобів забезпечення безпеки з'єднання з Інтернетом все частіше ставляться вимоги безперервності роботи. Ці вимоги часто суттєво впливають на політику безпеки, вимагаючи компромісних рішень між вартістю дублюючих комплектів і вартістю тривалої роботи без засобів забезпечення безпеки.

Простим прикладом є брандмауер. Брандмауер може виявитися критичним місцем, адже якщо він вийде з ладу, зв'язок з Інтернетом буде унеможливлено на час усунення аварії. Якщо тимчасова втрата зв'язку з Інтернетом особливо не впливає на діяльність організації, політика може

визначати, що робота з Інтернетом припиняється до тих пір, доки не буде відновлений брандмауер. Для організацій з низьким рівнем ризику політика може дозволяти відключати брандмауер і працювати з Інтернетом без нього на час аварії. Проте, якщо зв'язок з Інтернетом важливий або організація має високий рівень ризику, політика може вимагати використання брандмауера з гарячим або холодним резервом. Завдання організації визначають, яке рішення буде ухвалено.

Для дуже великих організацій продуктивність може також диктувати використання декількох засобів безпеки, таких, як брандмауери і сервери автентифікації. Наприклад, організації, що забезпечують діяльність декількох тисяч зовнішніх користувачів в Інтернеті, можуть потребувати декілька з'єднань з Інтернетом класу Т1, що в свою чергу зажадає використання кількох брандмауерів. Організації з кількома тисячами внутрішніх користувачів, що мають тенденцію з'єднуватися з системою в один і той же час (вранці, увечері і т.ін.), можуть потребувати кілька серверів автентифікації для того, щоб час підключення був у допустимих межах.

Основними способами задоволення вимог постійної доступності є такі.

- **Планування ресурсів.** Помічено цікавий феномен – як тільки брандмауер встановлено, користувачі починають скаржитися, що з'єднання з Інтернетом стало повільнішим. Правильно вибрані засоби безпеки, такі, як брандмауер, зазвичай не є найвужчим місцем у системі. Але важливо детальне планування виділення ресурсів, оскільки засоби безпеки, які сильно зменшують продуктивність роботи, швидко відключатимуться. Дані зі специфікацій брандмауерів повинні ділитися навпіл при моделюванні потрібної продуктивності, а продуктивність критичних засобів забезпечення безпеки повинна перевірятися і настроюватися в тестовій мережі.
- **Надмірність** – для всіх організацій, окрім тих, що мають низький рівень ризику, необхідний резервний брандмауер у гарячому резерві. Аналогічно використання серверів автентифікації або серверів безпечного віддаленого доступу вимагає можливості швидко перемикатися на резервний сервер. Синхронізація – ось головне питання при використанні резервних серверів безпеки, адже всі оновлення, резервні копії і модифікації повинні проводитися на обох системах одночасно.
- **Відновлення** – коли блок, що вийшов з ладу, відновлено, потрібно здійснити ретельний контроль його конфігурації. Необхідно проаналізувати конфігурацію програм і устаткування, аби гарантувати працездатність усіх необхідних продуктів і виявити, чи виключені непотрібні сервіси, які могли додатися у процесі відновлення. Будь-які налагоджувальні можливості, що використалися для тестування, мають бути видалені або відключені.

• **Легкість використання.** Склад користувачів багатьох систем, підключених до Інтернету, може бути вельми різноманітним – від секретарів до вчених, від новачків до досвідчених користувачів. Часто вимогою бізнесу є доступність усіх додатків для середнього користувача. Цю вимогу важко оцінити, але з погляду безпеки часто воно перекладається так: "Якщо засіб безпеки стає перешкодою людям при виконанні ними своєї роботи, ви повинні відключити такий засіб". Двома складовими елементами легкості використання є зменшення числа разів, коли користувач повинен автентифікуватися в системі, і розробка інтерфейсу користувачів із засобами безпеки, такими, що відповідають рівню або запитам користувачів системи.

• **Одноразова реєстрація.** Для виконання повсякденних завдань користувачу може знадобитися реєстрація на великому числі комп'ютерів і мереж. Часто кожна система вимагає від користувача введення імені і пароля. Оскільки запам'ятовування великого числа паролів для користувачів є важким, це призводить до того, що паролі пишуться на папері (часто на моніторах ПЕОМ) або забуваються. Іншою реакцією користувача є використання одного і того ж пароля на всіх комп'ютерах. Проте різні системи можуть мати різні правила для паролів або мати різні періоди перевірки коректності пароля, що може знову привести користувачів до записування кількох паролів на папері.

Системи з однією автентифікацією на початку процесу роблять використання декількох паролів прозорим для користувача. Це реалізується такими способами:

Деякі системи створюють скрипти, що містять пари ім'я-пароль і команди входу у віддалені системи. Це позбавляє користувача клопоту, але завантажує обслуговуючий персонал, якому потрібна підтримка скриптів. Такі скрипти часто вимагають безпечного зберігання, їх неавторизоване використання може дати доступ до всіх систем, на яких зареєстрований користувач.

Інший підхід базується на Kerberos і використовує криптографію для передачі привілеїв користувача мережі або сервера, до якого користувачу потрібен доступ. Ці системи вимагають створення і роботи серверів привілеїв, а також інтеграції цієї технології до кожної системи, до якої повинен мати доступ користувач.

• **Розробка призначеного для користувача інтерфейсу.** Розробка призначеного для користувача інтерфейсу для засобів забезпечення безпеки в Інтернеті повинна бути узгоджена з інтерфейсом інших додатків, які регулярно використовуються користувачами.

Коли засоби безпеки отримуються або вбудовані в додатки, призначений для користувача інтерфейс знаходиться поза зоною контролю організації. Але для засобів, розроблених в організації, важливо, аби інтерфейс був зручний для користувача, а не для співробітника служби безпеки.

3.1.2. Основні принципи забезпечення безпеки в мережі Інтернет, захист за допомогою брандмауерів.

При підключенні мережі організації до мережі Інтернет необхідно прийняти ряд певних організаційно-технічних заходів щодо її захисту.

При побудові захисту слід виходити з того, що будь-який захист ускладнює використання системи, яка захищається, за прямим призначенням обмежує функціональні можливості, використовує обчислювальні і трудові ресурси, вимагає фінансових витрат на створення та експлуатацію. Чим вищий захист, тим більш дорогою у створенні та обслуговуванні стає система і тим менш зручною для безпосередніх користувачів. Тому, захищаючи мережу, слід виходити з доцільної вартості захисту. Тобто витрати на захист повинні бути пропорційні цінності ресурсу, що захищається.

Існує ряд основних принципів, що дозволяють організувати досить безпечне підключення до мережі Інтернет порівняно простими засобами.

Мабуть, основним загальновизнаним засобом такого захисту є міжмережевий екран **(брандмауер)**. Міжмережевий екран встановлюється між мережею, що захищається, і мережею Інтернет, і виконує роль мережевого фільтра. Він настроюється так, щоб пропускати допустимий трафік від користувачів мережі, що захищається, до служб Інтернет і назад, і обмежити трафік з боку Інтернет у мережу, що захищається тільки необхідними службами, наприклад: smtp, dns, ntp.

Допустимість того або іншого трафіку визначається мережевим адміністратором відповідно до політики інформаційної безпеки організації. Наприклад, може бути дозволений доступ з частини комп'ютерів мережі, що захищається, до web і ftp-серверів Інтернет і двонаправлений доступ між Інтернет і поштовим сервером мережі, що захищається, але заборонені всі інші протоколи і напрями трафіку.

З огляду на те, що міжмережевий екран фізично розташовується на місці мережевого шлюзу (маршрутизатора), логічно є доцільним об’єднати їх функції в одному пристрої. Це дозволяє одним засобом захистити й локальну мережу і, безпосередньо, сам шлюз. Така опція передбачена для маршрутизаторів компанії Cisco Systems (називається Firewall Feature Set). Проте дане правило є необов'язковим і міжмережевий екран може бути подано окремим пристроєм.

У простому випадку виконання функцій міжмережевого екрана можна організувати за допомогою мережевого фільтра на основі листів доступу (access-lists). Листи доступу визначають правила, за якими дозволяється або забороняється проходження трафіку з певними ознаками від одного мережевого інтерфейсу маршрутизатора до іншого усередині самого маршрутизатора. IP-адреса або діапазон IP-адрес джерела і приймача, тип протоколу, номер

порту призначення або відправлення, ряд інших службових ознак IP-пакета можуть використовуватися як ознаки (рис. 3.2).

Відмінність і недолік листів доступу в порівнянні зі справжнім міжмережевим екраном полягає в тому, що вони дозволяють створити статичний односторонній фільтр, тоді як мережеве з'єднання є динамічним процесом. Листи доступу не дозволяють контролювати параметри IP-пакета, залежні від попередніх пакетів. Звідси виникає складність застосування листів доступу для тонкої настройки фільтрації трафіку в точній відповідності з прийнятою політикою безпеки. Зокрема, з цієї причини листи доступу не в змозі захистити від такого різновиду мережевої атаки, як "крадіжка з'єднання" або "хай-джекинг".

У Firewall Feature Set указані проблеми розв'язуються за допомогою того, що він відстежує кожне мережеве з'єднання окремо і контролює весь процес у динаміці. При встановленні нового TCP-сеансу міжмережевий екран створює для нього новий процес, який контролює правильність з'єднання до самого моменту його завершення. При цьому кожен пакет, що приходить на транспортному рівні, перевіряється на відповідність попередньому, а всі "підозрілі" пакети вибраковуються.

Таким чином, стає можливим застосування фільтра доступу внутрішнього комп'ютера до зовнішньої мережі, що не дозволяє зовнішньому комп'ютеру самостійно звернутися до внутрішнього. Іншими словами, в налаштуваннях міжмережевого екрана задаються правила для проходження трафіку від одного інтерфейсу до іншого, для кожного напряму і кожного тракту окремо. Якщо правило вирішує проходження IP-пакета від інтерфейсу внутрішньої мережі до Інтернет - інтерфейсу, то на підставі такого пакета формується логічний тунель у маршрутизаторі, через який вже можуть пройти у відповідь пакети від зовнішнього одержувача. Як тільки з'єднання переривається або вичерпується час очікування, тунель закривається, і звернення ззовні до внутрішнього комп'ютера будуть виключені. З цієї ж причини екран не пропустить пакети у зворотному напрямку, якщо ініціатором з'єднання є зовнішній комп'ютер. Крім того, міжмережевий екран, на відміну від листів доступу, може контролювати зміст IP-пакетів у полі даних і відбраковувати пакети, що містять потенційно-небезпечні коди, наприклад java-аплети. Існують міжмережеві екрани, здатні виявити в IP-пакетах ознаки відомих мережевих атак і перервати таке з'єднання, але це вже достатньо дорогі системи.

Другою цеглинкою забезпечення захищеності мережі є "заміна мережевої адреси" – (Network Address Translation), або NAT. Це заміна в IP-пакеті реальної адреси комп'ютера внутрішньої мережі на будь-яку іншу задану адресу при посиланні її до зовнішньої мережі. Таким чином, для внутрішньої мережі стає можливим використання діапазонів адрес, які не вживаються в мережі

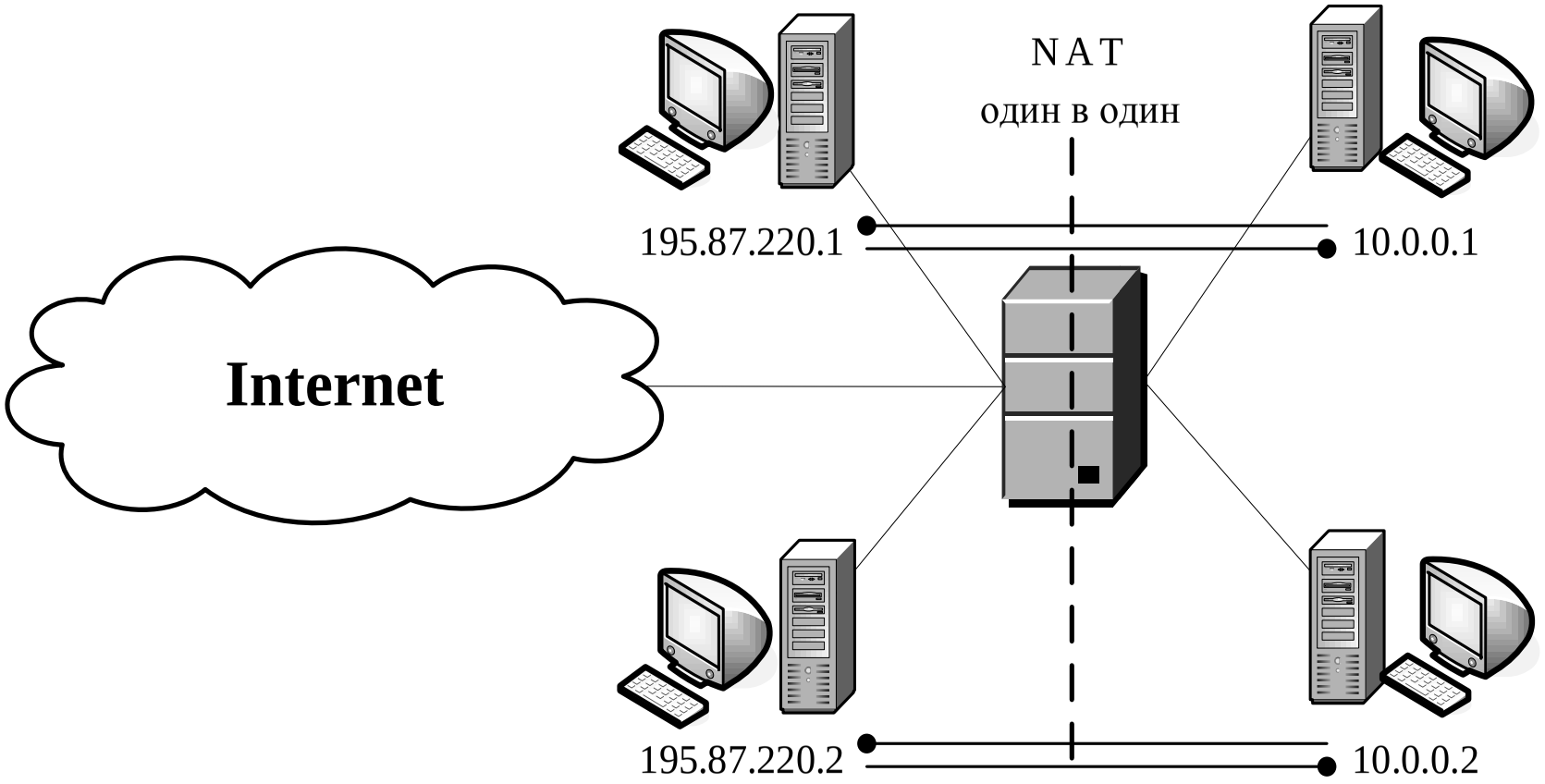


Рисунок 3.2 – Приклад побудови мережі на основі листів доступу

Інтернет (10.0.0.0-10.255.255.255). Це дозволяє запобігти прямому зверненню ззовні до внутрішніх комп'ютерів і приховати структуру мережі, що захищається. Існує кілька різновидів NAT. Найбільш простою і найбільш дешевою з точки зору захисту є трансляція фіксованої внутрішньої адреси у фіксованій зовнішній. При цьому зловмисник безперешкодно “бачить” такий комп'ютер в зовнішній мережі, оскільки йому однозначно відповідає певна зовнішня адреса. Проте вона необхідна при організації сервера, до якого потрібно забезпечити доступ ззовні.

Друга форма NAT – це трансляція групи внутрішніх адрес до однієї зовнішньої. При цьому всі внутрішні комп'ютери можуть працювати з мережею Інтернет одночасно, а маршрутизатор розрізняє, кому яку відповідь перетранслювати за службовими даними TCP-з'єднання. У зовнішній мережі створюється враження, що до неї звертається тільки один комп'ютер. Така заміна істотно ускладнює життя зловмиснику, оскільки повністю приховує внутрішні комп'ютери і перешкоджає “вирахуванню” жертви. Зловмисник, навіть побачивши звернення, що випливають з внутрішньої мережі, не зможе визначити, з якого комп'ютера вони виходять (рис. 3.3).

Крім того, це виключає можливість ініціативного звернення ззовні до внутрішнього комп'ютера, оскільки для маршрутизатора в цьому випадку відсутнє правило прив'язки зовнішньої адреси до внутрішньої.

Третя форма NAT – використання для заміни внутрішніх адрес пулу виділених адрес. Тобто внутрішній комп'ютер, виходячи в Інтернет, одержує вільну в даний момент адресу з пулу. При цьому адреси підміняються динамічно і кожне нове TCP-з'єднання може бути встановлено з іншою IP-адресою. Це також створює додаткові труднощі зловмиснику, оскільки позбавляє його можливості атакувати будь-який внутрішній комп'ютер прицільно (рис. 3.4).

Загалом, сказане відносно другої форми NAT є справедливим і для третьої форми. Якщо запит надходить ззовні, то маршрутизатор не в змозі зв'язати адресу з пулу з адресою в мережі, що захищається. Тому такий запит не досягне мети.

Демілітаризована зона. Як правило, організації потрібно мати у себе деякі мережеві ресурси, до яких відкрито доступ з мережі Інтернет. Зазвичай це поштовий, dns-і web-сервери. Механізм їх роботи припускає можливість вільного або майже необмеженого звернення з мережі Інтернет. Відповідно, ймовірність їх зламу вища, ніж решти комп'ютерів мережі.

З цієї причини розміщувати їх усередині зони, що захищається, недоцільно з погляду безпеки, оскільки у разі зламу вони можуть стати воротами для атаки внутрішніх комп'ютерів.

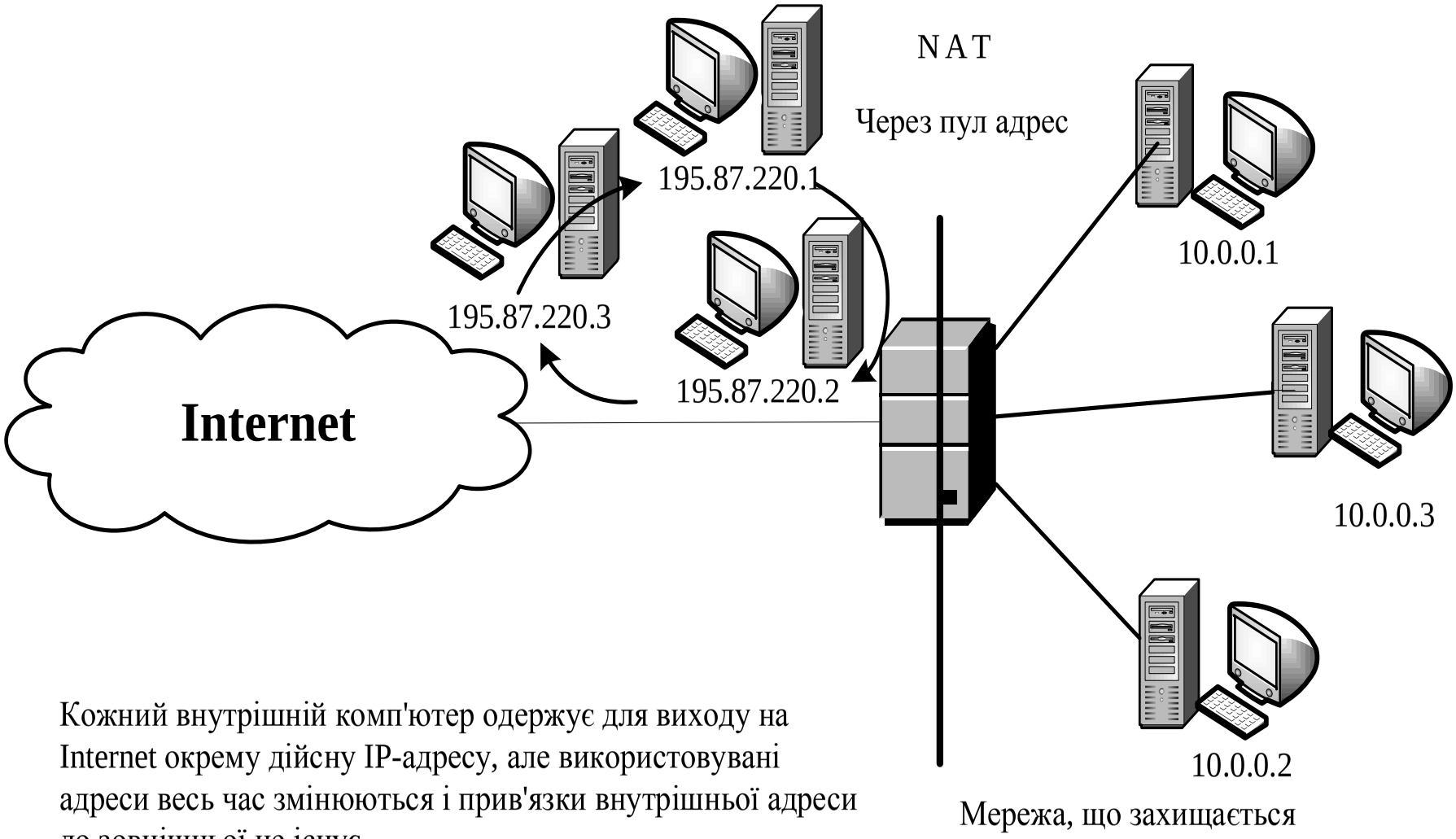


Рисунок 3.4 – Приклад побудови мережі з використанням для заміни внутрішніх адрес пулу виділених адрес

Для мінімізації ризику і збереження функціональності такі сервери встановлюють за основним шлюзом мережі, але перед міжмережевим екраном, що забезпечує захист внутрішніх комп'ютерів. Логічну зону їх розміщення називають “демілітаризованою зоною” (рис. 3.5).

З рис. 3.5 видно, що ніщо не заважає встановити другий Firewall на основному шлюзі мережі. Це є логічним рішенням і дозволяє одночасно підвищити рівень захисту внутрішньої мережі і захистити сервери демілітаризованої зони. При правильному налагодженні обох міжмережевих екранів зловмиснику буде вже набагато важче дістатися до внутрішньої мережі організації.

Наявність другого міжмережевого екрана (рис 3.6.) дещо ускладнює конфігурацію мережевого устаткування і настройку роботи всіх елементів мережі. Для додаткового підвищення захищеності можна використовувати Firewall’и різних виробників. Тоді, якщо в одному з них буде виявлено уразливість, інший не дозволить зловмиснику безперешкодно проникнути до мережі, як це мало б місце при використанні Firewall’ов одного типу.

Тут слід особливо підкреслити, що для унеможливлення зловмисного втручання мережевий доступ до шлюзів і міжмережевих екранів має бути відключений. З погляду безпеки пристрої, що охороняють мережу, повинні конфігуруватися і адмініструватися тільки через консольний порт локально.

Використання **proxy-сервера** також підвищує рівень захищеності мережі, оскільки виключає необхідність прямого виходу в Інтернет комп'ютерів-користувачів. При цьому також стає можливим більш суворий контроль за даними в IP-пакетах на рівні мережевих додатків. Proxy-сервер працює як посередник між призначеним для користувача додатком і віддаленим мережевим ресурсом до Інтернет. Принцип його роботи схематично показано на рис. 3.7.

Proxy-сервер складається з двох частин: клієнтської і серверної. Клієнтська частина дивиться у бік Інтернет, серверна – у бік клієнтського комп'ютера. Коли клієнтський комп'ютер звертається до віддаленого сайту через proxy-сервер, його клієнтський мережевий додаток взаємодіє з серверною частиною proxy-сервера. При цьому proxy-сервер на рівні додатку передає клієнтський запит своїй клієнтській частині, і вона вже від імені proxy-сервера посилає даний запит на віддалений сайт. Тобто в IP-пакеті, що відправляється, стоятиме вже адреса proxy-сервера. Потім одержана відповідь передається у зворотний бік від клієнтської частини proxy-сервера його серверної частини, з якою безпосередньо взаємодіє призначений для користувача комп'ютер.

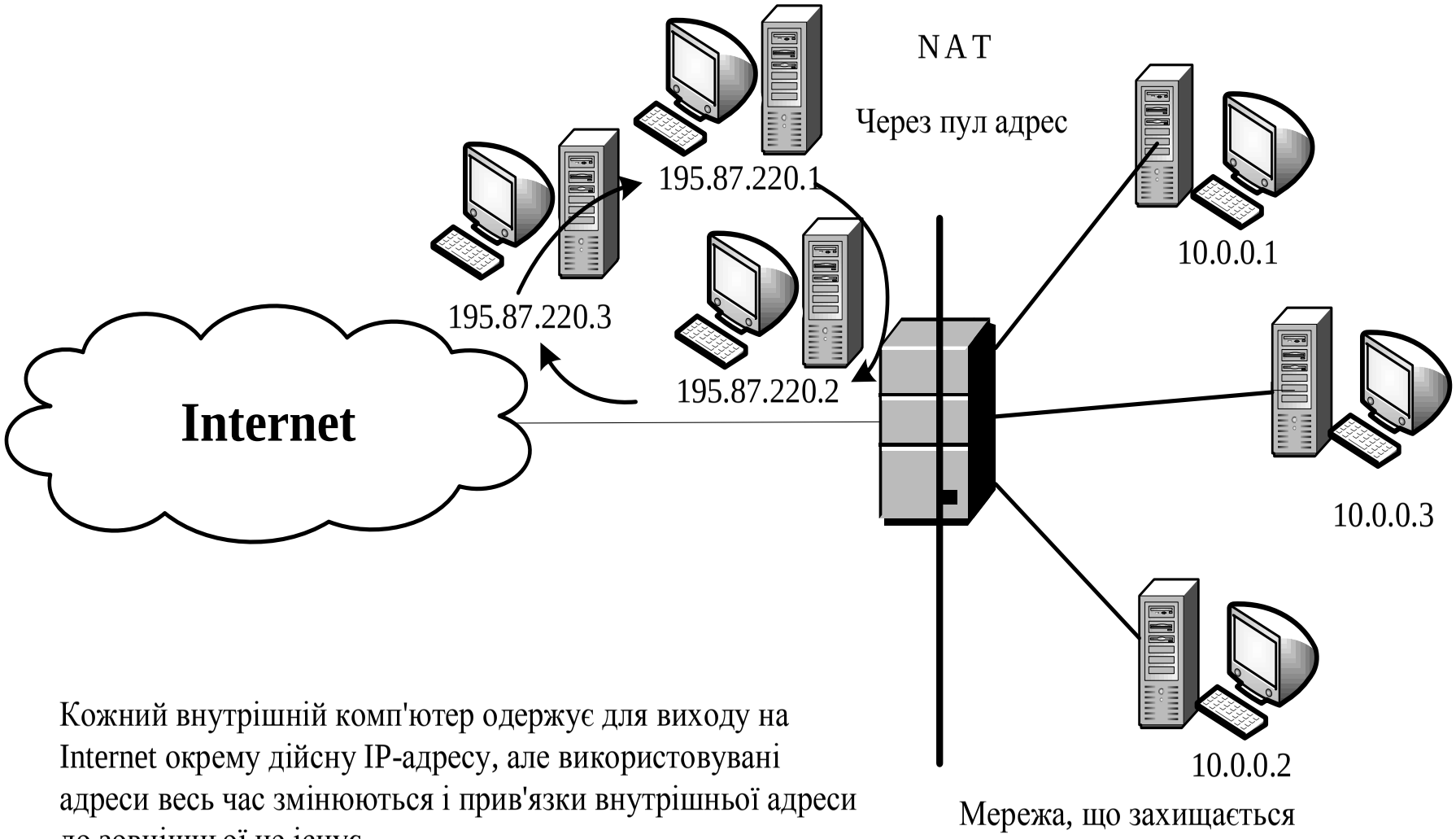


Рисунок 3.4 – Приклад побудови мережі з використанням для заміни внутрішніх адрес пулу виділених адрес

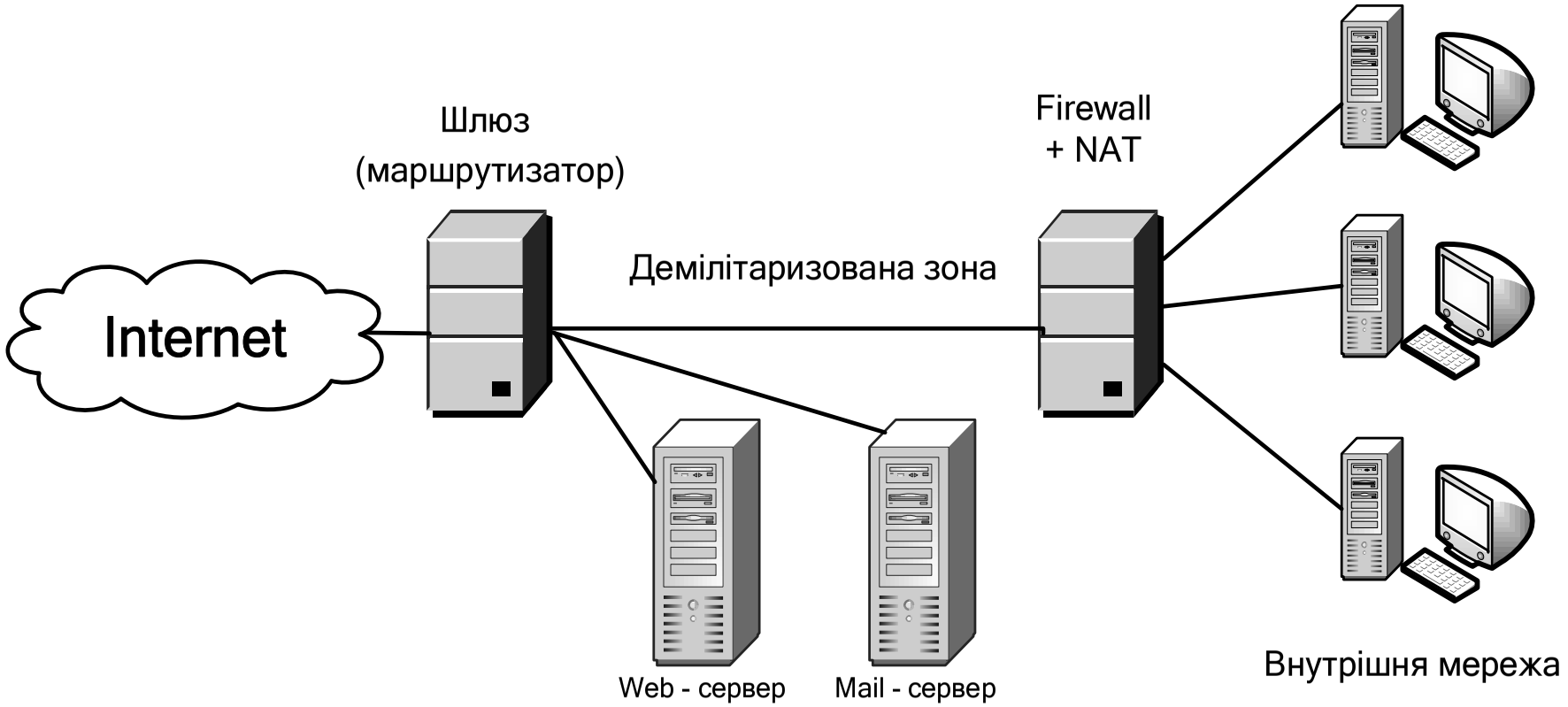


Рисунок 3.5 – Приклад побудови мережі з демілітаризованою зоною

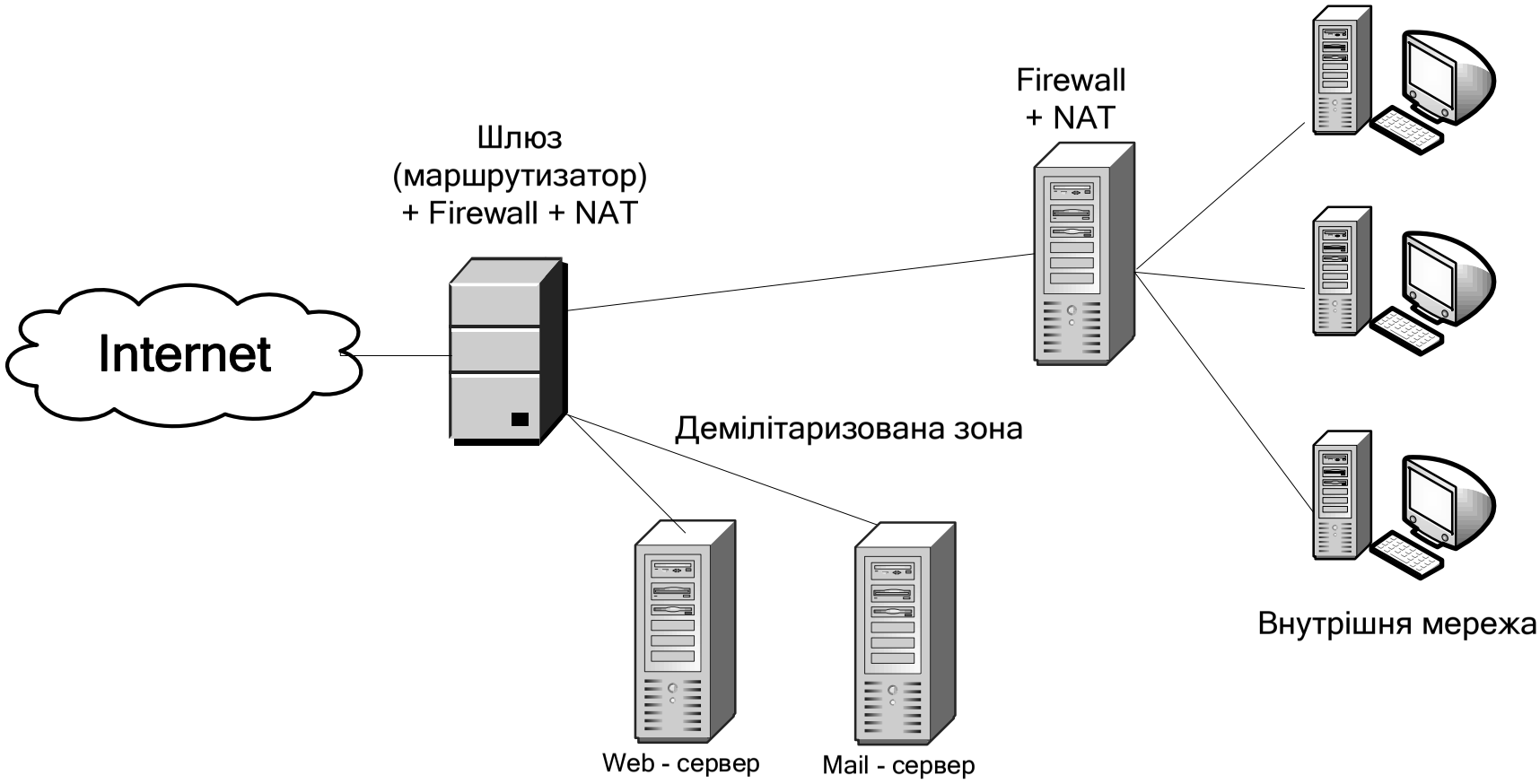


Рисунок 3.6 – Приклад побудови мережі з другим Firewall’ом (брандмауером)

Таким чином, пряме з'єднання клієнтських комп'ютерів з віддаленим сайтом виключається. Усередині proxy-сервера передача даних між клієнтською частиною і серверною відбувається вже не на транспортному рівні, а на рівні протоколу додатка, чим забезпечується легкість контролю команд і даних на відповідність установленим стандартам. Крім того, це дозволяє забезпечити достатньо надійний контроль над передачею ймовірних зловмисних кодів усередині даних.

Навіть у разі успішної атаки з боку Інтернет по відкритих протоколах у цьому випадку буде пошкоджено тільки proxy-сервер, що не становить інформаційної цінності, а призначені для користувача комп'ютери залишатимуться в безпеці ще якийсь час.

Оскільки proxy-сервер працює тільки за кількома відомими протоколами (HTTP, FTP та ін.) і не пропускає через себе решту пакетів, він дуже сильно обмежує можливості зловмисника щодо використання мережевих троянських коней для закріплення на якомусь з призначених для користувача комп'ютерів.

Залишати mail-сервер в демілітаризованій зоні, з одного боку, небажано, оскільки на ньому фактично зберігається поштова база даних з листуванням локальних користувачів, а демілітаризована зона не може забезпечити належного рівня захисту мережевим ресурсам. З іншого боку, якщо заховати mail-сервер усередині локальної мережі, то він або не зможе взаємодіяти із зовнішнім світом, або стане брамою з зовнішнього світу до внутрішньої мережи, якою потенційно зможе скористатися зловмисник.

З огляду на це хорошим рішенням є використання **двох поштових серверів**. Основний сервер установлюється всередині мережі, що захищається, і є не видимим для зовнішнього світу. Всі локальні користувачі поштової системи реєструються на ньому і мають до нього прямий доступ. Відповідно, вся вхідна кореспонденція зберігається на ньому в поштових скриньках локальних користувачів. Відправка електронної пошти також здійснюється через нього.

Другий, або зовнішній, поштовий сервер установлюється в демілітаризованій зоні і забезпечує взаємодію по e-mail з Інтернет. Він настроюється так, щоб усю пошту, що приходить на ім'я користувачів організації, миттєво пересилати на внутрішній поштовий сервер. Таким чином, в його поштовій базі даних немає жодного облікового запису користувачів організації і жоден лист не відправляється на довготривале зберігання. Тобто якщо зловмисник зламає поштовий сервер, то не дістане доступу до архівів листування. Проте після зламу зловмисник дістає можливість перехоплення і читання транзитної пошти. Тому потрібен ретельний контроль за подібною ситуацією і негайне вживання заходів при підозрі на НСД.

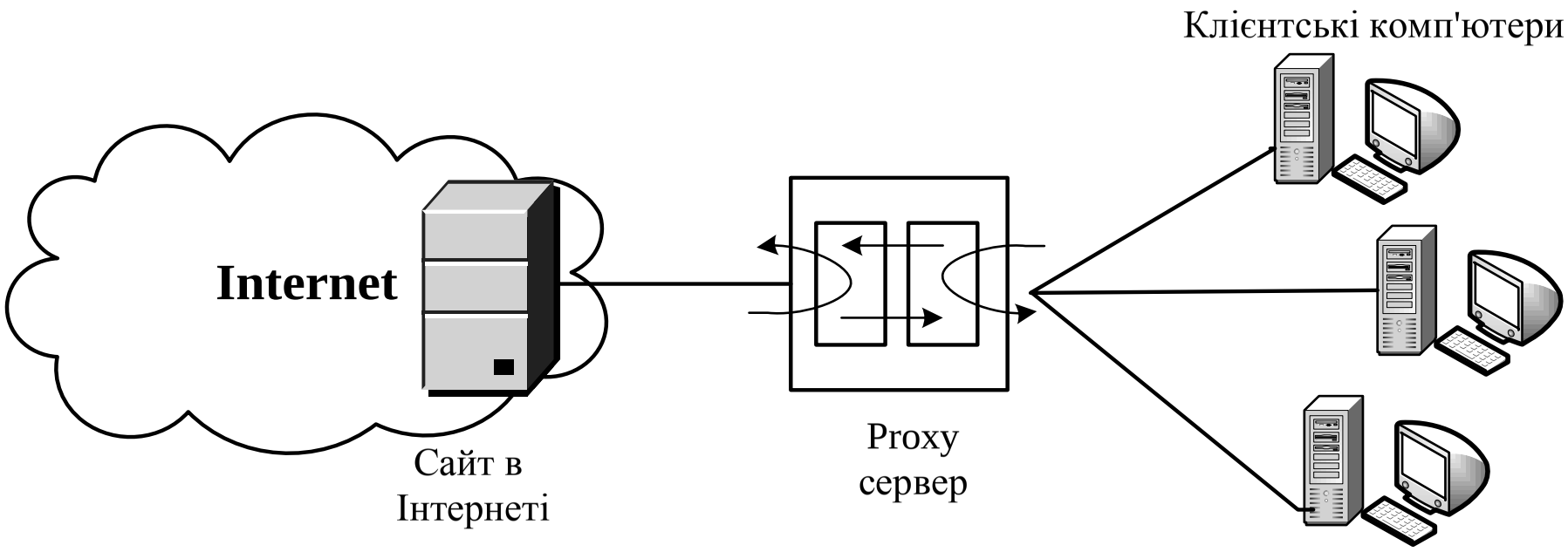


Рисунок 3.7 – Приклад побудови мережі з proxy-сервером

Важливою перевагою такої схеми є те, що навіть із зламаного зовнішнього поштового сервера не так просто дістатися до внутрішньої мережі, що захищається. Обмін даними між зовнішнім і внутрішнім поштовими серверами відбувається через міжмережевий екран з єдиним дозволеним портом (smtp) за єдиною дозволеною парою адрес. Звернення до інших комп'ютерів і за іншими протоколами блокуватиметься. Тому впливати з нього безпосередньо на комп'ютери користувачів внутрішньої мережі неможливо.

Поштова система потребує **антивірусного захисту.** Операційна система Windows дуже уразлива перед деякими різновидами поштових вірусів. Користувачу достатньо встановити вказівник на інфікований конверт, щоб вірус активізувався. Але набагато небезпечнішим є те, що механізм роботи поштових вірусів може бути використаний зловмисником для закидання в захищену зону мережевого троянського коня. Він дозволить зловмиснику потай викачувати дані з вашої мережі і вивідати інформацію, що цікавить його. Тому забезпеченню антивірусного захисту тракту доставки пошти до внутрішньої мережі слід приділити достатньо серйозну увагу.

Існує ряд програмних засобів, призначених для контролю кореспонденції на поштових серверах на предмет наявності в ній вірусів у процесі приймання і пересилки електронної пошти. Одним з таких засобів є програма kavkeeper з пакета Антивірус Касперського для Linux Server версії 4.0 і вище.

Принцип її роботи полягає в тому, що вся пошта, яка проходить через сервер, спочатку перенаправляється спеціальному користувачу, в ролі якого виступає антивірусний процес. Він сканує вміст кожного листа на наявність у ньому фрагментів відомих вірусів. Якщо лист містить щось схоже на вірус, він вилучається з процесу передачі і, залежно від настроювань антивірусу, піддається заданій обробці. Повідомлення про виявлений вірус відсилаються відправнику і одержувачу інфікованого листа, а також на ім'я вказаних адміністраторів системи. Після перевірки листи, що не викликають підозр, відсилаються за призначенням.

Тим самим на рівні поштового сервера ставиться надійний заслін відомим вірусам, які розповсюджуються за допомогою електронної пошти. А оскільки kavkeeper розпізнає тільки віруси, сигнатури яких знаходяться в його базі даних, необхідно регулярно оновлювати антивірусну базу даних з офіційного сайту. Інакше мережа може стати уразливою для знов створених вірусів.

Log-сервер – це загальновідомий механізм протоколювання системних подій на серверах і клієнтських робочих станціях. Розробники ПЗ включають до своїх продуктів фрагменти коду, які на ту або іншу подію генерують відповідні текстові повідомлення. Система збирає дані повідомлення в log-файлах, які потім можуть аналізуватися адміністратором або користувачем з

метою з'ясування, які події відбувалися в системі якийсь час тому. Це дозволяє, наприклад, з'ясувати, чому не запускається та або інша програма або перестав функціонувати певний сервіс. Дуже корисні log-файли для пошуку слідів зламу системи і відвідувань її несанкціонованими гостями. А оскільки злам, як правило, супроводжується безліччю заборонених дій, це викликає велику кількість системних повідомлень, що осідають у log-файлах.

З цієї причини зловмисник завжди прагне стерти сліди своєї присутності, видаливши або вичистивши log-файли. В обох випадках адміністратору буде дуже важко зрозуміти, що саме відбулося в системі: яким чином до неї проникли, як довго знаходилися, що встигли використати. Або навіть просто переконатися, що все гаразд. Тому обов'язковою умовою для мережі, підключеної Інтернет, є наявність в ній окремого log-сервера. Принцип його роботи полягає в тому, що кожна операційна система може посилати повідомлення про системні події по UDP протоколу на віддалений сервер. Це можуть робити також маршрутизатори і міжмережеві екрани. Збираючи такі повідомлення на спеціально виділеному сервері, ми забезпечуємо їм збереження від рук зловмисника. Тому для мінімізації імовірності зламу log-сервер повинен бути призначений тільки для збору log-повідомлень. Він не повинен виконувати будь-яких інших функцій і інші мережеві додатки, окрім syslogd. У цьому випадку після зламу комп'ютерів мережі на log-сервері залишаться відповідні повідомлення, знищити які зловмисник вже не зможе.

Таким чином, як приклад підключення локальної мережі установи (організації) до мережі Інтернет для захисту інформації і системних ресурсів можливо (а у ряді випадків й доцільно) використання схеми, поданої на рис. 3.8.

3.2. Захист електронної пошти. Система PGP

Система PGP (Pretty Good Privacy – цілком надійна секретність), що є плодом зусиль однієї людини – Філа Циммермана (Phil Zimmermann), забезпечує конфіденційність і сервіс автентифікації, які можна використовувати для електронної пошти і додатків зберігання файлів.

3.2.1. Коротка характеристика функцій системи PGP

Система PGP швидко одержала визнання і сьогодні використовується дуже широко. Серед причин такого швидкого визнання можна виділити наступні.

Система PGP широко доступна у версіях, що виконуються на великій кількості платформ, включаючи DOS/Windows, UNIX, Macintosh і багато

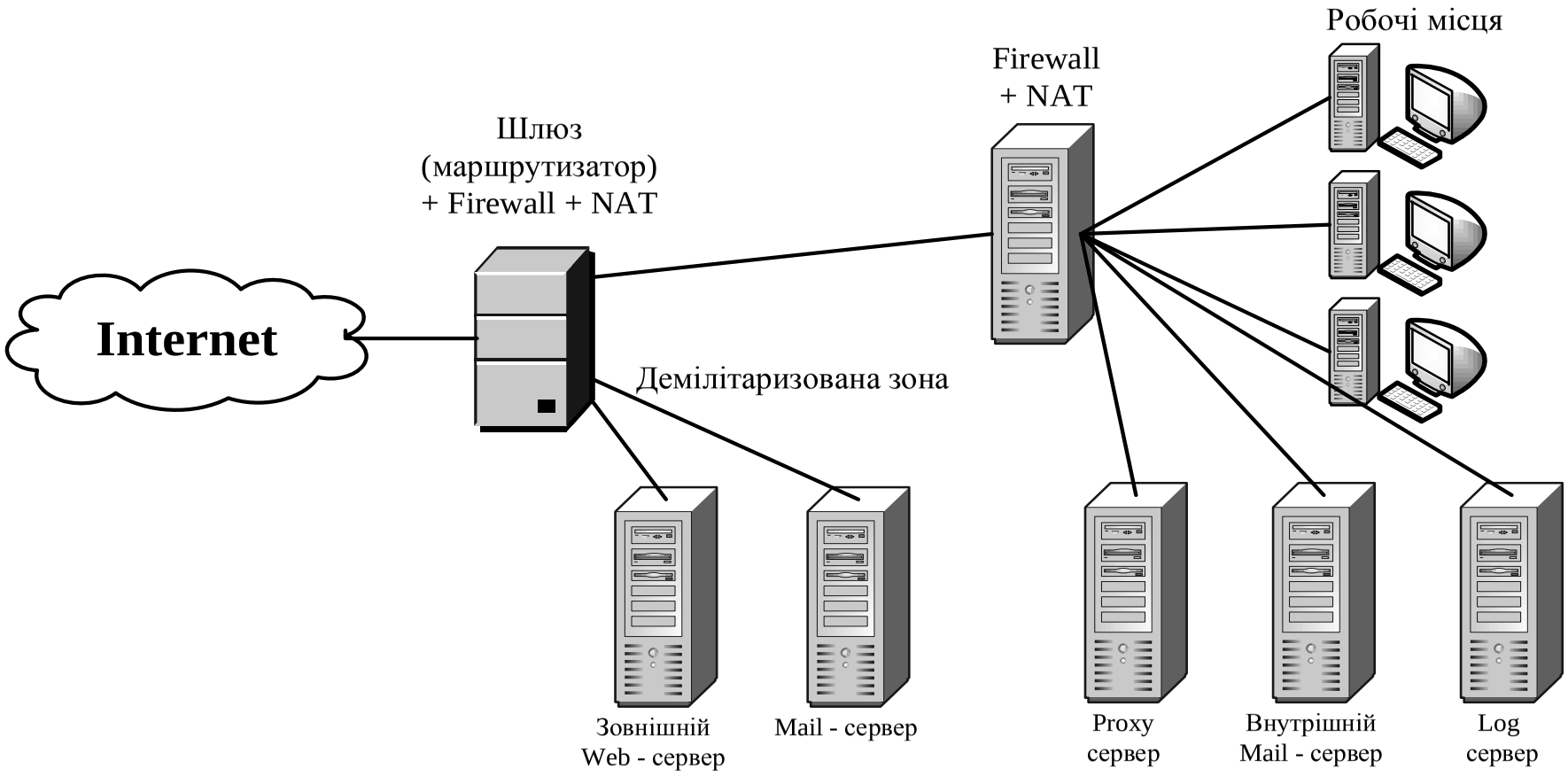


Рисунок 3.8 – Приклад побудови локальної мережі при підключенні до мережі Інтернет

інших. Крім того, існує комерційна версія, покликана задовольнити користувачів, які вважають за краще мати підтримку виробника.

Система PGP використовує загальновідомі популярні алгоритми, які витримали перевірку практикою і вважаються виключно надійними. Зокрема, до пакета включено алгоритми шифрування з відкритим ключем RSA, DSS і алгоритм Діфі-Хелмана, алгоритми традиційного шифрування CAST-128, IDEA, 3DES і AES, а також алгоритм хешування SHA-1.

Система PGP має дуже широку галузь застосування – від корпорацій, які хочуть мати стандартизовану схему шифрування файлів і повідомлень, до простих користувачів, які потребують захисту свого листування з іншими користувачами в мережі Інтернет або іншій мережі.

Система PGP не була розроблена урядовою або іншою офіційною організацією, і тому непідконтрольна їм.

Наведемо список позначень, використовуваних надалі при розгляді матеріалу:

K – сеансовий ключ, який використовується в схемі традиційного шифрування;

KRa – особистий ключ A, який використовується в схемі шифрування з відкритим ключем;

KUa – відкритий ключ A, який використовується в схемі шифрування з відкритим ключем;

EP – шифрування в схемі з відкритим ключем;

DP – розшифрування в схемі з відкритим ключем;

ЄC – шифрування в схемі традиційного шифрування;

DC – розшифрування в схемі традиційного шифрування;

H – функція хешування;

|| – конкатенація;

Z – стиснення за допомогою алгоритму ZIP;

R64 – перетворення у формат radix-64 ASCII.

У документації PGP часто використовується термін “секретний ключ”, що означає ключ, який становить пару з відкритим ключем у схемі шифрування з відкритим ключем. У зв'язку з цим існує можливість переплутати такий ключ з секретним ключем, який використовується для традиційного шифрування. Тому використовуємо термін “особистий ключ”.

Спочатку розглянемо загальні принципи роботи PGP, потім з'ясуємо, як створюються і зберігаються криптографічні ключі, і обговоримо питання керування відкритими ключами.

3.2.2. Принцип роботи системи

Сервіс системи PGP, якщо не розглядати керування ключами, складається з п'яти функцій: автентифікації, конфіденційності, стиснення, сумісності на рівні електронної пошти і сегментації (табл. 3.5). Розглянемо кожну з них.

На рис. 3.9 а показано схему сервісу цифрового підпису, що використовується в системі PGP. Послідовність дій при цьому виглядає таким чином.

Відправник створює повідомлення за допомогою алгоритму SHA-1, внаслідок чого виходить 160-бітовий хеш-код повідомлення.

Одержаний хеш-код шифрується за допомогою алгоритму RSA з використанням особистого ключа відправника, і результат додається на початок повідомлення.

Одержувач використовує RSA з відкритим ключем відправника, щоб розшифрувати і відновити хеш-код.

Одержувач генерує новий хеш-код одержаного повідомлення і порівнює його з розшифрованим хеш-кодом. Якщо хеш-коди збігаються, повідомлення вважається справжнім.

Комбінація SHA-1 і RSA забезпечує ефективну схему цифрового підпису. Зважаючи на надійність RSA, одержувач упевнений в тому, що тільки власник відповідного секретного ключа міг створити цей підпис. Надійність SHA-1 дає одержувачу впевненість у тому, що ніхто інший не міг створити інше повідомлення з відповідним хеш-кодом і, отже, з підписом з оригінального повідомлення.

Підписи можуть також генеруватися за допомогою DSS/SHA-1.

Хоча підписи зазвичай приєднуються до повідомлень або файлів, для яких вони створюються, часто буває так, що підтримуються і відокремлені підписи. Відокремлений підпис може зберігатися і передаватися окремо від самого повідомлення. Це стає корисним у цілому ряді випадків. Користувач може мати окремий протокол підписів усіх посланих і одержаних їм повідомлень. Відокремлений підпис виконаної програми може згодом допомогти виявити зараження програми вірусом. Нарешті, такі підписи можуть використовуватися тоді, коли підписувати документ повинна не одна, а кілька сторін, як, наприклад, у разі контракту. Підпис кожної сторони виявляється тоді незалежним і, таким чином, застосовується тільки до даного документа. Інакше підписи повинні бути вкладеними так, що друга сторона підписувала б документ разом з підписом першої сторони і т.ін.

Таблиця 3.5 – Коротка характеристика функцій системи PGP

Функція	Використовувані алгоритми	Опис
1	2	3
Цифровий підпис	DSS/SHA або RSA/SHA	За допомогою SHA-1 створюється хеш-код повідомлення. Одержаний таким чином профіль повідомлення шифрується за допомогою DSS або RSA з використанням особистого ключа відправника і включається до повідомлення
Шифрування повідомлення	CAST, IDEA, AES або потрійний DES з трьома ключами і алгоритмом Діфі-Хелмана, RSA	Повідомлення шифрується за допомогою CAST-128 або IDEA, або 3DES з одноразовим сеансовим ключем, відправником, що генерується. Сеансовий ключ шифрується за допомогою алгоритму Діфі-Хелмана або RSA з використанням відкритого ключа одержувача і включається до повідомлення
Стиснення	ZIP	Повідомлення можна стиснути для зберігання або передачі, використовуючи ZIP

Продовження табл. 3.5

1	2	3
Сумісність на рівні електронної пошти	Перетворення у формат radix-64	Щоб забезпечити прозорість для всіх додатків електронної пошти, шифроване повідомлення можна перетворити на рядок ASCII, використовуючи перетворення у формат radix-64
Сегментація	—	Щоб задовольнити обмеження максимального розміру повідомлень, PGP виконує сегментацію і зворотну збірку повідомлення

У сучасних автоматизованих системах керування, комп'ютерних системах і мережах, різних інформаційних і телекомунікаційних системах, інформаційно-телекомунікаційних системах висуваються високі вимоги до забезпечення цілісності, спостережливості і автентичності (справжності) інформації на всіх етапах їх життєвого циклу. При цьому, під інформацією ми будемо розуміти сукупність усіх даних і програм, що використовуються в системі чи технології, незалежно від їхнього логічного чи фізичного подання. Під інформацією будемо розуміти також і повідомлення, що циркулюють у відповідних системах чи технологіях. Досвід застосування і проведені дослідження показали, що ці високі вимоги, особливо з реалізації функції причетності (неспростовності), можуть бути забезпечені тільки за рахунок застосування електронного цифрового підпису. Електронний цифровий підпис (ЕЦП), по суті, являє собою додані до інформації дані, обчислені за допомогою криптографічного перетворення інформації, що захищається, і параметри, наявність яких дозволяє упевнитися в цілісності інформації і дійсності її джерела, а так само забезпечити захист від підробки з боку отримувача.

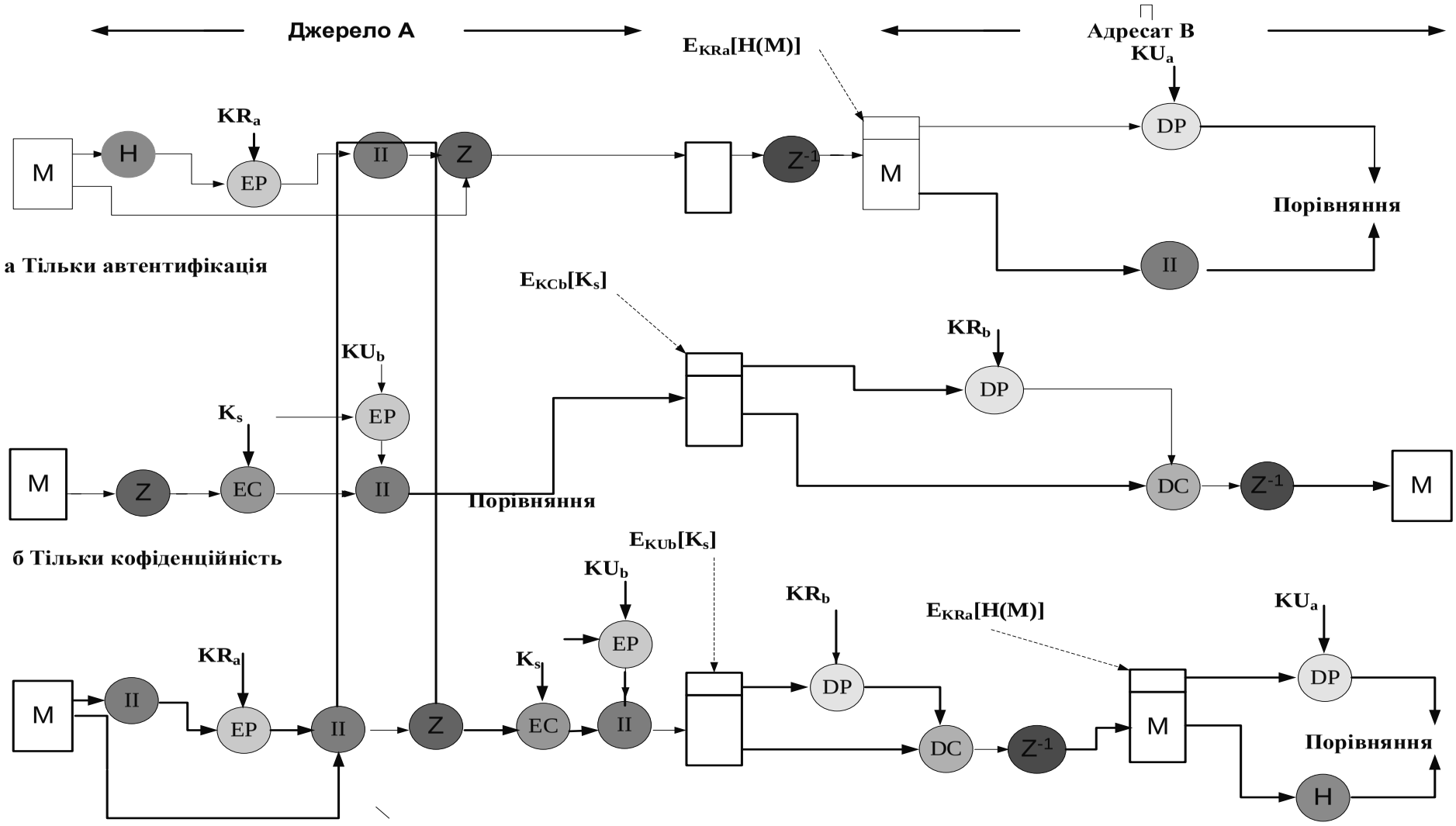


Рисунок. 3.9 – Криптографічні функції системи PGP

Власне кажучи, електронний цифровий підпис (цифровий підпис) є цифровим еквівалентом підпису (штампа, печатки, водяного знаку і т.д.), наявність якого в повідомленні, чи даних програми дозволяє з високою імовірністю визначити джерело (джерела) цього повідомлення чи даних і юридично довести, що з зазначеною припустимою імовірністю P_n тільки він міг сформувати цей підпис, але підробити його в процесі заданого часу, при обмежених ресурсах, зловмисник може з імовірністю, що не перевищує заданої величини P_z. Причому ЕЦП у такий спосіб обчислюється на основі інформації, що захищається, з використанням особистого (конфіденційного) ключа конкретного суб'єкта чи об'єкта, що є його джерелом чи відправником. Перевірка цілісності і дійсності виконується з використанням відкритого ключа, причому знання відкритого ключа не дозволяє підробити ЕЦП з імовірністю, що перевищує P_z.

Криптографічні перетворення типу ЕЦП мають задовольняти ряд вимог, основними з них є такі:

- алгоритми розробки і перевірки ЕЦП мають бути відкритими, тобто несекретними;
- алгоритми розробки і перевірки ЕЦП повинні мати не вище ніж поліноміальну складність;
- алгоритм побудування конфіденційного ключа і/чи підробки підпису повинний мати не нижче, ніж експонентну, тобто практично не реалізовану складність;
- ЕЦП має бути чутливою до будь-яких змін підписаних даних, тобто виявляти порушення цілісності;
- імовірність появи двох однакових підписів у різних повідомленнях не має перевищувати припустимого значення;
- обчислювальна складність вироблення і перевірки ЕЦП має бути мінімізована і мати близькі за величиною значення;
- забезпечувати захист від підміни, підробки й імітації повної ЕЦП із необхідною імовірністю;
- ЕЦП, отримані для однієї і тієї ж інформації в різний час і на різних пристроях мають відрізнятися з великою імовірністю;
- ключі вироблення ЕЦП мають бути конфіденційними, а ключі перевірки ЕЦП – відкритими;
- ЕЦП повинен мати максимальну стійкість до виявлення будь-яких змін, підробок і порушень;
- має існувати можливість прийняття ЕЦП із різними рівнями стійкості і складністю вироблення і перевірки ЕЦП;

- можливість програмної, апаратно-програмної й апаратної реалізації ЕЦП із приблизно однаковою складністю;
- можливість використання ЕЦП як з однаковими загальносистемними параметрами в мережі, так і індивідуальними для окремої частини об'єктів (суб'єктів);
- можливість багаторівневого вироблення і перевірки ЕЦП однієї і тієї ж інформації з використанням різних ключів і за необхідності різних загальносистемних параметрів;
- використаний ЕЦП має дозволяти проведення дослідних експериментів з метою забезпечення судового розгляду й арбітражу.
- має існувати можливість збереження ЕЦП як разом з інформацією, що захищається, так і окремо від неї.

Наприкінці XX-го століття протоколи цифрового підпису отримали широке поширення в силу збільшення комп'ютеризації документообігу. У літку 2000 року президентом США Біллом Клінтоном був підписаний і вступив у дію з 1 жовтня указ «Electronic Signatures in Global and National Commerce Act», що прирівнюється в комерційних документах електронний підпис до чорнильного (більш того і сам цей указ став першим документом, підписаним електронним цифровим підписом). Європейський Союз видав розпорядження, відповідно до якого ЕЦП незабаром матиме силу у всіх країнах Союзу. Над ініціативами в даній області, взаємодіючи один з одним, працює багато азіатських держав, причому в деяких з них ЕЦП уже закріплений законодавчо. У Російській Федерації опублікований і доступний по Internet Проект федерального закону «Про електронний цифровий підпис».Уряд Сінгапуру оголосив, що з 2008 р. електронні гроші в цій країні стануть легальною валютою, які мають ходіння нарівні з наявними, при цьому здійснювати розрахунки з будь-якими торговими організаціями можна буде за допомогою кишенькового комп'ютера чи сотового телефону. В Україні Закон „Про електронний цифровий підпис” з 1 січня 2004 р. вступає в.

Паралельно з розвитком криптографічних систем, ще більш інтенсивно розвиваються математичні методи і криптоаналітичні системи, що сприяє підвищенню вимог до стійкості криптосистем, в окремому випадку до електронного цифрового підпису.

Класифікація відомих ЕЦП

Абсолютна більшість розроблених і використовуваних у світі ЕЦП базується на використанні несиметричних криптографічних перетворень, виконуваних у кільцях, полях Галуа і групі точок еліптичних кривих. До ЕЦП, реалізованих у кільцях, необхідно віднести RSA подібні алгоритми, до перетворень у полях Галуа – алгоритми Діффі-Хеллмана і Ель-Гамаля. Досвід застосування і проведення досліджень ЕЦП, що базуються на перетвореннях у

кільцях і полях, показали, що вони практично вичерпали себе і найближчим часом не забезпечуватимуть необхідної стійкості. Одним зі способів вирішення поставленої задачі є збільшення довжини ключа нині діючих цифрових підписів: *RSA* і *DSA*. Однак збільшення довжини ключа підвищує вимогу цих криптосистем до обчислювальних можливостей ЕОМ, що не завжди є прийнятним (не всі організації в стані поміняти весь парк комп'ютерів для здійснення прийнятного рівня швидкодії обновлених алгоритмів електронного цифрового підпису). Для вирішення цього протиріччя розроблені і почали впроваджуватися нові модифіковані криптографічні перетворення, що виконуються в групі точок еліптичних кривих. З'явилося значне число методів, на їх основі розроблені стандарти і проекти стандартів. Тому стає цікавою задача їх вивчення, виявлення особливостей і можливостей, аналізу стійкості і складності виконання ЕЦП. Основою при порівняльному аналізі звичайно ж мають бути вимоги, наведені вище. Разом з тим, при виконанні порівняльного аналізу необхідно задатися видами атак і типами погроз ЕЦП відповідної інформації.

На наш погляд основними видами атак на ЕЦП є такі:

1. *Атака на основі відомого відкритого ключа* (key-only attack). Найслабкіша з атак, практично завжди доступна криптоаналітику (зловмиснику). Вона може виконуватися при апріорній визначеності криптоаналітика щодо реалізації ЕЦП, знанні загальносистемних параметрів, а також діючих відкритих ключах.

2. *Атака на основі відомих підписаних повідомлень* (known-message attack). Для цієї атаки передбачається, що в розпорядженні криптоаналітика є деяке число пар $(m, \langle r, s \rangle)$ підписаних повідомлень m, при цьому він не може вибирати повідомлення m. Крім цього криптоаналітик знає систему і параметри ЕЦП.

3. *Проста атака з вибором підписаних повідомлень* (generic chosen-message attack). У цьому випадку криптоаналітик має можливість вибрати деяку кількість підписаних повідомлень, знає загальносистемні параметри і має доступ до відкритих ключів після вибору підписаних повідомлень.

4. *Спрямована атака з вибором повідомлення* (direct chosen-message attack). Криптоаналітик знає загальносистемні параметри, може за своїм розсудом вибирати відкритий ключ і після цього вибирати підписані повідомлення.

5. *Адаптивна атака з вибором підписаного повідомлення* (adaptive chosen-message attack). При здійсненні атаки криптоаналітик може вибирати відкритий ключ, а також підписане повідомлення. При цьому вибір наступного підписаного повідомлення він може робити на основі знання припустимого підпису попереднього обраного повідомлення.

Проведений аналіз показав, що кожна атака спрямована на досягнення визначеної мети. З урахуванням цього можна виділити такі види погроз, у порядку зростання небезпеки, для всіх схем електронних цифрових підписів:

1. *Екзистенціальна підробка* (existential forgery). Погроза полягає в створенні криптоаналітиком для будь-якого, можливо безглуздого повідомлення m', що відрізняється від перехопленого, реального (правильного) ЕЦП.

2. *Селективна підробка* (selective forgery). Являє загрозу створення для заздалегідь обраного повідомлення m правильного ЕЦП.

3. *Універсальна підробка* (universal forgery). Ця погроза полягає у знаходженні криптоаналітиком алгоритму формування підпису, функціонально еквівалентному дійсному алгоритму ЕЦП, що дозволяє створити чи модифікувати дійсні підписані повідомлення.

4. *Повне розкриття* (total break). При цій погрозі криптоаналітик може обчислити таємний ключ, можливо відмінний від d, але відповідний відкритому ключу Q. Це дозволяє криптоаналітику формувати цифрові підписи для будь-яких повідомлень і надалі нав'язувати ці повідомлення кореспондентам.

Іншим основним сервісом, запропонованим PGP, є конфіденційність, що забезпечується шифруванням повідомлень, призначених для передачі або зберігання у вигляді файлів. В обох випадках можна використовувати традиційне шифрування за допомогою алгоритму CAST-128. Альтернативою є застосування алгоритмів IDEA, 3DES або AES. Може використовуватися і режим зворотного зв'язку шифрованих 64-бітових блоків (режим CFB).

Як завжди, необхідно розв’язувати проблему розподілу ключів. У PGP кожний ключ схеми традиційного шифрування застосовується тільки один раз. Це означає, що для кожного повідомлення генерується новий ключ у вигляді випадкового 128-бітового числа. Таким чином, хоча в документації такий ключ називається сеансовим, насправді він є одноразовим. З огляду на те, що ключ задіюється тільки один раз, такий сеансовий ключ приєднується до повідомлення і передається разом з повідомленням. Щоб захистити ключ, він шифрується з використанням відкритого ключа одержувача. На рис. 3.9 б показано відповідну схему, яка може бути описана таким чином.

Відправник генерує повідомлення і випадкове 128-бітове число, яке виступає як сеансовий ключ тільки для цього повідомлення.

Повідомлення шифрується за допомогою алгоритму CAST-128 (або IDEA, або 3DES) і даного сеансового ключа.

Сеансовий ключ шифрується за допомогою алгоритму RSA і відкритого ключа одержувача і приєднується до початку повідомлення.

Одержувач використовує RSA з особистим ключем, щоб розшифрувати і тим самим відновити сеансовий ключ.

Сеансовий ключ застосовується для розшифрування повідомлення.

Щоб забезпечити альтернативу використанню RSA для шифрування ключа, в PGP пропонується параметр Diffie-Hellman (алгоритм Діфі-Хелмана). Алгоритм Діфі-Хелмана є алгоритмом обміну ключами. Насправді в PGP використовується варіант цього алгоритму з можливостями шифрування/ розшифрування, відомий як алгоритм Ель-Гамаля (ElGamal).

У зв'язку з цим можна зробити декілька зауважень. По-перше, щоб зменшити час шифрування, перевага віддається використанню комбінації традиційного шифрування і шифрування з відкритим ключем, а не простому використанню RSA або алгоритму Ель-Гамаля, коли повідомлення шифрується безпосередньо: CAST-128 і інші алгоритми традиційної схеми шифрування значно швидші, ніж RSA або алгоритм Ель-Гамаля. По-друге, використання алгоритмів схеми шифрування з відкритим ключем розв’язує проблему розподілу сеансових ключів, оскільки тільки для одержувача виявляється можливим відновити сеансовий ключ, приєднаний до повідомлення. У цій ситуації, швидше, кожне повідомлення є незалежною одиничною подією з своїм власним ключем. До того ж, унаслідок самої природи електронної пошти, що є системою з проміжним зберіганням даних, використання процедури підтвердження зв'язку для того, щоб переконатися в ідентичності сеансового ключа обох сторін, не є практично зручним рішенням. Нарешті, використання одноразових ключів у традиційній схемі шифрування ще більш підсилює і без того достатньо надійний алгоритм традиційного шифрування. Тільки невеликий обсяг відкритого тексту шифрується з використанням одного ключа, і між ключами немає ніякого зв'язку. Таким чином, уся схема виявляється захищеною тією мірою, в якій захищений алгоритм схеми шифрування з відкритим ключем. Тому PGP пропонує користувачу вибір для довжини ключа від 768 до 3072 біт (довжина ключа DSS для підписів обмежується величиною в 1024 біт).

Як показано на рис. 3.9 в, для одного повідомлення можна використовувати обидві служби. Спочатку для повідомлення у вигляді відкритого тексту генерується підпис, який додається в початок повідомлення. Потім відкритий текст повідомлення і підпис шифруються за допомогою алгоритму CAST-128 (або IDEA, або 3DES), а сеансовий ключ шифрується за допомогою RSA (або алгоритму Ель-Гамаля). У загальному випадку виявляється зручнішим зберігати підпис з відкритим текстом повідомлення. До того ж, з погляду можливостей трибічної верифікації, якщо спочатку генерується підпис, третій стороні не потрібно піклуватися про ключ традиційного шифрування, щоб перевірити підпис.

Таким чином, при використанні обох служб відправник спочатку підписує повідомлення за допомогою власного особистого ключа, потім

шифрує повідомлення за допомогою сеансового ключа і, нарешті, шифрує сеансовий ключ за допомогою відкритого ключа одержувача.

За умовчанням PGP стискає повідомлення після створення підпису, але перед шифруванням. Це має сенс з погляду зменшення обсягу даних як при передачі електронної пошти, так і при зберіганні у вигляді файлів.

Дуже важливим є вибір місця застосування алгоритму стиснення, позначеного на рис. 3.9 як Z при стисненні і як Z-1 при розпаковуванні даних.

Підпис генерується до стиснення з таких причин:

- Слід підписувати нестисле повідомлення, щоб у майбутньому мати можливість зберігати повідомлення в нестислому вигляді разом з підписом. Якщо підписати стислий документ, то для верифікації необхідно буде зберігати стислу версію повідомлення або стискати повідомлення кожного разу, коли потрібна верифікація.
- Навіть за наявності можливості динамічно повторно стискати повідомлення для верифікації такий підхід несе в собі додаткові труднощі через алгоритм стиснення PGP: алгоритм не є детермінованим і різні реалізації алгоритму вибирають різні варіанти виконання для оптимізації співвідношення швидкості виконання і стиснення, а в результаті виходять стислі файли різної форми. Такі різні алгоритми стиснення є переносними через те, що будь-яка версія алгоритму може правильно відновити дані, одержані за допомогою будь-якої іншої версії. Застосування функції хешування і створення підпису після стиснення примусило б у всіх реалізаціях PGP застосовувати один і той же алгоритм стиснення.

Шифрування повідомлення застосовується після стиснення для того, щоб підсилити криптографічний захист повідомлення. З огляду на те, що стисле повідомлення має меншу надмірність у порівнянні з оригінальним відкритим текстом, криптоаналіз виявляється важчою справою.

При використанні PGP шифрується, принаймні, частина переданого блока. Якщо потрібен тільки цифровий підпис, то шифрується профіль повідомлення (з використанням особистого ключа відправника). Якщо має місце сервіс конфіденційності, шифрується (з використанням одноразового симетричного ключа) повідомлення плюс підпис (за наявності останньої). Таким чином, частина або весь вихідний блок повідомлення є потік довільних 8-бітових байтів. Проте багато систем електронної пошти дозволяють використовувати тільки блоки, що складаються з символів тексту ASCII. Щоб задовольнити таке обмеження, PGP забезпечує сервіс конвертації “сирого” 8-бітового двійкового потоку в потік друкованих символів ASCII.

Для цього використовується схема конвертації radix-64. Кожна група з трьох байтів двійкових даних перетвориться на чотири символи ASCII, до яких

приєднується контрольна сума (CRC), що дозволяє виявити помилки при передачі даних.

Конвертація у формат radix-64 збільшує довжину переданого повідомлення на 33 %. Сеансовий ключ і порція підпису повідомлення відносно компактні, а відкритий текст повідомлення стискається. Фактично стиснення з остатком компенсує розширення, що одержується внаслідок перекладу у формат radix-64. Якщо ігнорувати відносно невеликий підпис і компоненти ключа, типовий повний вплив стиснення і розширення для файлу довжини X повинен приблизно дорівнювати 1,33 x 0,5 x X = 0,665 x X. Таким чином, має місце загальне стиснення приблизно на одну третину.

Заслуговує згадки наступний аспект алгоритму radix-64: він сліпо конвертує вхідний потік у формат radix-64, незважаючи на вміст, навіть якщо введення виявляється текстом ASCII. Таким чином, якщо повідомлення підписане, але не шифрується і конвертація застосовується до всього блока, то вихідний потік даних буде незрозумілим випадковому спостерігачу, що вже забезпечує певний рівень конфіденційності. PGP можна конфігурувати так, щоб конвертація у формат radix-64 виконувалося тільки для порції підпису відкритого повідомлення. Це дає одержувачу можливість прочитати повідомлення без використання PGP. Але PGP все ж таки доведеться використовувати, якщо необхідно перевірити підпис.

На рис. 3.10 показаний зв'язок між чотирма описаними вище службами. При передачі, якщо це потрібно, підпис генерується за допомогою хеш-коду відкритого тексту. Потім відкритий текст і підпис, якщо останній є, стискаються. Далі, якщо потрібна конфіденційність, блок (стислий відкритий текст або стислі підпис і відкритий текст) шифрується і до початку додається шифрований відкритим ключем ключ шифрування традиційної схеми. Нарешті, весь одержаний блок конвертується у формат radix-64.

На стороні одержувача блок, який надходить, спочатку конвертується назад з формату radix-64 у двійковий. Потім, якщо повідомлення зашифроване, одержувач відновлює сеансовий ключ і розшифровує повідомлення. Одержаний в результаті блок розтискується. Якщо повідомлення підписане, одержувач відновлює одержаний хеш-код і порівнює його з хеш-кодом, обчисленим ним самим.

Засоби електронної пошти часто обмежують максимально допустиму довжину повідомлення. Наприклад, багато засобів електронної пошти, доступних через Інтернет, допускають пересилання повідомлень завдовжки не більше 50000 байт. Будь-яке повідомлення, яке має більшу довжину, повинно бути розбито на сегменти меншої довжини, кожний з яких посилається окремо.

Щоб відповідати такому обмеженню, PGP автоматично розбиває дуже довгі повідомлення на сегменти, достатньо малі для того, щоб їх можна було переслати за допомогою електронної пошти.

Сегментація проводиться після виконання всіх інших операцій, включаючи перетворення у формат radix-64. У результаті компоненти ключа і підпису з'являються тільки один раз, на початку першого сегмента. На стороні одержувача система PGP повинна відкинути заголовок електронної пошти і знов зібрати весь оригінальний блок повідомлення перед виконанням кроків, показаних на рис. 3.10 б.

3.2.3. Криптографічні ключі і зв'язки ключів

Система PGP використовує чотири типи ключів: одноразові сеансові ключі схеми традиційного шифрування, відкриті ключі, особисті ключі і парольні ключі схеми традиційного шифрування, описані нижче. Відносно цих ключів можна сформулювати такі три вимоги.

1. Наявність засобів генерування непередбачуваних сеансових ключів.

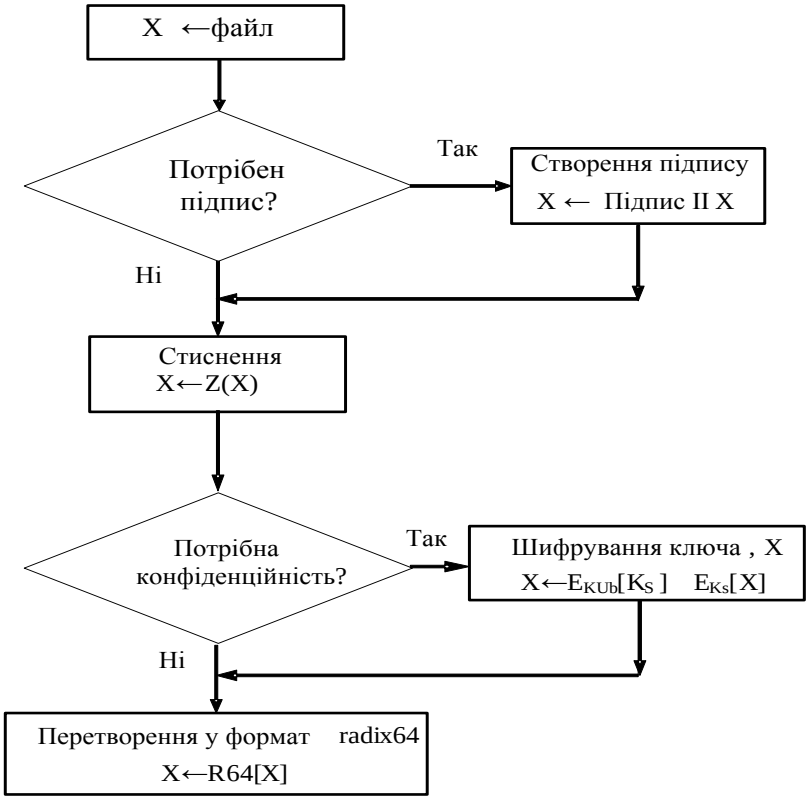
2. Наявність у користувача декількох пар відкритих/особистих ключів. Однією з причин такої вимоги є те, що користувач може час від часу міняти пару ключів. У результаті всі повідомлення на шляху проходження виявляться створеними зі старим ключем. До того ж одержувачі знатимуть тільки старий відкритий ключ, доки ними не буде одержано нову версію ключа.

3. На додаток до необхідності час від часу міняти ключі користувач може мати декілька пар ключів одночасно, щоб взаємодіяти з різними групами одержувачів або просто для того, щоб підсилити захист, обмежуючи обсяг матеріалу, що шифрується одним і тим же ключем. У результаті однозначної відповідності між користувачами і їх відкритими ключами немає.

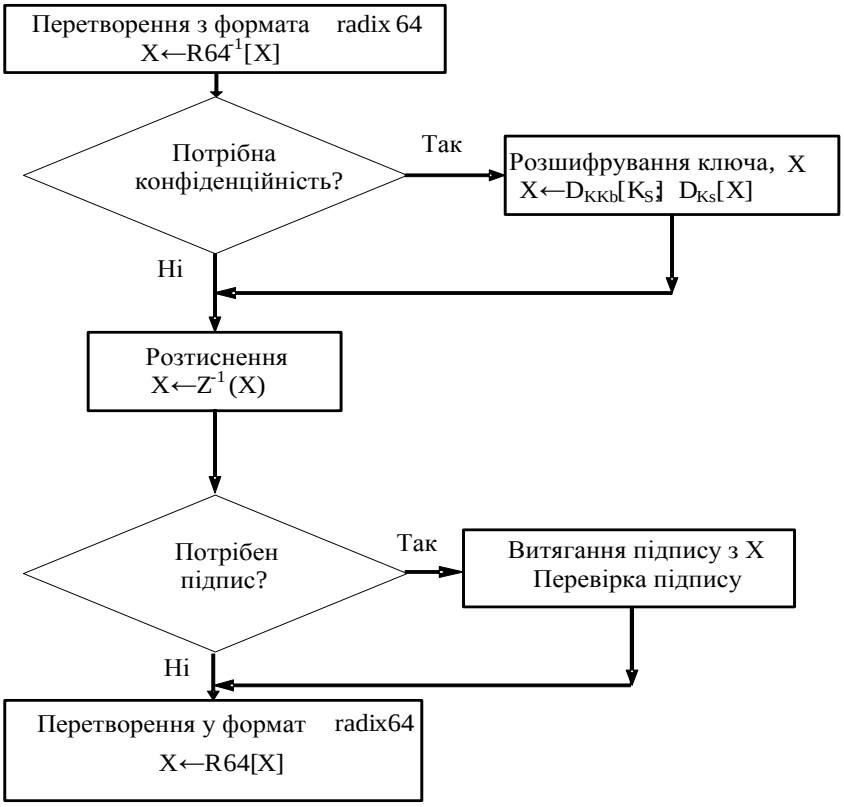
Таким чином, виникає необхідність в засобах, що дозволяють ідентифікувати конкретні ключі.

Кожен об'єкт системи PGP повинен підтримувати файл власних пар відкритих/особистих ключів, а також відкритих ключів кореспондентів.

Розглянемо ці вимоги за порядком



а Загальна схема відправки (на стороні А)



б Загальна схема отримання (на стороні В)

Рисунок 3.10 – Відправлення і приймання повідомлень PGP

Генерування сеансових ключів. Кожен сеансовий ключ зв'язується з одним повідомленням і використовується тільки для шифрування і розшифрування цього повідомлення. Пригадайте, що шифрування/розшифрування повідомлення виконується за допомогою алгоритму симетричної схеми шифрування. При цьому алгоритми CAST-128 і IDEA використовують 128-бітові ключі, а 3DES – 68-бітовий ключ. У подальшому обговоренні ми припускаємо використання CAST-128.

Випадкові 128-бітові числа генеруються за допомогою самого алгоритму CAST-128. Введення для генератора випадкових чисел складається з 128-бітового ключа і двох 64-бітових блоків, які розглядаються як відкритий текст, що підлягає шифруванню. Використовуючи режим шифрованого зворотного зв'язку, шифрувальник CAST-128 породжує два 64-бітові блоки шифрованого тексту, які зв'язуються конкатенацією, внаслідок чого формується 128-бітовий сеансовий ключ. Алгоритм, який при цьому використовується, зосновано на алгоритмі, описаному в документі ANSI X12.17.

"Відкритий текст" для генератора випадкових чисел, що формується з двох 64-бітових блоків, витягується з "рандомізованого" потоку 128-бітових чисел. Ці числа будуються на основі введення з клавіатури від користувача. Для створення "рандомізованого" потоку використовуються як час між натисненнями, так і інформація про фактично натиснуті клавіші. Таким чином, якщо користувач натискає випадкові клавіші в своєму звичайному темпі, буде породжений достатньо "випадковий" потік для введення. Це випадкове введення об'єднується з попереднім сеансовим ключем, виданим алгоритмом CAST-128, щоб сформувати дані для введення генератора В результаті, зважаючи на хороші перемішуючі властивості CAST-128, породжується послідовність сеансових ключів, яка виявляється практично непередбачуваною.

Ідентифікатори ключів. Як зазначено вище, шифроване повідомлення супроводжується використаним для шифрування сеансовим ключем у зашифрованому вигляді. Сеансовий ключ шифрується за допомогою відкритого ключа одержувача. Отже, тільки одержувач може розшифрувати сеансовий ключ і, таким чином, прочитати повідомлення. Якби кожен користувач використовував одну пару відкритого і особистого ключів, то одержувач відразу б знав, за допомогою якого з ключів можна розшифрувати сеансовий ключ – це єдиний особистий ключ одержувача. Проте ми висунули умову, щоб будь-який користувач міг мати будь-яке число пар відкритих/особистих ключів.

Як у цьому випадку одержувачу дізнатися, який з відкритих ключів використовувався для шифрування сеансового ключа? Простим рішенням є передача відкритого ключа разом з повідомленням. Одержувач міг би тоді упевнитися, що це дійсно один з відкритих ключів, а потім продовжити обробку повідомлення. Ця схема повинна працювати, але при цьому

пересилаються дуже багато зайвих даних. Відкритий ключ RSA може мати довжину в сотні десяткових розрядів. Іншим рішенням є скріплення з кожним відкритим ключем деякого ідентифікатора, унікального, принаймні, для одного користувача. Для цієї мети цілком підійде, наприклад, комбінація ідентифікатора користувача і ідентифікатора ключа. Тоді доведеться пересилати тільки значно коротший ідентифікатор ключа. Таке рішення, проте, породжує проблему керування і перевантаження: ідентифікатори ключів повинні приписуватися і зберігатися так, щоб як відправник, так і одержувач могли встановити відповідність між ідентифікаторами ключів і самими відкритими ключами. Це видається небажаним і дещо обтяжливим.

Рішенням, прийнятим в PGP, є присвоєння кожному відкритому ключу такого ідентифікатора, який з дуже високою імовірністю повинен виявитися унікальним для даного користувача. Ідентифікатор, що пов'язується з кожним відкритим ключем, розміщується в молодших 64 розрядах ключа. Це значить, що ідентифікатор відкритого ключа KU дорівнює (KUa mod 2м). Цієї довжини достатньо для того, щоб імовірність дублювання ідентифікаторів ключів виявилася дуже малою.

Ідентифікатор ключа потрібен і для цифрового підпису PGP. Через те що відправник може скористатися одним з декількох особистих ключів для шифрування профілю повідомлення, одержувач повинен знати, який відкритий ключ йому слід використовувати. Тому розділ цифрового підпису повідомлення включає 64-бітовий ідентифікатор відповідного відкритого ключа. При отриманні повідомлення одержувач перевіряє, що ідентифікатор відповідає відомому йому, відкритому ключу відправника, а потім продовжує перевірку підпису.

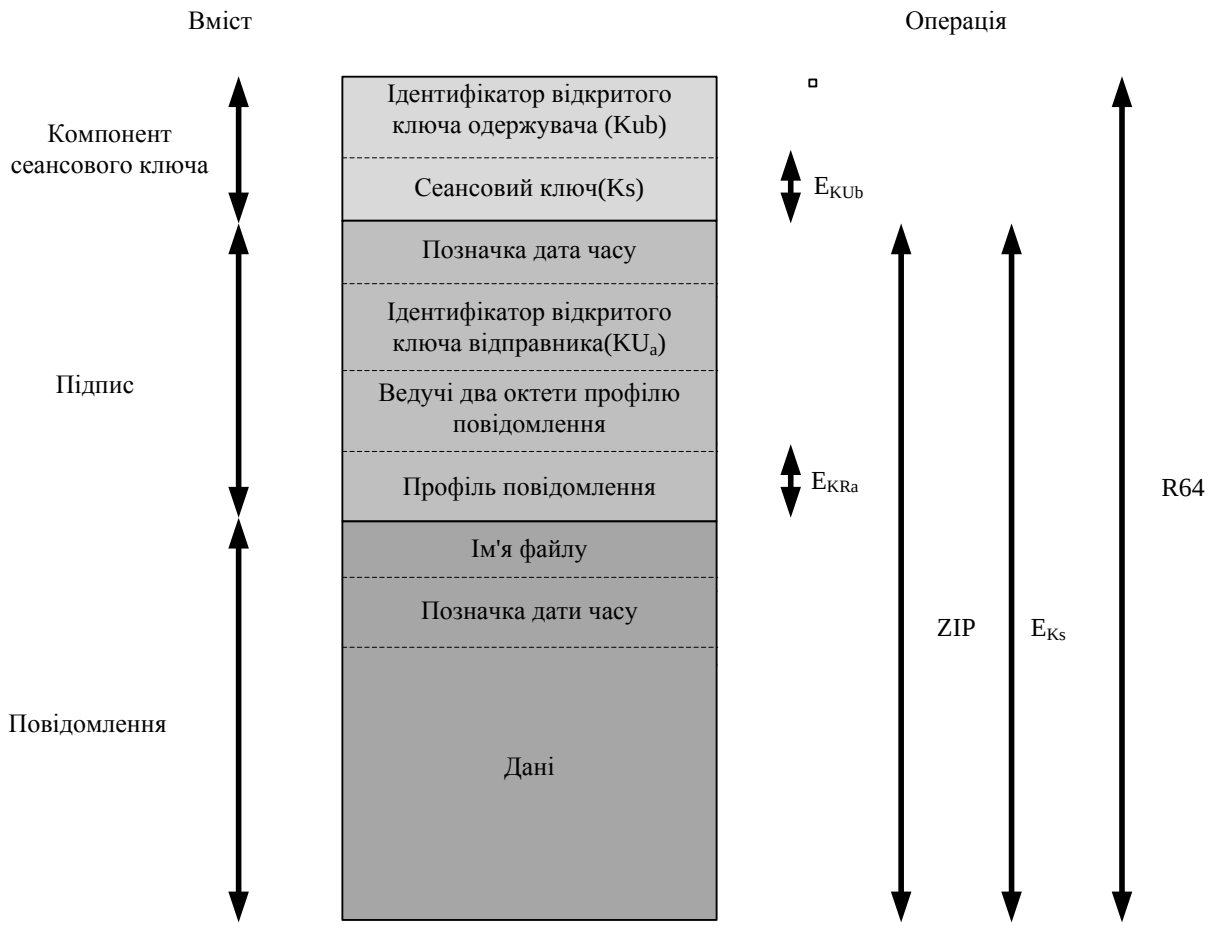
Тепер, визначивши поняття ідентифікатора ключа, ми можемо пильніше поглянути на формат переданого повідомлення, показаний на рис. 3.11.

Повідомлення складається з трьох компонентів: власне повідомлення, його підпис (необов'язково) і компонент сеансового ключа (необов'язково).

Компонент повідомлення включає фактичні дані, призначені для зберігання або передачі, а також ім'я файлу і позначку дати-часу, яка вказує на час створення повідомлення.

Компонент підпису включає такі компоненти.

Позначка дати-часу. Час створення підпису.



Позначення:
EKUb – шифрування з використанням особистого ключа для користувача B;
EKRa – шифрування з використанням відкритого ключа для користувача A;
EKs – шифрування з використанням сеансового ключа;
ZIP – функція стиснення ZIP;
R64 – функція перетворення на формат radix 64.

Рисунок 3.11 – Загальний формат повідомлення PGP (від A до B)

Профіль повідомлення. 160-бітовий профіль повідомлення, створений за допомогою SHA-1 і шифрований з використанням особистого ключа підпису відправника. Профіль обчислюється для позначки дати-часу підпису, зв'язаної конкатенацією з порцією даних компонента повідомлення. Включення указівника дати-часу підпису в профіль забезпечує захист від атак відтворення повідомлення. Виключення імені файлу і позначки дати-часу компонента повідомлення гарантує, що відокремлений підпис буде точно збігатися з підписом, що додається в префікс повідомлення. Відокремлені підписи

обчислюються для файлу, в якому немає ніяких полів заголовка (хедера) повідомлення.

Ведучі два октети профілю повідомлення. Щоб забезпечити одержувачу можливість визначити, чи відповідає відкритий ключ, що використовувався, для розшифрування профілю повідомлення з метою автентифікації, проводиться порівняння цих перших двох октетів відкритого тексту початкового профілю з першими двома октетами розшифрованого профілю. Ці октети також служать 16-бітовою послідовністю, використовуваною для перевірки повідомлення.

Ідентифікатор відкритого ключа відправника. Ідентифікує відкритий ключ, який повинен служити для розшифрування профілю повідомлення, і отже, ідентифікує особистий ключ, що використався для шифрування профілю повідомлення.

Компонент повідомлення і необов'язковий компонент підпису можуть бути стислими за допомогою ZIP і можуть бути зашифрованими з використанням сеансового ключа.

Компонент сеансового ключа включає сеансовий ключ і ідентифікатор відкритого ключа одержувача, який використовувався відправником для шифрування даного сеансового ключа.

Увесь блок зазвичай переводиться у формат radix-64.

Вище зазначалось, що ідентифікатори ключів у PGP дуже важливі і що два ідентифікатори ключів включаються в будь-яке повідомлення PGP, що припускає конфіденційність і автентифікацію. Ці ключі необхідно зберігати і організувати деяким стандартизованим чином для ефективного застосування всіма даними, що беруть участь в обміні, сторонами. Схема, використовувана в PGP, припускає створення в кожному вузлі пари структур даних: одну для зберігання пар відкритих/секретних ключів даного вузла, а іншу – для зберігання відкритих ключів інших користувачів, відомих даному вузлу. Ці структури даних називаються відповідно зв'язкою особистих ключів і зв'язкою відкритих ключів.

Загальна структура зв'язки особистих ключів показана на рис. 3.12. Зв'язку можна вважати таблицею, в якій кожен рядок являє одну пару відкритого/особистого ключів, що належать даному користувачу.

Кожен рядок містить такі поля.

Позначка дати-часу. Дата і час створення даної пари ключів.

Ідентифікатор ключа. Молодші 64 розряди відкритого ключа даного рядка.

Відкритий ключ. Відкритий ключ даної пари.

Особистий ключ. Особистий ключ даної пари; це поле шифрується

* – поле, використане для індексації таблиці

Кільце особистих ключів

Позначка дати-часу	Ідентифікатор ключа*	Відкритий ключ	Шифрований особистий ключ	Ідентифікатор користувача*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(Pi)}[KR_i]$	Користувачі
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Кільце відкритих ключів

Позначка дати-часу	Ідентифікатор ключа*	Відкритий ключ	Довіра власнику
•	•	•	•
•	•	•	•
•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$trust\_flag_i$
•	•	•	•
•	•	•	•
•	•	•	•

Ідентифікатор користувача*	Законність ключа	Підпис (підписи)	Довіра підпису (підписам)
•	•	•	•
•	•	•	•
•	•	•	•
Користувач i	$trust\_flag_i$		
•	•	•	•
•	•	•	•
•	•	•	•

Рисунок 3.12 – Загальна структура зв'язок особистих і відкритих ключів

Ідентифікатор користувача. Зазвичай тут розміщується адреса електронної пошти користувача (наприклад, ssemenov@ukr.net). Проте користувач може вказати для кожної пари ключів різні імена (наприклад, Semenov, SSemenov, SergSemenov і т.п.) або використовувати один ідентифікатор користувача кілька разів.

Зв'язку особистих ключів може бути індексовано за полем Ідентифікатор користувача або за полем Ідентифікатор ключа; мету такої індексації ми з'ясуємо пізніше.

Хоча передбачається, що зв'язка особистих ключів повинна зберігатися тільки на машині користувача, який створив і володіє відповідними парами ключів, і що вона повинна бути доступна тільки цьому користувачу, є сенс зробити значення особистих ключів захищеними настільки, наскільки це можливо. Відповідно сам особистий ключ у відкритому вигляді у зв'язці ключів не зберігається, а шифрується за допомогою CAST-128 (або IDEA, або 3DES). При цьому використовується наступна процедура.

Користувач вибирає фразу-пароль, яка служитиме для шифрування особистих ключів.

Коли система за допомогою RSA генерує нову пару відкритого/особистого ключів, вона вимагає від користувача вказати таку фразу-пароль. З неї за допомогою SHA-1 генерується 160-бітовий хеш-код, а потім пароль видаляється.

Система шифрує особистий ключ за допомогою CAST-128, використовуючи 128 біт хеш-коду як ключ. Хеш-код потім видаляється, а шифрований особистий ключ зберігається у зв'язці особистих ключів. Згодом, коли користувач звертається до зв'язки особистих ключів, щоб витягнути особистий ключ, йому доведеться знову вказати фразу-пароль. PGP витягне шифрований особистий ключ, обчислить хеш-код пароля і розшифрує особистий ключ з допомогою CAST-128 з даним хеш-кодом.

Це дуже компактна і ефективна схема. Як і в будь-якій зосновані на паролях системі, захищеність усієї системи залежить від захищеності пароля. Щоб не піддатися спокусі записати пароль, користувач повинен використовувати таку парольну фразу, яку вгадати нелегко, а запам'ятати просто.

Загальна структура зв'язки відкритих ключів (рис. 3.12) дозволяє зберігати відкриті ключі інших користувачів, відомих даному. Поки що проігноруємо деякі поля, вказані в таблиці, і опишемо тільки частину з них.

- **Позначка дати-часу.** Дата і час створення даного запису.
- **Ідентифікатор ключа.** Молодші 64 розряди відкритого ключа даного запису.
- **Відкритий ключ.** Відкритий ключ даного запису.
- **Ідентифікатор користувача.** Власник даного ключа. З одним відкритим ключем можна пов'язати декілька ідентифікаторів користувача.

Зв'язка відкритих ключів може бути індексована або за полем *Ідентифікатор користувача*, або за полем *Ідентифікатор ключа*; мету такої індексації ми з'ясуємо пізніше.

Тепер ми можемо показати, як ці зв'язки ключів застосовуються при передачі і прийманні повідомлень. Для простоти в наступному прикладі ми проігноруємо стиснення і перетворення у формат radix-64. Спочатку розглянемо передачу повідомлення (рис. 3.13) і припустимо, що повідомлення повинне бути підписано і зашифровано. Джерело повідомлення PGP виконує такі кроки.

- **Створення підпису повідомлення**

PGP витягує особистий ключ відправника із зв'язки особистих ключів, використовуючи значення your_userid, що введено як ключ пошуку. Якщо відповідна команда не пропонує значення your_userid, вибирається перший особистий ключ у зв'язці.

PGP запрошує у користувача фразу-пароль, щоб розшифрувати особистий ключ. Створюється компонент підпису повідомлення.

- **2. Шифрування повідомлення**

PGP генерує сеансовий ключ і шифрує повідомлення.

PGP витягує відкритий ключ одержувача із зв'язки відкритих ключів, використовуючи значення her_userid як ключ пошуку.

Створюється компонент сеансового ключа повідомлення.

Об'єкт, що приймає PGP, виконує такі кроки (рис. 3.14).

- **Розшифрування повідомлення**

Система PGP витягує особистий ключ одержувача із зв'язки особистих ключів, використовуючи як ключ пошуку значення поля. Ідентифікатор ключа компонента сеансового ключа повідомлення. Запрошує у користувача фразу-пароль, щоб розшифрувати особистий ключ. Відкриває сеансовий ключ і розшифрує повідомлення.

- **Автентифікація повідомлення**

PGP витягує відкритий ключ відправника із зв'язки відкритих ключів, використовуючи як ключ пошуку значення поля Ідентифікатор ключа компонента підпису повідомлення та відновлює одержаний профіль повідомлення. Далі система обчислює профіль повідомлення для прийнятого повідомлення і порівнює його з профілем, що прийшов разом з повідомленням, щоб переконатися в їх ідентичності.

- **Керування відкритими ключами**

Як можна здогадатися з наведеного вище опису, PGP містить ясний і ефективний набір взаємозв'язаних функцій і форматів, що забезпечують надійну конфіденційність і засоби автентифікації. Для завершеності системи необхідно розв’язати ще одну проблему, а саме проблему керування

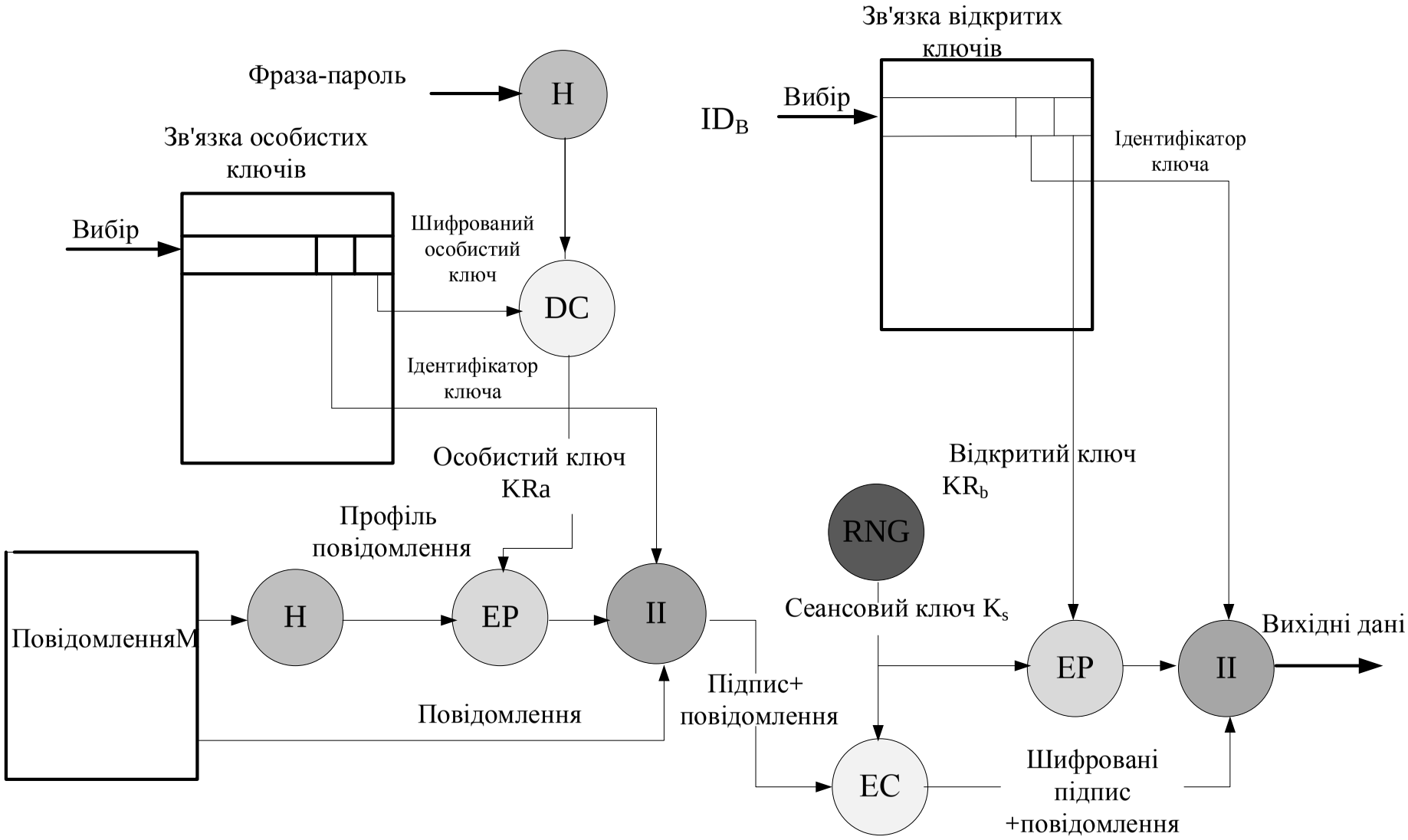


Рисунок 3.13 – Створення повідомлення PGP (від A до B без стиснення і перетворення у формат radix-64)

відкритими ключами. У документації PGP про важливість цієї проблеми говориться таке: “Проблема захисту відкритих ключів від несанкціонованого втручання є окремою і найбільш складною практичною проблемою додатків, що використовують відкриті ключі. Це “ахіллесова п'ята” криптографії з відкритим ключем, і значною мірою складність відповідного програмного забезпечення визначається складністю рішення саме цього завдання.”

PGP пропонує структуру для розв’язання цієї проблеми і ряд опцій, які можуть при цьому використовуватися. З огляду на те, що система PGP призначена для використання в найрізноманітнішому середовищі, не встановлюється ніякої жорсткої схеми керування відкритими ключами, як, наприклад, це зроблено в системі S/MIME.

Суть проблеми полягає в наступному. Користувач **A** повинен побудувати зв'язку відкритих ключів, що містить відкриті ключі інших користувачів, щоб взаємодіяти з ними, використовуючи PGP. Припустимо, що зв'язка ключів сторони **A** включає відкритий ключ, приписаний стороні **B**, але насправді власником цього ключа є сторона **C**. Така ситуація, зокрема, може мати місце, якщо учасник **A** узяв ключ з електронної дошки оголошень, яку учасник використовував для того, щоб переслати відкритий ключ, але ключ був скомпрометований якимось **C**. У результаті виникла загроза по двох напрямах. По-перше, **C** може посилати повідомлення *A*, фальсифікуючи підпис **B**, так що **A** вважатиме повідомлення такими, що прибули від **B**. По-друге, **C** може прочитати будь-яке шифроване повідомлення **A** до **B.**

Для мінімізації ризику того, що зв'язка відкритих ключів користувача містить помилкові відкриті ключі, можливо запропонувати декілька варіантів дій. Припустимо, що **A** потрібно одержати надійний відкритий ключ **B.** Пропонується декілька варіантів процедури, які при цьому можна було б використовувати.

Отримання ключа від **B** особисто (фізично). Користувач **B** може зберегти свій відкритий ключ (KUb) на дискеті і вручити цю дискету користувачу **A.** Користувач **A** потім може завантажити такий ключ з дискети в свою систему. Це дійсно безпечний шлях, але він має свої очевидні обмеження.

Перевірка ключа по телефону. Якщо **A** може розпізнати **B** по телефону, то **A** може подзвонити **B** і попросити продиктувати ключ у форматі radix-64. Ще зручніший варіант виглядає так: **B** може передати свій ключ користувачу **A** у вигляді електронного повідомлення. Користувач **A** може за допомогою PGP і з використанням SHA-1 згенерувати 160-бітовий профіль ключа і подати його в шістнадцятковому форматі; такий профіль називається “відбитком” ключа. Після цього **A** може подзвонити **B** і попросити продиктувати рядок, відповідний відбитку його ключа. Якщо два відбитки співпадуть, ключ вважається перевіреним.

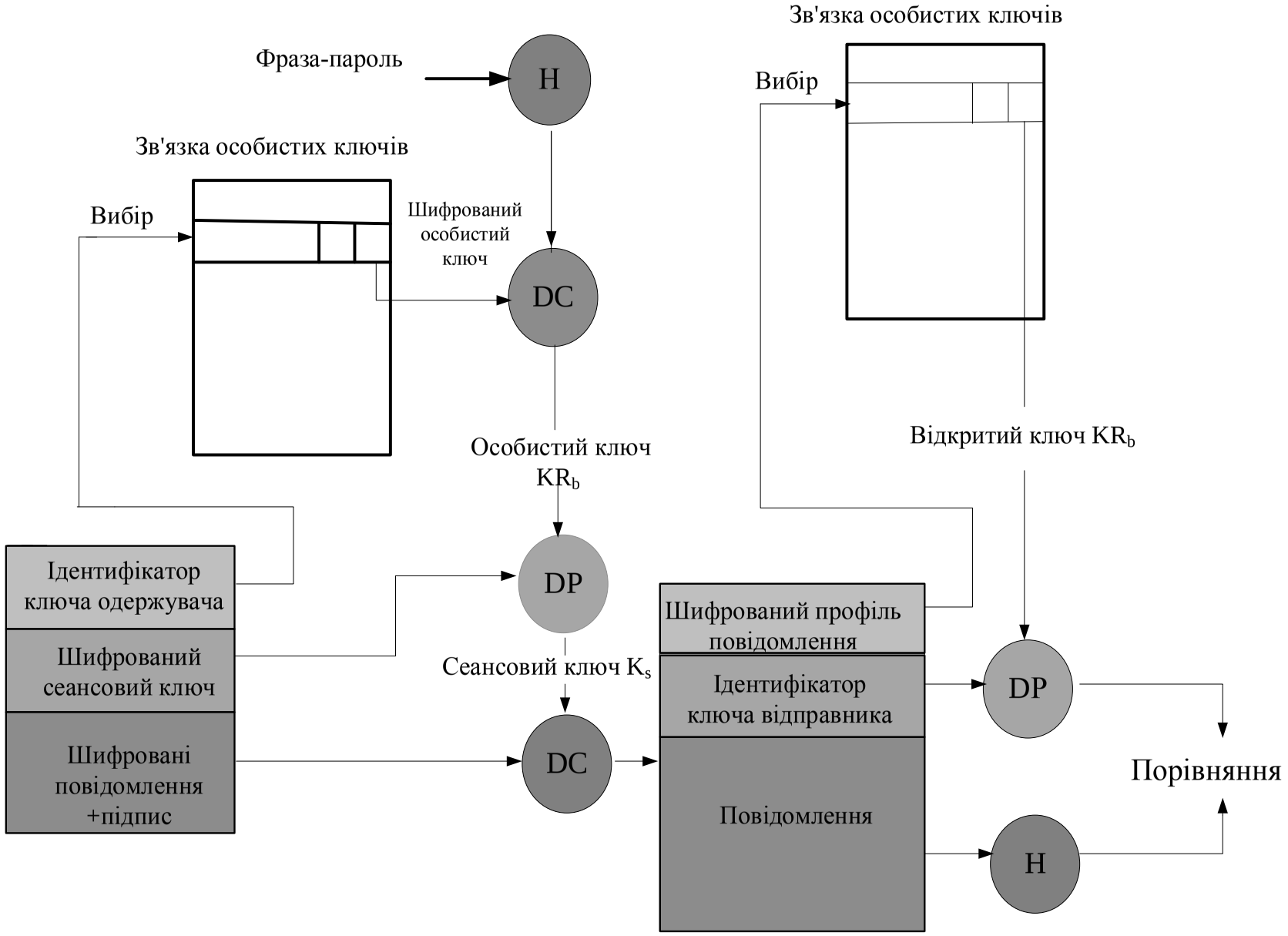


Рисунок 3.14 – Отримання повідомлення PGP (від A до B без стиснення і перетворення у формат radix-64)

Отримання відкритого ключа **B** від того, що викликає довіру обом сторонам надійного посередника **D**. Для цього постачальник **D** створює підписаний сертифікат. Такий сертифікат повинен включати відкритий ключ, час створення ключа і термін його дії. Сторона **D** генерує профіль SHA-1 цього сертифіката, шифрує його за допомогою свого особистого ключа і приєднує одержаний підпис до сертифіката. З огляду на те, що створити такий підпис може тільки **D**, ніхто інший не зможе фальсифікувати відкритий ключ і заявити, що цей ключ був підписаний стороною **D**. Підписаний сертифікат може бути доставлений безпосередньо стороні **A** стороною **B** або **D** чи виставлений на електронній дошці оголошень.

Отримання відкритого ключа **B** від надійного уповноваженого вузла сертифікації. Знову ж таки, сертифікат відкритого ключа створюється і підписується уповноваженим вузлам. Користувач **A** може потім отримати доступ до такого вузла, вказавши своє ім'я користувача, і одержати підписаний сертифікат.

У цих випадках користувач **A** повинен вже мати екземпляр відкритого ключа постачальника сертифікатів і бути упевненим, що цей ключ надійний. В кінці **A** повинен сам вирішити, наскільки надійною для нього є сторона, яка виступає в ролі постачальника.

Хоча в системі PGP не висувається ніяких вимог відносно вибору уповноважених центрів сертифікації і ступенів довіри, PGP пропонує зручні засоби використання ступенів довіри, скріплення ступенів довіри з відкритими ключами та інформацію про використання ступенів довіри.

Базова схема виглядає таким чином. Будь-який елемент зв'язки відкритих ключів є сертифікатом відкритого ключа. З кожним таким елементом зв'язується поле відповідності ключа, що задає ступінь довіри, з яким PGP вважатиме дійсним власником даного відкритого ключа вказаного користувача. Чим вище йде ступінь довіри, тим сильнішою буде прив'язка ідентифікатора користувача до даного ключа. Це поле обчислюється PGP. З кожним елементом зв'язується також певне (можливо, нульове) число підписів для даного сертифіката, які були зібрані власником зв'язки ключів. У свою чергу з кожним підписом зв'язується поле довіри підпису, що визначає ступінь, за яким PGP довіряє даному об'єкту підписувати сертифікати відкритих ключів. Значення поля відповідності ключа виводиться із сукупності значень полів довіри підпису для даного елемента зв'язки ключів. Нарешті, кожен елемент визначає відкритий ключ, що пов'язується з конкретним власником, а відповідне поле довіри власнику вказує ступінь довіри, з яким цей відкритий ключ може використовуватися для підпису інших сертифікатів відкритих ключів; цей ступінь довіри визначається і присвоюється користувачем. Значення полів довіри підпису можна розглядати як кешування копії значень полів довіри власнику інших елементів зв'язки ключів.

Три згадуваних поля містяться в структурі, яка називається байтом прапорця довіри. Вміст цього прапорця довіри для кожного з цих трьох полів показано на табл. 3.6.

Припустимо, що ми маємо справу із зв'язкою відкритих ключів користувача **A**. Операцію визначення ступеня довіри може бути описано таким чином.

1. Коли **A** додає новий відкритий ключ до низки відкритих ключів, PGP має присвоїти значення прапорцю довіри, пов'язаного з власником цього відкритого ключа. Якщо власником є **A** (тому цей відкритий ключ повинен з'явитися також у зв'язці особистих ключів), то полю довіри власника автоматично присвоюється значення "найвища довіра" (ultimate trust). Інакше PGP питає користувача **A** про те, який рівень довіри слід приписати власнику цього ключа, і **A** повинен ввести відповідне значення. Користувач може вказати, що цей власник невідомий, ненадійний, мінімально надійний або цілком надійний.

2. Коли додається новий відкритий ключ, до нього можна додати одну або кілька підписів. Пізніше можна включити й інші підписи. Коли додається підпис, PGP виконує пошук у низці відкритих ключів, щоб з'ясувати, чи значиться ім'я автора цього підпису серед відомих власників відкритих ключів. Якщо так, то значення поля OWNERTRUST цього власника присвоюється полю SIGTRUST даного підпису. Інакше відповідному полю присвоюється значення "невідомий користувач".

3. Значення поля відповідності ключа обчислюється на базі значень полів довіри підписів даного елемента зв'язки. Якщо, принаймні, один підпис має значення "найвище" (ultimate) в полі довіри підпису, то в полі відповідності ключа встановлюється значення "повне" (complete). Інакше PGP обчислює зважену суму значень полів довіри. Для підписів з максимальним рівнем довіри призначається вага 1/X, а підписам з середнім рівнем довіри призначається вага 1/Y, де X і Y є параметрами, що задаються користувачем. Якщо загальна сума ваг постачальників комбінацій "ключ/ідентифікатор користувача" досягає 1, то вважається, що відповідність надійна і для поля відповідності ключа встановлюється значення "повне" (complete). Таким чином, за відсутності найвищої довіри для повної відповідності буде потрібно, принаймні, X підписів з максимальним рівнем довіри або Y підписів з середнім рівнем довіри, або відповідна їх комбінація.

Періодично PGP виконує перевірку зв'язки відкритих ключів, щоб підтримувати узгодженість. По суті, це спадний процес. Для кожного поля OWNERTRUST при такій перевірці PGP проглядає низку, знаходить усі підписи, автором яких є даний власник, і оновлює значення полів SIGTRUST, щоб ці значення відповідали значенню поля OWNERTRUST. Увесь процес

починається з ключів, для яких вказано найвищу довіру. Після цього значення всіх полів KEYLEGIT перераховуються на базі наявних підписів.

На рис. 3.15. показано зразкову схему скріплення довіри підпису і відповідності ключа й відображено структуру зв'язки відкритих ключів. У даному випадку користувач одержав декілька відкритих ключів: частину безпосередньо від їх власників, а решту від третьої сторони, наприклад, з сервера ключів.

Вершина, позначена на рисунку "**Ви**", являє елемент зв'язки відкритих ключів, відповідний даному користувачу. Цей ключ, очевидно, відповідає власнику, тому значенням поля OWNERTRUST є найвища довіра. Для будь-якої іншої вершини поле OWNERTRUST у зв'язці ключів має значення невизначеної довіри, якщо тільки користувачем не задано якесь інше значення. У даному прикладі користувач вказав, що він завжди довіряє підписувати інші ключі користувачам **D**, **E**, **F** і **L**. Часткову довіру підписувати інші ключі одержали користувачі **A** і **B**.

Таким чином, зафарбовування або його відсутність такої для вершин на рис. 3.15. указує рівень довіри, визначеної для цих користувачів. Деревовидна структура говорить про те, якими користувачами були підписані відповідні ключі. Якщо ключ був підписаний користувачем, чий ключ також присутній у даній зв'язці ключів, від підписаного ключа до користувача, що підписав даний ключ, йде стрілка. Якщо ключ підписав користувач, який немає ключа в даній зв'язці , стрілка йде від підписаного ключа до знака питання, що означає: сторона, яка підписала ключ даному користувачу, невідома.

На рис. 3.15 проілюстровано, що всі ключі, власникам яких повністю або частково довіряє даний користувач, були підписані цим користувачем, за винятком вершини **L**. Такий підпис користувача не завжди необхідний, як тут це має місце для вершини **L**, але на практиці більшість користувачів, швидше за все, підпишуть ключі більшості власників, яким вони довіряють. Тому, наприклад, навіть якщо ключ **E** вже був підписаний надійним постачальником **F**, користувач вирішив підписати ключ **E** особисто.

Ми припускаємо, що двох частково надійних підписів достатньо для того, аби сертифікувати ключ. Отже, ключ користувача **H** розцінюється системою PGP як надійний (тобто, відповідний власнику) з огляду на те, що він підписаний користувачами **A** і **B**, яким даний користувач частково довіряє.

люч може бути визначено як надійний, якщо він підписаний однією цілком надійною або двома частково надійними сторонами, але може статися, що користувачу цього ключа не довіряють підписувати інші ключі. Наприклад, ключ **N** є надійним, оскільки він підписаний стороною **E**, якій даний користувач довіряє, але підписувати інші ключі стороні **N** не довіряють, оскільки даний користувач не присвоїв **N** відповідне значення рівня довіри.

Таблиця 3.6 – Вміст байта прапорця довіри

(а) Ступінь довіри, що приписується власнику відкритого ключа (розміщується після пакета інформації про ключ, визначається користувачем)	(б) Ступінь довіри, що приписується парі "відкритий ключ/ідентифікатор користувача" (розміщується після ідентифікатора користувача, обчислюється PGP)	(в) Ступінь довіри, що приписується підпису (розміщується після пакета підписів, кешірувана копія значення поля OWNERTRUST для даного постачальника підпису)
Поле OWNERTRUST: - невизначена довіра; - невідомий користувач; - мінімальний рівень довіри для підпису; - середній рівень довіри для підпису; - максимальний рівень довіри для підпису; - даний ключ присутній у зв'язці секретних ключів (найвища довіра) Біт BUCKSTOP - встановлюється, якщо даний ключ присутній у зв'язці секретних ключів	Поле KEYLEGIT: - невідома або невизначена відповідність; - ненадійна відповідність власнику ключа; - мінімально надійна відповідність власнику ключа; - повна відповідність власнику ключа Біт WARNONLY - встановлюється, якщо користувач бажає одержати тільки попередження, коли для шифрування використовується не цілком підтверджений ключ	Поле SIGTRUST: - невизначена довіра; - невідомий користувач; - мінімальний рівень довіри для підпису; - середній рівень довіри для підпису; - максимальний рівень довіри для підпису; - даний ключ присутній у зв'язці секретних ключів (найвища довіра) Біт CONTIG - встановлюється, якщо підпис сходить по безперервному ланцюжку надійних сертифікатів до власника зв'язки ключів з найвищою довірою

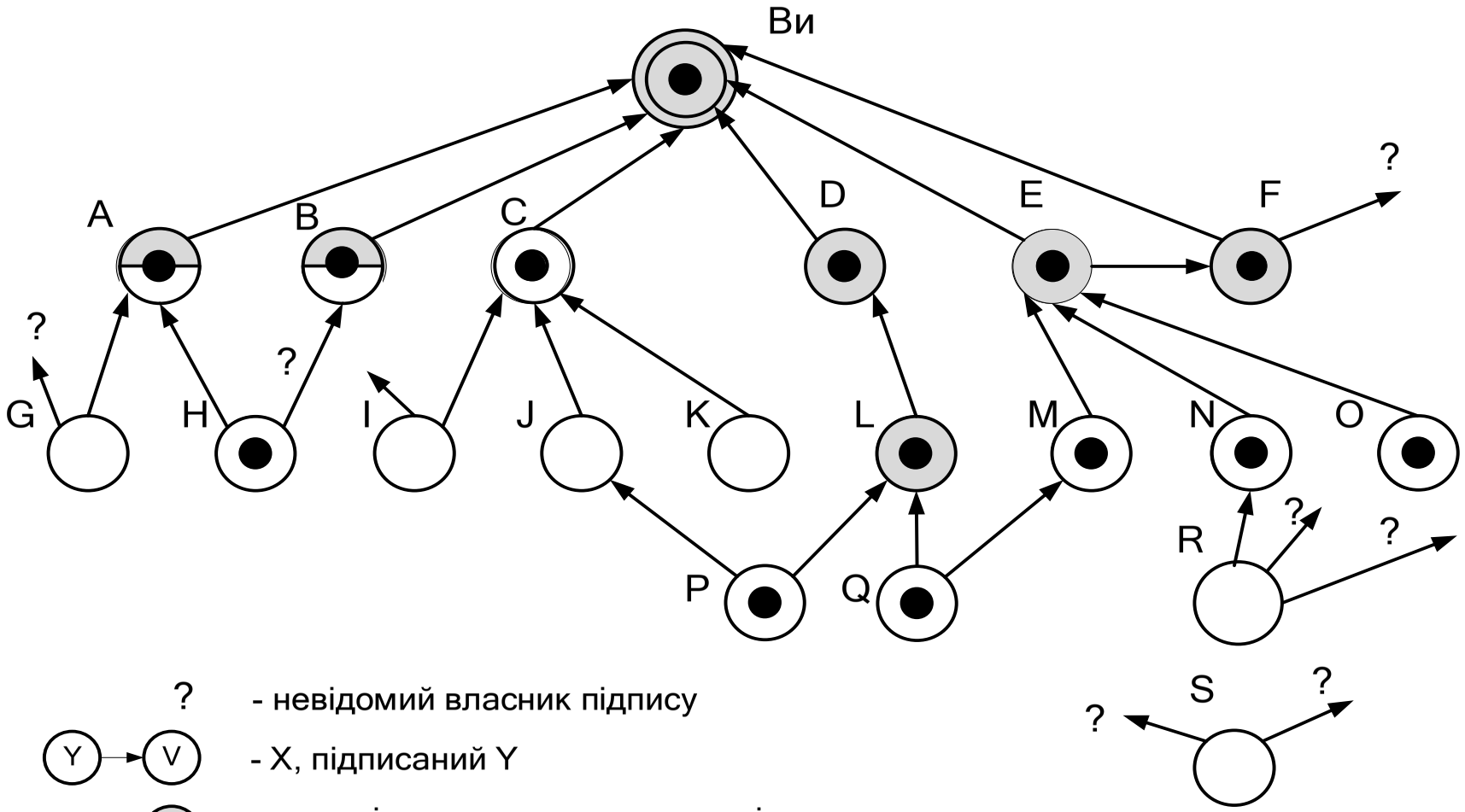


Рисунок 3.15 – Приклад моделі довіри PGP

Тому, хоча ключ **R** і підписано стороною **N**, система PGP не вважає його **R** надійним. Така ситуація має абсолютно певний сенс. Якщо ви хочете послати секретне повідомлення деякому адресату, зовсім не обов'язково, щоб ви довіряли цьому адресату якоюсь мірою. Необхідно лише, щоб ви були упевнені в тому, що маєте надійний відкритий ключ відповідного користувача.

На рис. 3.15 показано також приклад окремої "вершини-сироти" **S** з двома невідомими підписами. Такий ключ міг бути одержаний з сервера ключів. PGP не може вважати цей ключ надійним тільки тому, що цей ключ прийшов з сервера, який має хорошу репутацію. Користувач повинен оголосити цей ключ надійним, підписавши його особисто або повідомивши PGP про те, що він готовий повністю довіряти одній із сторін, що вже підписали даний ключ.

Таким чином, з одним відкритим ключем у зв'язці відкритих ключів можна зв'язати кілька ідентифікаторів користувачів. Це може мати місце, наприклад, у тому випадку, коли якась сторона змінила своє ім'я або виступає за допомогою підписів під багатьма іменами, указуючи для себе, наприклад, різні адреси електронної пошти. Отже, відкритий ключ можна розглядати як корінь якогось дерева. Відкритий ключ має деяке число пов'язаних з ним ідентифікаторів користувачів з рядом підписів, що асоціюються з кожним із цих ідентифікаторів. Прив'язка конкретного ідентифікатора користувача до ключа залежить від підписів, що пов'язуються з цим ідентифікатором користувача, так що ступінь довіри даному ключу (для використання цього ключа в цілях підписання інших ключів) виявляється функцією всіх відповідних підписів.

3.3. Захист електронної пошти. Система S/MIME

Для того щоб зрозуміти структуру протоколу (системи) S/MIME, розглянемо формати електронної пошти RFC 822 і MIME, які є основою цього протоколу.

3.3.1. Формат поштового повідомлення (RFC-822)

Документ RFC 822 визначає формат текстових повідомлень, які пересилаються електронною поштою. Він став стандартом для текстових поштових повідомлень в Інтернет і широко застосовується дотепер. У контексті стандарту RFC 822 повідомлення розглядаються як документи, що мають конверт і вміст. Конверт містить всю необхідну інформацію, потрібну для того, щоб виконати пересилання і доставку. Вміст є об'єктом, який доставляється одержувачу. Стандарт RFC 822 стосується тільки надання вмісту. Проте стандарт вмісту включає цілий ряд полів заголовка, які можуть

використовуватися поштовою системою для створення конверта. Цей стандарт дозволяє спростити програмі електронної пошти отримання такої інформації.

Загальна структура повідомлення відповідного стандарту RFC 822 дуже проста. Повідомлення формується з певного числа рядків заголовка, за яким йде текст необмеженої довжини (тіло повідомлення). Заголовок повідомлення відокремлюється від тіла повідомлення порожнім рядком. Іншими словами, повідомлення є потоком тексту ASCII, де всі рядки до першого порожнього рядка вважаються рядками заголовка, що використовуються частиною агента користувача поштової системи. Рядок заголовка зазвичай складається з ключового слова, що завершується двокрапкою, за якою йдуть ключові параметри, причому довгі рядки формат дозволяє розбивати на кілька рядків. Найчастіше використовуються такі ключові слова, як From (Від), To (До), Subject (Тема) і Date (Дата). Ось приклад повідомлення.

Date: Tue, 16 Jan 1998 10:37:17 (EST)
From: "SSemenov" <ss@ukr.net>
Subject: Синтаксис RFC 822
To: Sanja@mail.ru
c: Irina@rambler.ru

Привітання. Цей розділ починає безпосередньо тіло повідомлення, яке відокремлюється від заголовка повідомлення порожнім рядком.

Ще одним полем, яке часто присутнє в заголовках повідомлень RFC 822, є поле Message-ID (Ідентифікатор повідомлення). Це поле містить унікальний ідентифікатор, що пов'язується з даним повідомленням.

3.3.2. Багатоцільові розширення електронної пошти. Стандарт MIME

Стандарт MIME є розширенням базового стандарту RFC 822, покликаним розв’язати деякі проблеми і подолати обмеження протоколу SMTP (Simple Mail Transfer Protocol – простий протокол передачі пошти) або деякого іншого протоколу передачі пошти і RFC 822. Відомий наступний список обмежень схеми SMTP/822.

SMTP не дозволяє передавати виконувані файли й інші об'єкти в двійковому форматі. Існує ряд схем перетворення двійкових файлів у текстові (до них належить й популярна схема UUencode/UUdecode для UNIX), які потім можуть бути використані різними поштовими системами SMTP. Проте жодна з таких схем не є ані формальним, ані фактичним стандартом.

SMTP не дозволяє передавати текстові дані, що включають символи національних мов, оскільки такі символи подаються 8-бітовими кодами з

десятковими значеннями від 128 і вище, а SMTP обмежує можливості передачі даних 7-бітовими кодами ASCII.

Сервери SMTP можуть відкинути електронне повідомлення, що перевищує певні розміри.

Шлюзи SMTP, що виконують трансляцію кодів ASCII в коди EBCDIC і назад, можуть мати різні таблиці перекладу, що виливається в проблеми трансляції.

Шлюзи SMTP, пов'язані з мережами електронної пошти, які використовують протокол X.400, не можуть обробити нетекстові дані, що включаються в повідомлення стандарту X.400.

Деякі реалізації протоколу SMTP не цілком суворо дотримуються стандарту SMTP, визначеного в документі RFC 821. При цьому виникають такі типові проблеми:

- зникнення, додавання або зміна порядку символів повернення каретки і переходу на новий рядок;
- усікання або розрив рядків, що мають довжину більше 76 символів;
- видалення пропусків, що є в кінці рядка (символів табуляції і пропуску);
- доповнення пропусками рядків повідомлень до однакової довжини;
- перетворення символів табуляції в набори декількох символів пропуску.

Стандарт MIME повинен розв'язуваті ці проблеми, зберігаючи сумісність з існуючими реалізаціями RFC 822. Відповідні технічні специфікації наводяться в документах RFC з номерами від 2045 до 2049.

Технічні специфікації MIME включають такі елементи.

Визначається п'ять нових полів заголовка повідомлення, які можуть включатися до заголовка RFC 822. Ці поля несуть у собі інформацію про вміст повідомлення.

Визначається декілька форматів вмісту, які задають стандарти уявлення документів мультимедіа в повідомленнях електронної пошти.

Визначаються стандарти кодувань переданих даних, що дозволяють захистити вміст повідомлення від зміни при здійсненні поштовими системами перетворення переданих даних з одного формату до іншого.

Вище вже згадувалися п'ять полів заголовка повідомлення. Далі ми опишемо формати вмісту й кодування переданих даних.

Нижче перераховано п'ять полів заголовка, які визначаються стандартом MIME.

MIME-Version (версія MIME). Відповідний параметр повинен мати значення 1.0. Це поле указує, що повідомлення відповідає стандартам RFC 2045 і 2046.

Content-Type (тип вмісту). Описує дані, поміщені в тіло повідомлення, достатньо детально для того, щоб агент одержувача зміг вибрати відповідний агент або механізм, що дозволяє надати одержані дані користувачу або обробити їх якимсь іншим відповідним чином.

Content-Transfer-Encoding (кодування переданого вмісту). Указує тип перетворення, що використалося для того, щоб подати тіло повідомлення у вигляді, прийнятному для пересилання поштою.

Content-ID) (ідентифікатор вмісту). Служить для того, щоб унікальним чином ідентифікувати об'єкти MIME серед множини контекстів.

Content-Description (опис вмісту). Текстовий опис об'єкта у вмісті повідомлення; корисно тоді, коли об'єкт має форму, недоступну для прочитання (наприклад, звукові дані).

Будь-яке або всі ці поля можуть бути присутніми в звичайному заголовку RFC 822. Будь-яка реалізація, як мінімум, повинна підтримувати обробку полів **MIME-Version, Content-Type і Content-Transfer-Encoding,** а поля **Content-ID** і **Content-Description** є опціями і в реалізації одержувача можуть ігноруватися.

Чимала частина специфікацій MIME торкається визначення множини допустимих типів вмісту. Це наслідок необхідності стандартизації шляхів передачі інформації, поданої у найрізніших форматах у середовищі мультимедіа.

У табл. 3.7 подано список типів вмісту, описаних в RFC 2046. Існує сім основних типів вмісту і в загальній сумі 15 підтипів. Загалом, тип вмісту указує принциповий тип даних, а підтип визначає конкретний формат, який використовується для подання цього типу даних.

Для типу **text** тіла повідомлення не вимагається ніякого спеціального програмного забезпечення, щоб побачити такий текст, якщо не враховувати кодування для набору символів, за допомогою яких цей текст був написаний. Головним підтипом даного типу є **plain text**, що означає просто рядок символів ASCII або символів ISO 8859. Підтип **enriched** дає можливість використовувати гнучкіший формат.

Тип **multipart** указує на те, що тіло повідомлення складається з декількох незалежних частин. Поле **Content-Type** заголовка повідомлення включає параметр *boundary* (межа), який задає роздільник, що відокремлює частини тіла повідомлення. Цей роздільник не повинен бути присутнім у жодній з частин повідомлення.

Кожен такий роздільник повинен починатися з нового рядка і формується з двох дефісів, за якими йде значення роздільника. Роздільник, що завершує, указує кінець останньої частини повідомлення і має ще суфікс, який складається з двох дефісів. Усередині кожної частини може бути присутнім необов'язковий звичайний заголовок MIME.

Таблиця 3.7 – Типи вмісту MIME

Тип	Підтип	Опис
1	2	3
Text (текст)	Plain (простий)	Неформатований текст; може бути в кодуванні ASCII або ISO 8859
	Enriched (розширений)	Забезпечує велику гнучкість формату
Multipart (багатоком-понентний)	Mixed (змішаний)	Різні частини незалежні, але повинні передаватися разом. Вони повинні бути подані одержувачу в тому порядку, в якому вони з'являються в поштовому повідомленні
	Parallel (паралельний)	Відрізняється від змішаного тільки тим, що не визначається порядок, в якому частини повідомлення повинні бути подані одержувачу
	Alternative (альтернативний)	Різні частини є альтернативними варіантами однієї і тієї ж інформації. Вони розташовуються у порядку зростання точності відповідності оригіналу, і поштова система одержувача повинна вибрати для відображення користувачу "кращий" варіант
	Digest (реферативний)	Подібний змішаному, але за умовчанням для поля тип/підтип кожної частини вибира-ється message/rfc822
Message (повідомлення)	rfc822	Тіло повідомлення саме є інкапсульованим повідомленням у форматі RFC 822
	Partial (фрагментарний)	Служить для того, щоб вирішити фрагментацію великих поштових об'єктів деяким прозорим для одержувача способом
	External-body (зовнішний)	Містить указівник на об'єкт, що існує десь в іншому місці

Продовження таблиці 3.7

1	2	3
Image (зображення)	Jpeg gif	Зображення у форматі JPEG, кодування JFIF Зображення у форматі GIF
Video (відео)	mpeg	Формат MPEG
Audio (звук)	Basic (основний)	Одноканальне 8-бітове кодування стандарту ISDN з частотою вибірки 8 КГц, виконано за законом компанування з мю-характеристикою
Application (додаток)	PostScript Octet-stream (потік байтів)	Формат Adobe PostScript Двійкові дані загального вигляду, що складаються з 8-бітових байтів

Нижче наданий простий приклад складеного повідомлення, що включає дві частини, кожна з яких є простим текстом (це приклад з документа RFC 2046 з перекладом пояснень).

From: Irina Il'ina < ilirvi@rambler.ru>
To: Natalija Lubchenko <n_lubchenko@ukr.net>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

Це преамбула. Вона ігнорується, але тут створювач повідомлення може помістити пояснення для читача, система якого не відповідає стандарту MIME.

--simple boundary

Це текст ASCII, що неявно типізується. Він **не закінчується** символом обриву рядка.

--simple boundary

Content-type: text/plain; charset=us-ascii

Це текст ASCII, що явно типізується. Він **закінчується** символом обриву рядка

--simple boundary--

Це епілог. Він теж ігнорується**.**

Існує чотири підтипи для типу **multipart** повідомлення, і всі вони мають загальний синтаксис. Підтип **multipart/mixed** використовується за наявності декількох незалежних частин, які повинні бути зв'язані в певному порядку. Для

підтипу **multipart/parallel** порядок частин не істотний. Якщо система одержувача дозволяє, всі частини повідомлення можуть бути подані паралельно. Наприклад, зображення або текстова частина можуть супроводжуватися голосовим коментарем, який програється, поки відображається зображення або текст.

Для підтипу **multipart/alternative** різні частини є різними поданнями однієї і тієї ж інформації. Розглянемо наступний приклад.

From: Irina Il'ina < ilirvi@rambler.ru>
To: Natalija Lubchenko <n_lubchenko@ukr.net>
Subject: Formatted text mail
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary=boundary42
--boundary42
Content-Type: text/plain; charset=us-ascii
... тут розміщується варіант повідомлення у вигляді простого тексту ...
--boundary4 2
Content-Type: text/enriched
... те ж повідомлення у вигляді тексту в розширеному форматі RFC 1896
--boundary42--

У цьому підтипі частини повідомлення впорядковані за збільшенням їх переваги. Що стосується даного прикладу, то, якщо приймаюча система здатна відображати повідомлення в розширеному текстовому форматі, так він і буде відображений, інакше використовується простий текстовий формат.

Підтип **multipart/digest** застосовується тоді, коли кожна частина тіла повідомлення інтерпретується як повідомлення RFC 822 зі своїм заголовком. Цей підтип дозволяє створювати повідомлення, частини якого є окремими повідомленнями. Наприклад, модератор групи може збирати повідомлення електронної пошти учасників групи, зв'язувати ці повідомлення і пересилати їх інкапсульованими в одному повідомленні MIME.

Тип **message** забезпечує ряд важливих можливостей MIME. Підтип **message/rfc822** указує на те, що тіло надає повне повідомлення, яке включає заголовок і тіло повідомлення. Не дивлячись на назву цього підтипу, інкапсульоване повідомлення може бути не тільки простим повідомленням RFC 822, але і будь-яким повідомленням MIME.

Підтип **message/partial** дає можливість розділити велике повідомлення на декілька частин, які повинні знову бути зібрані в системі одержувача. Для цього

підтипу в полі **Content-Type: Message/Partial** указуються три параметри: ідентифікатор, загальний для всіх фрагментів цього повідомлення, порядковий номер, унікальний для кожного фрагмента, і загальне число фрагментів.

Підтип **message/external-body** указує на те, що фактичних даних, які повинні супроводжувати це повідомлення, в тілі повідомлення немає. Натомість тіло повідомлення містить інформацію, необхідну для того, щоб дістати доступ до цих даних. Як і у випадку з іншими типами повідомлення, підтип **message/external-body** має зовнішній заголовок й інкапсульоване повідомлення з власним заголовком. Необхідним єдиним полем ув зовнішньому заголовку є поле **Content-Type**, що ідентифікує це повідомлення як повідомлення підтипу **message/external-body**. Внутрішній заголовок є заголовком повідомлення для інкапсульованого повідомлення. Поле **Content-Type** в зовнішньому заголовку повинне включати параметр типу доступу, який указує метод доступу, наприклад FTP (протокол передачі файлів).

Тип **application** посилається на інші види даних, зазвичай або на дані, які не можуть бути інтерпретовані в двійковому форматі, або на інформацію, яка повинна оброблятися поштовим застосуванням.

Іншим основним компонентом специфікацій MIME, окрім типу змісту, є визначення кодування переданого тіла повідомлення. Мета вказівки кодування – виключення спотворень при пересиланні через якомога більше число різних середовищ.

Стандарт MIME визначає два методи кодування даних. Поле **Content-Transfer-Encoding** може насправді приймати шість різних значень, указаних в табл. 3.8.

Ще одним значенням поля Content-Transfer-Encoding є **x-token**, яке говорить про те, що діє деяка інша схема кодування, для якої повинно бути вказано ім'я. Це може бути спеціальна схема кодування постачальника або конкретного застосування.

Визначальними схемами кодування є **quoted-printable** і **base64**. Ці дві схеми пропонуються для того, щоб забезпечити можливість вибору з варіантів передачі, один з яких, по суті, є прийнятною для читання людиною, а інший – безпечний для пересиланні всіх типів даних способом і, до того ж, достатньо компактний. Кодування **quoted-printab**le доцільно використовувати тоді, коли більшість байтів даних відповідають друкованим символам ASCII. По суті, вона переводить недруковані символи в шістнадцяткові подання їх кодів і вставляє оборотні (м'які) переходи на новий рядок, щоб обмежити довжини рядків до 76 символів. Кодування **base64**, відоме також під назвою **radix-64**, є загальноприйнятим перетворенням довільних даних з двійкового формату в такий формат, який не спотворюється внаслідок обробки даних програмами пересиланні пошти. Це кодування використовується також у PGP.

Таблиця 3.8 – Кодування MIME

Розмір	Методи кодування MIME
7bit	Усі дані подаються короткими рядками символів ASCII
8bit	Усі рядки є короткими, але можуть містити символи, що не є символами ASCII (байти з ненульовими старшими бітами)
binary	Можуть бути присутніми не тільки символи, що не є символами ASCII, але і рядки не обов'язково повинні бути достатньо короткими для транспорту SMTP
quoted-printable	Дані кодуються таким чином: якщо початкові дані є основним текстом ASCII, то в кодованій формі вони залишаються значною мірою розпізнаваними людиною
base64	Дані кодуються відображенням 6-бітових блоків вхідних даних у 8-бітові блоки вихідних, всі вони є друкованими символами ASCII
x-token	Нестандартне кодування з вказаним ім'ям

На рис. 3.16 схематично подане складне багатокомпонентне повідомлення.

Це повідомлення складається з п'яти частин, які повинні відображатися послідовно: дві увідні частини у вигляді простого тексту, вкладене багатокомпонентне повідомлення, частина у вигляді тексту розширеного формату і завершальне інкапсульоване текстове повідомлення, в якому використовуються символи, що не є символами ASCII.

Вкладене багатокомпонентне повідомлення складається з двох частин, які повинні відображатися паралельно: зображення і звуковий фрагмент.

Важливим поняттям в MIME і S/MIME є поняття канонічної форми.

Канонічна форма є форматом, відповідним даному типу вмісту і стандартизованим для використання при обміні між системами. В цьому відношенні канонічна форма відрізняється від власної форми вмісту, яка може залежати від даної конкретної системи. Табл. 3.9, з документа RFC 2049, повинна допомогти зрозуміти суть цього поняття.

3.3.3. Повідомлення S/MIME

Адміністратори багатьох організацій прагнуть захистити повідомлення електронної пошти своїх співробітників. Secure MIME (S/MIME – Secure/Multipurpose Internet Mail Extension – захищені багатоцільові розширення електронної пошти) – рішення безпеки, реалізоване в більшості

From: Irina Il'ina < ilirvi@rambler.ru>

To: Natalija Lubchenko <n_lubchenko@ukr.net>

Subject: Приклад багатокомпонентного повідомлення

Content-Type: multipart/mixed;

boundary=unique-boundary-l

Це розділ преамбули багатокомпонентного повідомлення. Одержувач, що розпізнає даний формат, повинен ігнорувати цю преамбулу. Прочитавши цей текст, у вас, можливо, з'явиться бажання використовувати поштову програму, здатну правильно відображати багатокомпонентні повідомлення**.**

--unique-boundary-1

Тут розміщується деякий текст...

(Відзначимо, що попередній порожній рядок означає відсутність полів заголовка для цього тексту, а також те, що текст складається з символів US-ASCII. Останнє можна було б вказати явно, як це зроблено в наступній частині повідомлення.)

--unique-boundary-1

Content-type: text/plain; charset=US-ASCII

Цей текст міг би бути частиною попередньої частини, але він ілюструє можливість явної вказівки типу частини вмісту порівняно з неявним.

--unique-boundary-1

Content-Type: multipart/parallel; boundary=unique-boundary-2

--unique-boundary-2

Content-Type: audio/basic Content-Transfer-Encoding, base64

Тут розміщуються кодовані у форматі base64 дані, що являють одноканальний 8-бітовий цифровий запис звуку з частотою вибірки 8000 Гц, виконану за законом компандування з мю-характеристикою...

--unique-boundary-2 Content-Type: image/jpeg Content-Transfer-Encoding: base64

Тут розміщується кодоване у форматі base64 зображення ...

--unique-boundary-2

--unique-boundary-1

Content-type: text/enriched

Це текст у <bold><italic> розширеному форматі, визначеному в RFC 1896.

--unique-boundary-1

Content-Type: message/rfc822

From: (mailbox in US-ASCII)

To: (address in US-ASCII)

Subject: (subject in US-ASCII)

Content-Type: Text/plain; charset=ISO-8859-l

Content-Transfer-Encoding: Quoted-printable

Тут розміщується текст у кодах ISO-8859-1...

--unique-boundary-1—

Рисунок 3.16 – Приблизна структура повідомлення MIME

Таблиця 3.9 – Власна і канонічна форми

Форма	Опис
Власна форма	Тіло повідомлення, яке має бути передано, створюється у власному форматі відповідної системи. Використовуються власні набори символів і, можливо, локальні угоди про символи закінчення рядків. Тіло повідомлення може бути текстовим файлом UNIX, растровим зображенням Sun, індексованим файлом VMS, звуковими даними в залежному від системи форматі, що зберігаються тільки в пам'яті, або чимось іншим, що відповідає локальній моделі подання деякої інформації. Головне, що дані створюються в деякій "рідній" формі, яка задається типом носія
Канонічна форма	Все тіло повідомлення, включаючи додаткову інформацію, наприклад, довжину записів і, можливо, інформацію про атрибути файлу, переводиться в універсальну канонічну форму. Найуживаніша канонічна форма тіла повідомлення визначається конкретним типом тіла повідомлення, а також взаємозв'язаними з ним атрибутами. Перетворення в правильну канонічну форму може означати перетворення набору символів, трансформацію звукових даних, стиснення або якісь інші дії, застосовні до конкретного типу вмісту. Якщо використовується перетворення набору символів, слід чітко розуміти семантику подання вмісту, яка може істотно ускладнювати будь-які перетворення наборів символів (наприклад, з урахуванням синтаксично важливих символів у тексті, підтип якого відрізняється від простого)

сучасних поштових програм, яке допомогає зберегти конфіденційність, цілісність поштових повідомлень і перевірити достовірність даних. S/MIME забезпечує крізний захист – не тільки в процесі пересилання повідомлень, але й при зберіганні в базі даних поштового сервера.

Система S/MIME є удосконаленням з погляду захисту стандарту формату MIME електронної пошти в мережі Інтернет, що базується на використанні технології RSA Data Security. Хоча PGP і S/MIME є стандартами IETF, існують

підстави вважати, що S/MIME стане стандартом комерційного і промислового використання, тоді як PGP залишиться альтернативою для захисту особистої електронної пошти більшості індивідуальних користувачів.

З погляду загальних функціональних можливостей S/MIME і PGP дуже схожі. Обидві системи пропонують можливість підписувати та/або шифрувати повідомлення.

S/MIME забезпечує можливість використання таких функцій:

- **Упаковані дані.** Складаються з шифрованого вмісту будь-якого типу і ключів шифрування вмісту для одного або більшого числа одержувачів.
- **Підписані дані**. Цифровий підпис формується за допомогою обчислення профілю для вмісту повідомлення, яке вимагає підпису, і потім шифрується з використанням особистого ключа сторони, яка підписує цей вміст. Після цього вміст разом з підписом переводяться у формат **base64.** Повідомлення з підписаними даними може бути проглянуто тільки одержувачем, що має в своєму розпорядженні можливості S/MIME.
- **Відкриті підписані дані**. Як і у випадку підписаних даних, формується цифровий підпис вмісту. Проте в даному разі з використанням **base64** кодується тільки цифровий підпис. У результаті одержувачі без можливостей S/MIME зможуть проглянути вміст повідомлення, але не зможуть перевірити підпис.
- **Підписані й упаковані дані**. Можливості підпису і упаковки можуть бути вкладені одна в іншу, так що шифровані дані можуть бути підписані, а підписані або відкриті підписані дані можуть бути зашифровані.

S/MIME забезпечує перевірку достовірності даних, конфіденційність і цілісність повідомлень у форматі MIME. Робоча група Internet Engineering Task Force (IETF <http://www.ietf.org>) стандартизувала S/MIME 3.0 в документах Request for Comments (RFC) з номерами від 2632 до 2634.

S/MIME − відмінний приклад гібридного рішення шифрування, в якому об'єднані достоїнства симетричного і асиметричного шифрів і функцій хешування.

На рис. 3.17 показано етапи типового сценарію використання S/MIME.

Відправник хоче послати безпечне повідомлення (з гарантованою конфіденційністю, цілісністю, достовірністю даних) одержувачу.

Обмін S/MIME складається з таких кроків:

Відправник створює цифрову сигнатуру для повідомлення з використанням свого приватного ключа.

Відправник використовує єдиний ключ симетричного шифрування для шифрування повідомлення.

Щоб сформувати безпечний канал, в якому буде захищена конфіденційність ключа шифрування при його передачі через загальнодоступний канал зв'язку, відправник використовує відкритий ключ одержувача (з сертифіката одержувача), щоб зашифрувати ключ шифрування.

Результат цього етапу – захищена скринька (lockbox), в якій міститься зашифрована копія ключа шифрування.

Одержувач використовує свій приватний ключ, щоб розшифрувати захищену скриньку. В процесі розшифрування формується єдиний ключ шифрування. Далі одержувач розшифровує повідомлення за допомогою єдиного ключа шифрування. Тепер він може прочитати повідомлення.

Одержувач використовує відкритий ключ відправника для перевірки достовірності і цілісності повідомлення та засвідчує цифрову сигнатуру, яка є в сертифікаті відправника. S/MIME забезпечує стійке крізне шифрування поштових повідомлень. Це значить, що при використанні S/MIME повідомлення зашифровані не тільки на етапі пересилання, але і при зберіганні в локальній особистій теці Microsoft Outlook (.pst) і поштовій скриньці Microsoft Exchange Server

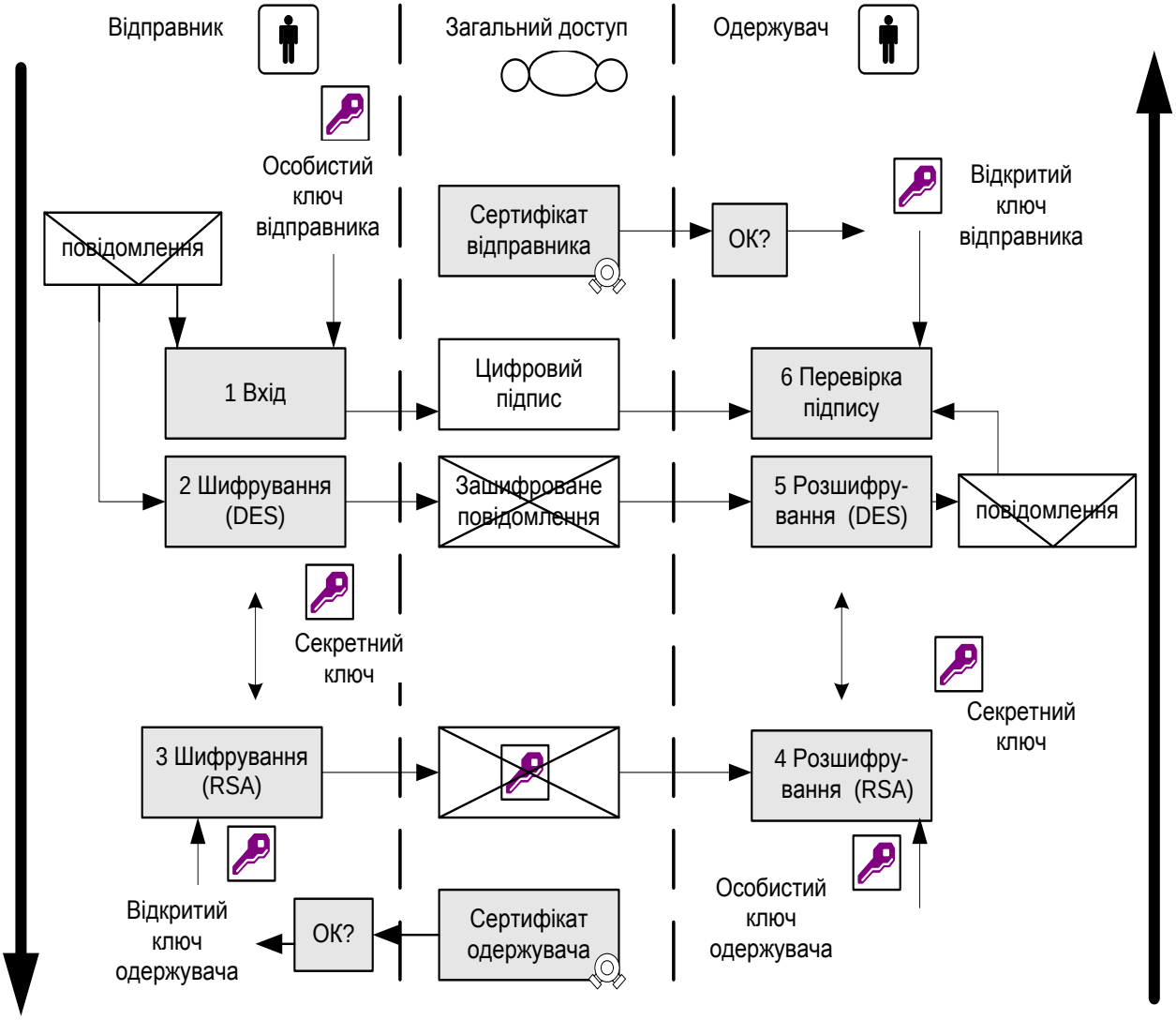


Рис. 3.17 – Схема використання S/MIME для автентифікації і забезпечення конфіденційності електронної пошти

Існують чотири поштових клієнти Microsoft: Microsoft Office Outlook 2003, Outlook Express 6.0, Outlook Web Access (OWA) і Pocket Outlook. Поштові клієнти Outlook і Outlook Express вже давно сумісні з S/MIME. З виходом Exchange Server 2003 забезпечена сумісність S/MIME з OWA. У підготовленому до випуску пакеті доповнень Messaging and Security Feature Pack for Windows Mobile 5.0 реалізовано функції S/MIME для Pocket Outlook. Pocket Outlook входить до складу операційної системи Windows Mobile (WM) і дозволяє користувачам звертатися до пошти зі смартфонів і пристроїв Pocket PC на базі Windows Mobile. Детальніше про WM можна прочитати за адресою <http://www.microsoft.com/windowsmobile/business/5/default.mspx>.

Як видно з таблиці, якнайповнішу функціональність S/MIME має в своєму розпорядженні Outlook 2003, в якому реалізовані служби S/MIME Enhanced Security Services (ESS) і зручні функції керування сертифікатами.

З Outlook 2003 легко опублікувати сертифікат у глобальному списку адрес Exchange Global Address List (GAL) або запитати новий сертифікат з комерційного Центру сертифікації (CA), такого, як VeriSign. Щоб опублікувати особисті сертифікати у GAL, потрібно натиснути на пункті Options в меню Tools, а потім на кнопку Publish to GAL на вкладці Security (рис. 3.18). Завдяки публікації сертифікатів в GAL, інші користувачі можуть завантажити їх і відправити підписані та/або зашифровані повідомлення. Після натиснення на кнопку Get a Digital ID Outlook 2003 перенаправляє користувача на Web-вузол Office Marketplace, де можна запитати сертифікат S/MIME від комерційного CA.

Outlook 2003 ESS забезпечує безпечне повідомлення. Таким чином, засвідчується цілісність прочитання, і відправник одержує криптографічний доказ, що цільовий одержувач дійсно прочитав і перевірив підписане повідомлення.

У S/MIME застосовуються цифрові сигнатури двох типів – явні і неявні. У явно підписаному повідомленні цифровий підпис відокремлений від підписаних даних. У неявно підписаному повідомленні цифровий підпис і повідомлення об'єднані в одному двійковому файлі. Тільки S/MIME-сумісні поштові клієнти можуть посилати S/MIME-підписані (явно або неявно) повідомлення. Поштові клієнти S/MIME придатні для читання як явно, так і неявно підписаних повідомлень. Поштові клієнти без функцій S/MIME можуть читати тільки явно підписані повідомлення. В цілому, якщо рівень функціональності S/MIME одержувача невідомий, слід посилати явно підписані повідомлення. Якщо рівень функціональності S/MIME одержувача відомий, то можна використовувати як явні, так і неявні підписи. Перевага неявного підпису – підвищена стійкість до перетворення поштовим шлюзом і серверами ретрансляції, які можуть зробити недійсною цифрову сигнатуру поштового

повідомлення. Як Outlook 2003, так і Outlook Express 6.0 підтримують явні і неявні підписи; у OWA застосовуються тільки явні підписи.

Для налаштування S/MIME в Outlook 2003 (продукти з найбільш вичерпним набором функцій S/MIME) слід перейти на вкладку Security (див. рис. 3.18) і натиснути на кнопку Settings. З'являється діалогове вікно Change Security Settings (рис. 3.19).

У поле Security Settings Name необхідно ввести або вибрати ім'я (у даному випадку, Jan De Clercq's S/MIME Settings). Потім зі списку Cryptography Format, що розкривається, потрібно вибрати пункт S/MIME і встановити прапорці

Рисунок 3.18 – Ілюстрація вкладки Security

Default Security Setting for this cryptographic message format і Default Security Setting for all cryptographic messages. У полі Signing Certificate необхідно вказати підписний сертифікат, а потім призначити алгоритм хешування (РЕКОМЕНДОВАНО SHA1). Виберіть шифрований сертифікат і задайте алгоритм шифрування (РЕКОМЕНДОВАНО 3DES). Встановивши

прапорець Send these certificates with signed messages, потрібно натиснути на кнопку OK.

Рис. 3.19 – Ілюстрація діалогового вікна Change Security Settings

Тепер можна повернутися до вкладки Security і задати вибрану за умовчанням поведінку S/MIME, яка застосовуватиметься до всіх повідомлень, користувачем, що відправляється. Як показано на рис. 3.18, за умовчанням у повідомленнях використовується явний підпис. Можна шифрувати та/або підписувати всі повідомлення, що випливають, завжди відправляти повідомлення з явним підписом та/або завжди запрошувати підтвердження приймання підписаних повідомлень.

Користувачі можуть змінювати стандартні параметри для окремих повідомлень. У вікні Message (рис. 3.20) вони можуть клацнути на піктограмі Encrypt або Sign або на пункті Options для доступу до функцій шифрування і підписки з розділу Security Settings в діалоговому вікні Options.

Користувачам необхідно знати, що, відправляючи зашифроване повідомлення, вони повинні мати доступ до відкритого ключа і сертифіката одержувача. Якщо використовується інфраструктура Windows Public Key Infrastructure (PKI), інтегрована з Active Directory (AD), то Outlook 2003 може одержати сертифікат одержувача від сервера Global Catalog (GC). Якщо

одержувачі не визначені в AD (наприклад, зовнішні одержувачі), то користувачі можуть одержати копії їх сертифікатів, попросивши одержувачів вислати підписані повідомлення електронної пошти. Потім користувачі створюють об'єкт "контакт" Outlook для цього одержувача, і Outlook автоматично вводить сертифікати одержувача в об'єкт "контакт".

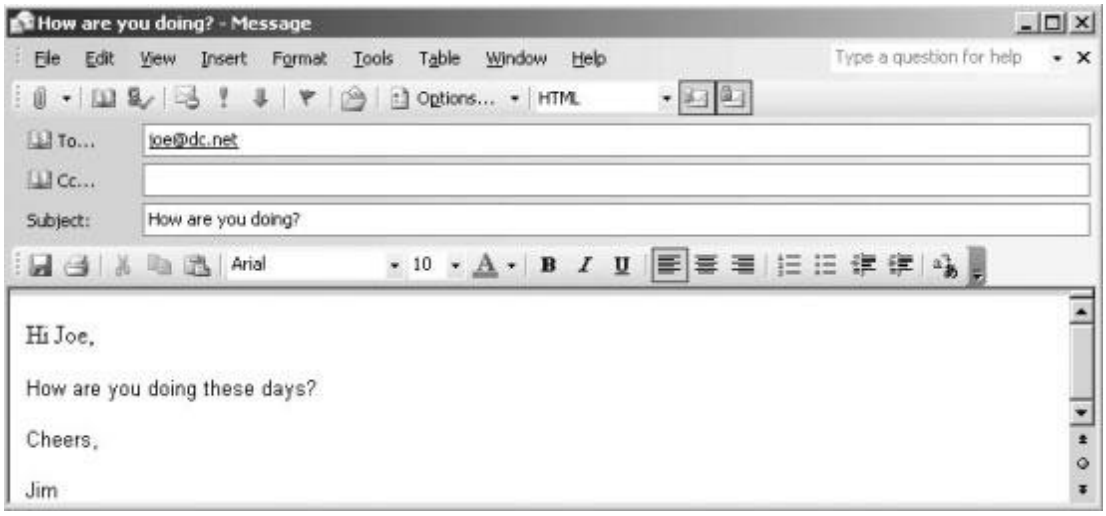


Рисунок 3.20 – Ілюстрація вікна Message

Особливості проектування і розгортання. Була розглянута проста зовнішня складова рішення безпеки пошти на базі S/MIME. При проектуванні і розгортанні ефективного механізму обробки сертифікатів необхідно враховувати і глибші аспекти. Перш ніж приступити до завдання, необхідно відповісти на таке питання: чи забезпечують вживані на підприємстві поштові клієнти достатні служби S/MIME? Якщо поштові клієнти не підтримують повний функціональний набір S/MIME, необхідний користувачам (табл. 3.10, в якій наведений огляд різних функціональних наборів S/MIME), то потрібно відновити клієнтські програми електронної пошти.

Якщо поточний поштовий клієнт зовсім несумісний з S/MIME, то користувачі не зможуть створювати S/MIME-захищені повідомлення або прочитати зашифровані або неявно підписані послання, хоча завжди зможуть прочитати явно підписані повідомлення. На серверах Exchange Server 2003 можна активізувати S/MIME для користувачів через OWA.

В основі S/MIME – цифрові сертифікати у форматі X.509, сертифікації, що генеруються центрами. Щоб надати користувачам сертифікати S/MIME, можна використовувати внутрішні центри сертифікації (ЦС) або купити сертифікати комерційного центру. Внутрішні центри сертифікації забезпечують підприємству повний контроль над "механізмами" сертифікації і дозволяють

надати важливі служби, такі, як архівація і відновлення ключів. Малим організаціям і організаціям, що потребують сертифікатів для обмеженого числа користувачів і застосувань, рекомендовано купити сертифікати S/MIME від комерційного центру. Це найбільш простий і економічний варіант.

При підготовці до розгортання S/MIME необхідно передбачити процедуру оформлення і відновлення сертифікатів S/MIME. У домені Windows 2003, який має в своєму розпорядженні клієнтів XP або новіших і вбудовану інфраструктуру Windows PKI, рекомендовано застосувати автоматичний метод оформлення і оновлення сертифікатів. У домені Windows 2000 Server з Windows PKI користувачі можуть вручну оформляти сертифікати за допомогою Web-інтерфейсу CA або оснащення Certificates консолі керування Microsoft Management Console (MMC). Якщо сертифікати S/MIME одержані від комерційного CA, то необхідно дотримуватися правил, установлених центром сертифікації.

Ведення центральної бази даних приватних ключів може бути зв'язано з ризиком застосування користувачем одного ключа, як для шифрування, так і підпису поштових повідомлень. Адміністратори-зловмисники можуть витягнути приватний ключ користувача з архівної бази даних і підписати повідомлення від імені користувача. Аби уберегти користувачів від цих небезпек, в більшості клієнтських і серверних програм S/MIME (зокрема, Exchange і Outlook) застосовуються пари подвійних ключів. У такій системі користувач завжди має дві пари ключів: одна для підпису (приватний ключ з пари архівується) і одна для шифрування (приватний ключ з цієї пари не архівується). Якщо в організації не встановлено вимоги архівації і відновлення, то можна використовувати S/MIME з однією парою ключів.

У решті випадків рекомендовано застосовувати дві пари ключів для кожного користувача S/MIME.

Чи мають потребу користувачі в посиленому захисті приватного ключа? Якщо користувачі застосовують S/MIME, щоб підписувати важливі поштові повідомлення, то необхідно додатково захистити приватний ключ. Компанія Microsoft запропонувала програмний механізм захисту, який запитує у користувачів пароль кожного разу, коли їх додатки звертаються до приватного ключа. На екрані Cryptographic Service Provider майстра Certificate Request Wizard (рис. 3.21) користувачі можуть надійно захистити приватний ключ, встановивши прапорець Enable strong private key protection при оформленні сертифіката S/MIME.

Таблиця 3.10 – Характеристики S/MIME поштових клієнтів Microsoft

Функція	Поштовий клієнт		
	Outlook 2003	Outlook Express 6.0	OWA
Поштовий протокол	MAPI/RPC, POP3, IMAP4, SMTP,HTTP	POP3, IMAP4, SMTP	HTTP
Постачальник сертифікатів	Exchange KMS, внутрішній ЦС, комерційний ЦС	Внутрішній ЦС, комерційний ЦС	Внутрішній ЦС, комерційний ЦС
Необхідний сервер обробки повідомлень Exchange	Ні	Ні	Так
Шифрування пошти (захист конфіденційності)	Так	Так	Так
Підпис пошти (перевірка справжності та цілісності)	Так	Так	Так
Безпечні Повідомлення	Так	Ні	Ні
Оформлення та оновлення сертифікатів користувачів	Може бути автоматизовано з використанням процедур оформлення сертифікатів Windows 2003 і XP	Може бути автоматизовано з використанням процедур оформлення сертифікатів Windows 2003 і XP	Може бути автоматизовано з використанням процедур оформлення сертифікатів Windows 2003 і XP
Перевірку відкликаних сертифікатів включено за умовчанням	Так	Ні	Так

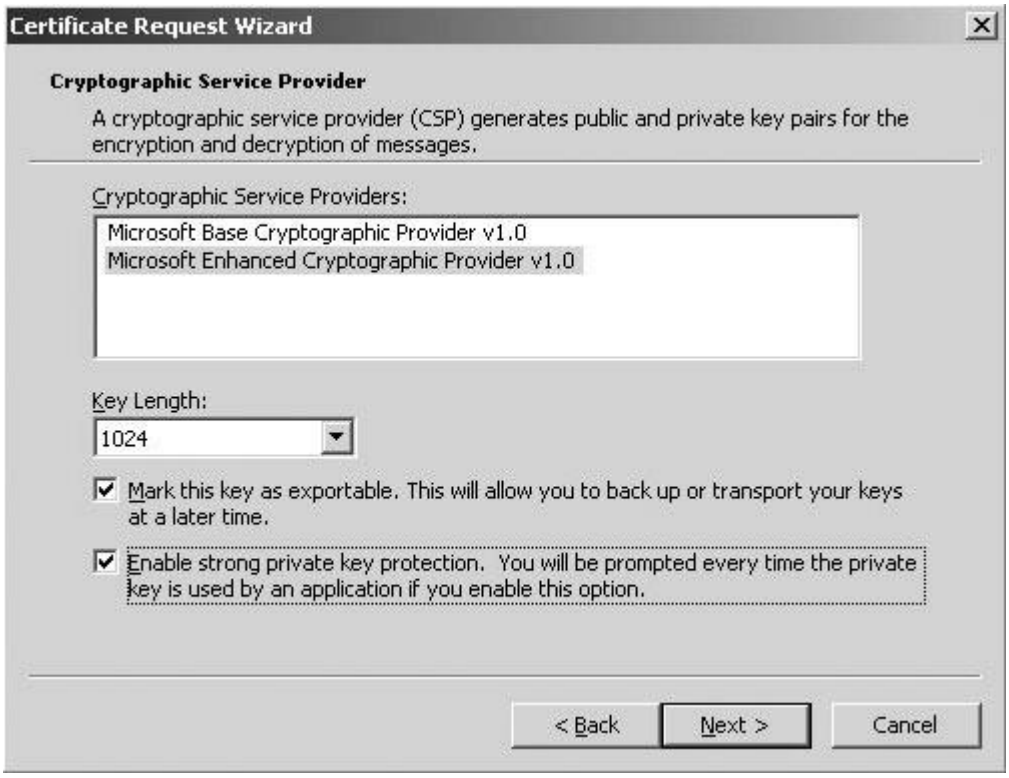


Рисунок. 3.21 – Ілюстрація екрана Cryptographic Service Provider майстра Certificate Request Wizard

На екрані з'явиться діалогове вікно Creating a new RSA exchange key, в якому потрібно встановити параметр High, щоб захистити приватний ключ паролем.

В інших рішеннях захисту приватного ключа застосовуються спеціалізовані апаратні засоби, такі, як пристрої USB, смарт-карти і пристрої читання смарт-карт або робочі станції з мікросхемою TPM (Trusted Platform Module).

У табл. 3.11 подані криптографічні алгоритми, які використані в системі S/MIME.

Стандарт цифрового підпису (алгоритм DSS) є основним алгоритмом створення цифрового підпису. Основним алгоритмом шифрування сеансових ключів у S/MIME є алгоритм Діфі-Хелмана, але фактично в S/MIME використовується варіант алгоритму Діфі-Хелмана, що забезпечує шифрування/розшифрування і відомий як алгоритм Ель-Гамаля. Як альтернатива як для підписів, так і для шифрування сеансових ключів може використовуватися алгоритм RSA. Це ті ж алгоритми, які застосовуються в PGP, оскільки вони забезпечують достатньо високий рівень захисту. Для функції хешування, використаної при створенні цифрових підписів, специфікації рекомендують 160-бітовий алгоритм SHA-1, але вимагають підтримки 128-бітового алгоритму MD5. Існують обґрунтовані сумніви в

достатній захищеності MD5, отже, SHA-1 є, очевидно, головною альтернативою. Проте MD5 застосовується дуже широко, чим і пояснюється його підтримка.

Для шифрування повідомлень рекомендовано "потрійний" DES з трьома ключами (tripleDES), але будь-яка гнучка реалізація повинна підтримувати 40-бітову версію алгоритму RC2. Останній є вельми слабким алгоритмом шифрування, проте відповідає експортним вимогам США.

Специфікації S/MIME включають опис процедури вибору алгоритму шифрування вмісту. По суті, агент відсилання повідомлень має вибір з двох варіантів. По-перше, він повинен визначити, чи здатний агент приймання повідомлень розшифрувати даний алгоритм шифрування. По-друге, якщо агент приймання здатний приймати тільки слабо шифрований вміст, агент відсилання повинен вирішити, чи є прийнятним використання слабкого шифрування. Для підтримки цього процесу вибору агент відсилання може оголошувати можливості розшифрування у порядку зростання переваги в будь-якому посланому повідомленні. Агент прийому може зберегти цю інформацію для подальшого застосування.

Агент відсилання повинен використовувати для вирішення подані нижче за правило в наступному порядку.

Якщо агент відсилання має список переваги можливостей розшифрування передбачуваного одержувача, рекомендовано вибрати з цього списку першу можливість (можливість з найвищою перевагою) з тих, які одержувач може використовувати.

Якщо агент відсилання не має такого списку можливостей передбачуваного одержувача, але має одне або ряд повідомлень, що прийшли від одержувача, то для повідомлення, яке відправляється, рекомендовано використовувати алгоритм шифрування, що застосовувався в останньому з одержаних підписаних і шифрованих повідомлень передбачуваного адресата.

Якщо агент відсилання нічого не знає про можливості розшифрування передбачуваного одержувача і не має наміру ризикувати тим, що одержувач не зможе розшифрувати повідомлення, то агенту відсилання рекомендовано використовувати tripleDES.

ОБОВ'ЯЗКОВО. *Визначення є абсолютною вимогою специфікації. Будь-яка реалізація повинна включати цю властивість або функцію, щоб відповідати даній специфікації.*

РЕКОМЕНДОВАНО. *У конкретному оточенні можуть існувати причини ігнорувати цю властивість або функцію, але РЕКОМЕНДОВАНО, щоб реалізація все ж таки мала відповідну властивість або функцію.*

Таблиця 3.11 – Криптографічні алгоритми, які використовуються в системі S/MIME

Функція	Вимога
Створення профілю повідомлення для формування цифрового підпису	ОБОВ'ЯЗКОВА підтримка SHA-1 і MD5 РЕКОМЕНДОВАНО використання SHA-1
Шифрування профілю повідомлення для формування цифрового підпису	Для агентів відсилання і приймання ОБОВ'ЯЗКОВА підтримка DSS. Для агента відсилання РЕКОМЕНДОВАНО підтримку шифрування RSA. Для агента приймання РЕКОМЕНДОВАНО підтримку верифікації підписів RSA з довжиною ключа від 512 до 1024 біт
Шифрування сеансового ключа для передачі з повідомленням	Для агентів відсилання і прийому ОБОВ'ЯЗКОВА підтримка алгоритму Діфі-Хелмана. Для агента відсилання РЕКОМЕНДОВАНО підтримку шифрування RSA з довжиною ключа від 512 до 1024 біт. Для агента прийому РЕКОМЕНДОВАНО підтримку розшифрування RSA
Шифрування повідомлення для передачі з використанням сеансового ключа	Для агента відсилання РЕКОМЕНДОВАНО підтримку шифрування tripleDES і RC2/40. Для агента прийому ОБОВ'ЯЗКОВА підтримка розшифрування tripleDES і РЕКОМЕНДОВАНО підтримку розшифрування RC2/40

Якщо агент відправлення нічого не знає про можливості розшифрування передбачуваного одержувача і не хоче ризикувати тим, що одержувач не зможе розшифрувати повідомлення, то агенту відсилання слід обов'язково використовувати RC2/40.

Якщо повідомлення повинно бути послано декільком одержувачам і не можна вибрати універсальний алгоритм шифрування, то агенту відправлення слід відправити два повідомлення. Проте в такому разі важливо відзначити, що повідомлення стає більш уразливим при передачі копії повідомлення із слабкішим захистом.

У S/MIME визначено ряд нових типів вмісту MIME, перелічених у табл. 3.12. Всі ці нові типи використовують позначення PKCS (Public-Key Cryptography Specifications – специфікації криптографії з відкритим ключем), опубліковані RSA Laboratories і доступні для S/MIME.

Таблиця 3.12 – Типи вмісту S/MIME

Тип	Підтип	Параметр smime	Опис
Multipart (багато-компонентний)	Signed (підписаний)	–	Відкрите підписане повідомлення з двох частин: повідомлення і його підписи
Application (додаток)	pkcs7-mime	signedData	Підписаний об'єкт S/MIME
	pkcs7-mime	envelopedData	Шифрований об'єкт S/MIME
	pkcs7-mime	degenerate signedData	Об'єкт, що містить тільки сертифікати відкритих ключів
	pkcs7-signature	–	Тип підпису, що є частиною повідомлення типу multipart/signed
	pkcs10-mime	–	Повідомлення запиту реєстрації сертифіката

Розглянемо кожний з цих типів.

• **Упаковані дані.** Підтип **application/pkcs7-mime** призначений для одного з чотирьох видів обробки S/MIME, кожного зі своїм унікальним параметром smime-типу. У всіх випадках об'єкт, що виходить у результаті, подається в так званому форматі BER (Basic Encoding Rules – основні правила кодування), визначеному в рекомендаціях X.209 групи ITU-T. Формат BER є набором рядків довільних байтів, які, таким чином, є даними в двійковому форматі. Такий об'єкт у зовнішньому повідомленні MIME повинен передаватися кодованим у форматі base64. Спочатку розглянемо об'єкт **enveloped Data** (упаковані дані).

При підготовці об'єкта **enveloped Data** MIME мають бути виконані такі дії.

Генерується псевдовипадковий сеансовий ключ для конкретного алгоритму симетричної схеми шифрування (RC2/40 або 3DES).

Для кожного одержувача сеансовий ключ шифрується за допомогою відкритого ключа одержувача і RSA.

Для кожного одержувача готується блок даних, званий **Recipient lnfo**

(інформація для одержувача), що містить сертифікат відкритого ключа відправника, ідентифікатор алгоритму, що був використаний для шифрування сеансового ключа, і шифрований сеансовий ключ.

Вміст повідомлення шифрується за допомогою сеансового ключа.

Блоки **Recipient lnfo**, за якими йде шифрований вміст повідомлення, разом складають блок **enveloped Data**. Ця інформація потім кодується у форматі **base64**. Ось приклад такого повідомлення (за винятком заголовків RFC 822).

Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m

Content-Transfer-Encoding: base64 Content-Disposition: attachment; filename=smime.p7m

rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF4 67GhIGfHfYT6 7n8HHGghyHhHUujhJh4VQpfyF4 67GhIGfHfYGTrfvbnjT6jH7756tbB9H f8HHGTrfvhJhjH77 6tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4 0GhIGfHfQbnj7 56YT64V

Щоб відновити шифроване повідомлення, одержувач спочатку знімає кодування **base64**. Потім використовується особистий ключ одержувача, щоб відкрити сеансовий ключ. Нарешті, вміст повідомлення розшифровується за допомогою цього сеансового ключа.

- **Підписані дані.** Підтип **signed Data** (підписані дані) призначений для документів, підписаних однією або декількома сторонами. Для ясності обмежимо дослідження випадком одного цифрового підпису. Для підготовки об'єкта **signed Data** MIME слід виконати такі дії.

Вибрати алгоритм створення профілю повідомлення (SHA або MD5).

Обчислити профіль повідомлення (значення хеш-функції) для вмісту, який повинен бути підписаний.

Профіль повідомлення зашифрувати за допомогою особистого ключа сторони, що підписує документ.

Підготувати блок, званий **Signer lnfo** (інформація сторони, що підписала), що містить сертифікат відкритого ключа сторони, яка підписала документ, ідентифікатор алгоритму, що використався для шифрування профілю повідомлення і шифрованого профілю повідомлення.

Об'єкт **signed Data** формується з ряду блоків, що включають ідентифікатор алгоритму створення профілю повідомлення, власне підписане повідомлення і блок **Signer lnfo**. Об'єкт **signed Data** може також включати набір сертифікатів відкритих ключів, достатній для того, щоб скласти ланцюжок від визнаного центру сертифікації вищого рівня довіри до сторони, що підписала документ. Ця інформація потім кодується у форматі **base64**. Ось приклад такого

повідомлення (з за винятком заголовків RFC 822).

Content-Type: application/pkcs7-mime; smime-type=signed-data; name=smime.p7m

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=smime.p7m

567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH77 6tbB9HG4VQbnj7 77n8HHGT9HG4VQpfyF4 67GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH HUujhJh4VQpfyF4 67GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh 6YT64VOGhIGfHfQbnj 75

Щоб відновити підписане повідомлення і перевірити підпис, одержувач спочатку знімає кодування **base64**. Потім використовується відкритий ключ сторони, що підписала документ, аби відкрити профіль повідомлення. Нарешті, одержувач самостійно обчислює профіль повідомлення і порівнює його з розшифрованим профілем повідомлення, щоб перевірити підпис.

- **Відкрите підписане повідомлення.** Відкрите підписане повідомлення виходить тоді, коли для вмісту використовується тип **multipart** і підтип **signed**. Як уже згадувалося, такий процес підпису не трансформує саме підписане повідомлення, так що воно пересилається у "відкритому" вигляді. Таким чином, одержувачі з можливостями MIME, але не S/MIME, все одно зможуть прочитати повідомлення, що надійшло.

Повідомлення типу **multipart/signed** включає дві частини. Перша частина може бути будь-якого типу MIME, але підготовлена так, щоб її не можна було змінити на шляху проходження від джерела до адресата. Тобто, якщо перша частина не подана в 7-бітовому кодуванні (7bit), то дані необхідно кодувати, використовуючи формат **base64** або **quoted-printable**. Потім ця частина повідомлення обробляється точно так, як в об'єкті **signed Data**, але в даній ситуації в результаті створюється об'єкт у форматі **signed Data**, поле вмісту якого виявляється порожнім. Цей об'єкт є відокремленим підписом. Потім він кодується у формат **base64**, аби стати другою частиною багатокомпонентного повідомлення. Для типу MIME цієї другої частини вибирається значення **application**, а для підтипу - **pkcs7-signature**. Ось приклад такого повідомлення.

Content-Type: multipart/signed; protocol="application/pkcs7-signature"; micalg=shal; boundary=boundary42

--boundary42 Content-Type: text/plain

Це відкритий текст підписаного повідомлення.

--boundary42

Content-Type: application/pkcs7-signature; name=smime.p7s

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6

4VQpfyF4 67GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj n8HHGTrfvhJhjH77 6tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4 7GhIGfHfYT64VQbnj 756

--boundary42--

•**Запит реєстрації.** Як правило, додаток або користувач звертається до центру сертифікації з приводу отримання сертифікатів відкритих ключів. Об'єкт S/MIME типу **application/pkcs -10** служить для того, щоб передати запит такого сертифіката. Запит сертифіката включає блок **certification Request lnfo** (інформація про запит сертифіката), за яким йде ідентифікатор алгоритму шифрування з відкритим ключем, що завершується підписом блока **certification Request lnfo**, виконаним за допомогою особистого ключа відправника. Блок **certification Request lnfo** включає ім'я об'єкта сертифікації (об'єкт, чий відкритий ключ повинен бути сертифікований) і надання відкритого ключа користувача у вигляді рядка бітів.

Повідомлення, що містять тільки сертифікати або список відкликаних сертифікатів (CRL), можуть посилатися у відповідь на запит реєстрації. Типом/підтипом такого повідомлення буде **application/pkcs7-mime** з параметром **degenerate** (вироджений) для smime-типу. Виконані при цьому дії аналогічні тим, що і при створенні повідомлення **signed Data**, за винятком того, що в даному випадку немає вмісту повідомлення і поле **signed lnfo** виявляється порожнім.

S/MIME використовує сертифікати відкритих ключів, відповідні версії 3 стандарти X.509. Схема керування ключами в S/MIME є, в деякому розумінні, гібридом суворої ієрархії сертифікатів X.509 і мережею довіри PGP. Як і в моделі PGP, адміністратори та/або користувачі S/MIME повинні забезпечити кожному клієнту список надійних ключів і список відкликаних сертифікатів. Це значить, що відповідальність за підтримку множини сертифікатів, необхідних для перевірки підписів, які надходять, і шифрування повідомлення, що відправляється, лягає на локальну систему. В той же час самі сертифікати підписуються уповноваженими центрами сертифікації.

Користувач S/MIME повинен виконувати такі функції керування ключами.

Генерування ключів. Для користувача відповідної утиліти адміністрування (наприклад, здійснюючою керування локальною мережею) ОБОВ'ЯЗКОВО повинна бути передбачена можливість генерувати окремі пари ключів Діфі-Хелмана і DSS і РЕКОМЕНДОВАНО мати можливість генерувати пари ключів RSA. Кожна пара ключів ОБОВ'ЯЗКОВО повинна генеруватися з використанням випадкових значень, одержаних від якісного недетермінованого

джерела таких значень, і повинна бути деяким чином захищена. Для агента користувача РЕКОМЕНДОВАНО генерувати пари ключів RSA з довжиною ключа від 768 до 1024 біт і у жодному випадку не генерувати ключі завдовжки менше 512 біт.

Реєстрація. Відкритий ключ користувача повинен бути зареєстрований за допомогою уповноваженого центру сертифікації для того, щоб одержати сертифікат цього ключа стандарту X.509.

Зберігання і пошук сертифікатів. Користувачу потрібен доступ до локального списку сертифікатів, щоб мати можливість перевірити підписи, які надходять, і шифрувати повідомлення, що відправляються. Такий список може підтримуватися користувачем або деяким локальним адміністративним об'єктом від імені групи користувачів.

3.4. Захист інформації в електронних платіжних системах

Сучасну практику банківських операцій, операцій у торгівлі взаємних платежів неможливо уявити без розрахунків із застосуванням пластикових карток. Завдяки надійності, універсальності і зручності пластикові картки завоювали міцне місце серед інших платіжних засобів.

Електронною платіжною системою називають сукупність методів і суб'єктів, які їх реалізують, що забезпечує в рамках системи використання банківських пластикових карток як платіжних засобів.

3.4.1. Електронні пластикові картки

Застосування банкоматів можливо при використанні деякого носія інформації, який міг би ідентифікувати користувача і зберігати певні облікові дані. Таким носієм інформації є пластикові картки.

Пластикова картка є пластиною стандартних розмірів (85,6x53,9x0,76 мм), виготовленою із спеціальної, стійкої до механічних і термічних дій пластмаси. Одна з основних функцій пластикової картки – забезпечити ідентифікацію особи, яка її використовує, як суб'єкта платіжної системи. Для цього на пластикову картку наносять логотипи банку-емітента і платіжної системи, яка обслуговує цю картку, ім'я власника картки, номер його рахунку, термін дії картки і т.п. Крім того, на карті може бути присутньою фотографія власника і його підпис. Алфавітно-цифрові дані – ім'я, номер рахунку та ін. можуть бути ембосовані, тобто нанесені рельєфним шрифтом. Це дає можливість при ручній обробці карток, що приймаються до оплати, швидко перенести дані на чек за допомогою спеціального пристрою-імпринтера, що здійснює "прокатування" картки (аналогічно отриманню другого екземпляра при використанні копіювального паперу).

За принципом дії розрізняють пасивні і активні пластикові картки. Пасивні пластикові картки всього лише зберігають інформацію на тому або іншому носії. До них належать пластикові картки з магнітною смугою.

Картки з магнітною смугою є на сьогодні найбільш поширеними – в обігу знаходиться понад два мільярди карток подібного типу. Магнітна смуга розташовується на зворотному боці картки і відповідно до стандарту ISO 7811 складається з трьох доріжок. З них перші дві призначені для зберігання ідентифікаційних даних, а на третю доріжку можна записувати інформацію (наприклад, поточне значення ліміту дебетової картки). Проте через невисоку надійність багато разів повторюваного процесу запису і читання запис на магнітну смугу зазвичай не практикується. Такі картки використовуються лише в режимі читання інформації.

Картки з магнітною смугою відносно уразливі для шахрайства. Проте розвинена інфраструктура існуючих платіжних систем і, зокрема, світових лідерів у галузі "карткового" бізнесу компаній є причиною інтенсивного використання карток з магнітною смугою і сьогодні.

Для підвищення захищеності своїх карток системи використовують додаткові графічні засоби захисту: голограми і нестандартні шрифти для ембосування.

Платіжні системи з подібними картками вимагають on-line авторизації в торгових точках і, як наслідок, наявність розгалужених, високоякісних засобів комунікації (телефонних ліній). Тому з технічної точки зору подібні системи мають серйозні обмеження до їх застосування в країнах з погано розвиненими системами зв'язку.

Помітна особливість активних пластикових карток – наявність вбудованої в неї електронної мікросхеми. Принцип пластикової картки з електронною мікросхемою запатентував у 1974 р. француз Ролан Моренно. Стандарт IS0 7816 визначає основні вимоги до карток на інтегральних мікросхемах або чипових картках. У недалекому майбутньому картки з мікросхемою витіснять картки з магнітною смугою. Тому зупинимося детальніше на основних типах карток з мікросхемою.

Картки з мікросхемою можна класифікувати за декількома ознаками.

Перша ознака – функціональні можливості картки. Тут можна виділити такі основні типи карток:

- картки-лічильники;
- картки з пам'яттю;
- картки з мікропроцесором.

Друга ознака – тип обміну з пристроєм, якій зчитує інформацію:

- картки з контактним зчитуванням;
- картки з індукційним зчитуванням.

Картки-лічильники застосовуються, як правило, в тих випадках, коли інша платіжна операція вимагає зменшення залишку на рахунку утримувача

картки на деяку фіксовану суму. Подібні картки використовуються в спеціалізованих застосуваннях з передоплатою (платня за використання телефону-автомата, оплата автостоянки і т.ін.). Очевидно, що застосування карток з лічильником обмежене і не має великої перспективи.

Картки з пам'яттю є перехідними між картками з лічильником і картками з процесором. Картка з пам'яттю – це, по суті, картка з лічильником, що перезаписується, до якої вжито заходів, що підвищують її захищеність від атак зловмисників. У простих з існуючих карток з пам'яттю обсяг пам'яті може складати від 32 байт до 16 кілобайт. Ця пам'ять може бути реалізована у вигляді програмованого постійного ППЗУ (EPROM), що запам'ятовує пристрої, яке допускає одноразовий запис і багатократне зчитування, або у вигляді пристрою ЕСППЗУ (EEPROM), що допускає багатократний запис і багатократне зчитування.

Картки з пам'яттю можна розділити на два типи: з незахищеною (повнодоступною) і захищеною пам'яттю.

У картках першого типу немає жодних обмежень для читання і запису даних. Їх не можна використовувати як платіжні, оскільки фахівець середньої кваліфікації може їх достатньо просто "зламати".

Картки другого типу мають зону ідентифікаційних даних і одну або декілька прикладних ділянок. Ідентифікаційна зона карток допускає лише одноразовий запис при персоналізації і надалі доступна лише для зчитування. Доступ до прикладних ділянок регламентується і здійснюється тільки при виконанні певних операцій, зокрема, при введенні секретного PIN-коду.

Рівень захисту карток з пам'яттю вищий, ніж у магнітних, вони можуть бути використані в прикладних системах, в яких фінансові ризики, пов'язані з шахрайством, відносно невеликі. Як платіжний засіб картки з пам'яттю використовуються для оплати таксофонів загального користування, проїзду в транспорті, в локальних платіжних системах (клубні картки). Картки з пам'яттю застосовуються також у системах допуску в приміщення і доступу до ресурсів комп'ютерних мереж (ідентифікаційні картки). Картки з пам'яттю мають нижчу вартість у порівнянні з картками з мікропроцесором.

Картки з мікропроцесором називають також інтелектуальними картками або смарт-картками (smart cards). Картки з мікропроцесором є, по суті, мікрокомп'ютерами і містять всі відповідні основні апаратні компоненти: центральний процесор (ЦП), оперативний запам'ятовувальний пристрій (ОЗП), постійний запам'ятовувальний пристрій (ПЗП), а також електричний стираючий програмований ПЗП (ЕСППЗП) (рис. 3.22).

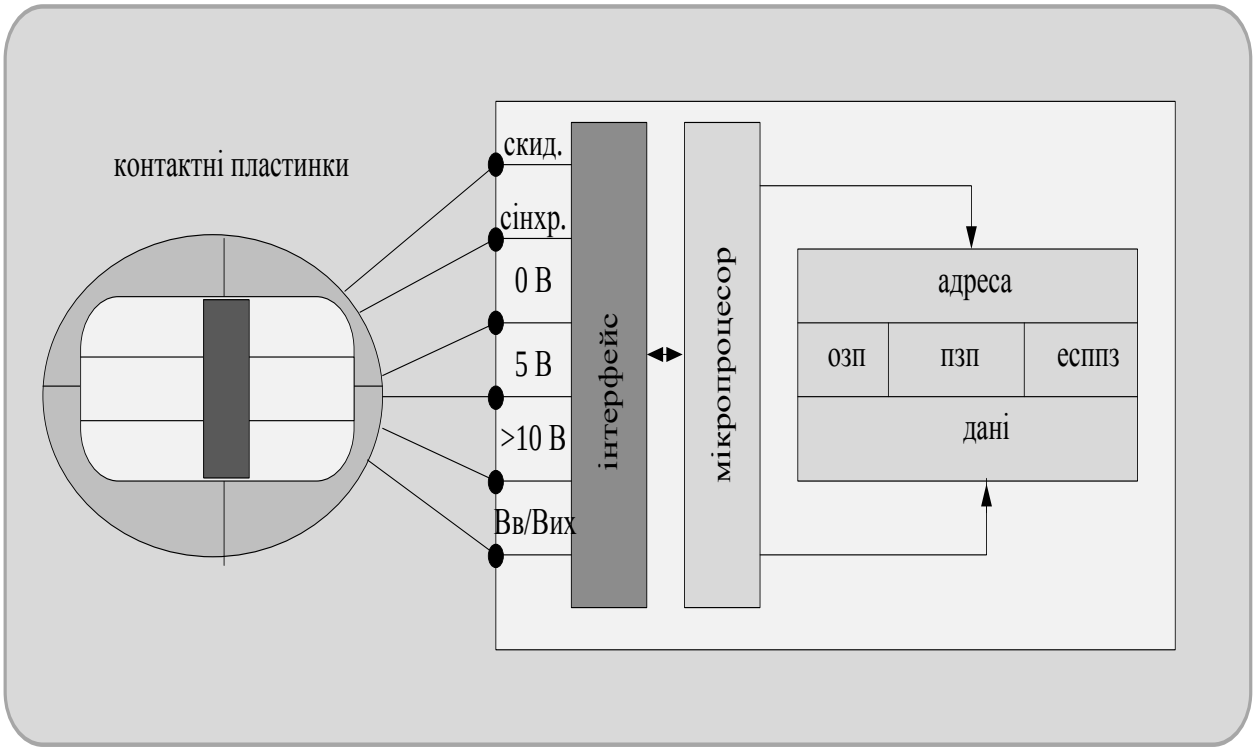


Рисунок 3.22 – Ілюстрація смарт-картки

У теперішній час у смарт-картки встановлюють:

- мікропроцесори з текстовою частотою 5 Мгц;
- оперативне ЗУ місткістю до 256 байт;
- постійне ЗУ місткістю до 10 Кбайт;
- незалежне ЗУ місткістю до 8 Кбайт.

У ПЗП записаний спеціальний набір програм, який називається операційною системою картки COS (Card Operation System). Операційна система підтримує файлову систему, яка забезпечує регламентацію доступу до даних, що базується в ЕСППЗУ (місткість якого зазвичайо знаходиться в діапазоні 1...8 Кбайт, але може досягати і 64 Кбайт). При цьому частина даних може бути доступна тільки внутрішнім програмам картки.

Смарт-картка забезпечує широкий набір функцій:

- розмежування повноважень доступу до внутрішніх ресурсів (завдяки роботі із захищеною файловою системою);
- шифрування даних із застосуванням різних алгоритмів;
- формування електронного цифрового підпису;
- ведення ключової системи;
- виконання всіх операцій взаємодії власника картки, банку і торговця.

Деякі картки забезпечують режим "самоблокування" (неможливість подальшої роботи з нею) при спробі несанкціонованого доступу. Смарт-картки дозволяють істотно спростити процедуру ідентифікації клієнта. Для перевірки

PIN-коду застосовується алгоритм, що реалізується мікропроцесором на картці. Це дозволяє відмовитися від роботи банкомату в режимі реального часу і централізованої перевірки PIN. Значні особливості роблять смарт-картку високозахищеним платіжним інструментом, який може бути використаний у фінансових операціях, що ставлять підвищені вимоги до захисту інформації. Саме тому мікропроцесорні смарт-картки розглядаються в даний час як найбільш перспективний вид пластикових карток.

За принципом взаємодії з пристроєм, який зчитує інформацію, розрізняють картки двох типів:

- картки з контактним зчитуванням;
- картки з безконтактним зчитуванням.

Картка з контактним зчитуванням має на своїй поверхні 8...10 контактних пластин. Розміщення контактних пластин, їх кількість і призначення виводів різні у виробників і природно, що зчитувачі для карток даного типу розрізняються також.

Останніми роками почали широко застосовуватися картки з безконтактним зчитуванням. У них обмін даними між картою і зчитувальним пристроєм проводиться індукційним способом. Очевидно, що такі картки надійніші і довговічніші.

Персоналізація і авторизація карток є важливими етапами підготовки і застосування пластикових карток.

Персоналізація картки здійснюється при видачі картки клієнту. При цьому на картку заносяться дані, що дозволяють ідентифікувати картку і її власника, а також здійснити перевірку платоспроможності картки при прийманні її до оплати або видачі готівки.

Під авторизацією розуміють процес затвердження продажу або видачі готівки за карткою. Для проведення авторизації точка обслуговування робить запит платіжній системі про підтвердження повноважень пред'явника картки і його фінансових можливостей. Технологія авторизації залежить від типу картки, схеми платіжної системи і технічної оснащеності точки обслуговування.

Історично склалося так, що першим способом персоналізації карток було ембосування.

Ембосування − це процес рельєфного тиснення даних на пластиковій основі картки. На картках банків-емітентів ембосуються, як правило, такі дані: номер картки; дати початку і закінчення терміну її дії; прізвище та ім'я власника.

Деякі платіжні системи, наприклад Visa, вимагають тиснення двох спеціальних символів, що однозначно ідентифікують належність банку-емітента до платіжної системи. Ембосери (пристрої для тиснення рельєфу на картці) випускають обмежене коло виготовлювачів . У ряді країн Заходу

законодавчо заборонено вільний продаж ембосерів. Спеціальні символи, що підтверджують належність картки до тієї або іншої платіжної системи, поставляються власнику ембосера тільки з дозволу керівного органу платіжної системи. Ембосована картка може служити засобом платежу при використанні імпринтера-пристрою для плющення сліпа (чека), який підтверджує досконалу платіжну операцію.

До персоналізації карток належать також кодування магнітної смуги або програмування мікросхеми.

Кодування магнітної смуги проводиться, як правило, на тому ж устаткуванні, що й ембосування. При цьому частина інформації про картку, що містить номер картки в період її дії, однакова як на магнітній смузі, так і на рельєфі. Проте бувають ситуації, коли після первинного кодування потрібно додатково занести інформацію на магнітну доріжку. В цьому випадку застосовуються спеціальні пристрої з функцією "читання-запис". Це можливо, зокрема, коли PIN-код для користування карткою не формується спеціальною програмою, а може бути вибраний клієнтом на свій розсуд.

Програмування мікросхеми не вимагає особливих технологічних прийомів, проте має деякі організаційні особливості. Зокрема, для підвищення безпеки й виключення можливих зловживань операції з програмування різних частин мікросхеми рознесли територіально і розмежували права різних співробітників, що беруть участь у цьому процесі.

Як правило, ця процедура розбивається на три етапи:

- на першому робочому місці виконується активація картки (введення її в дію);
- на другому робочому місці виконуються операції, пов'язані із забезпеченням безпеки;
- на третьому робочому місці проводиться власне персоналізація картки.

Традиційно процес авторизації проводиться "вручну", коли продавець або касир передає запит по телефону оператору (голосова авторизація) або автоматично, коли картка поміщається до POS-терміналу, дані прочитуються, касир вводить суму платежу, а власник картки із спеціальної клавіатури секретний PIN-код. Після цього термінал здійснює авторизацію, встановлюючи зв'язок з базою даних платіжної системи (on-line режим) або реалізовуючи додатковий обмін даними з самою карткою (off-line авторизація). У разі видачі готівки процес має аналогічний характер, з тією лише особливістю, що гроші в автоматичному режимі видаються спеціальним пристроєм-банкоматом, який і проводить авторизацію.

Для захисту карток від підробки і подальшого несанкціонованого застосування використовуються різні методи і способи. Наприклад, для персоналізації карток може застосовуватися нанесення на пластикову основу чорно-білої або кольорової фотографії власника картки методом термодруку.

На будь-якій картці завжди існує спеціальна смужка із зразком підпису власника картки. Для захисту картки як такої різні платіжні співтовариства застосовують спеціальні об'ємні зображення на лицьовій і зворотній стороні картки (голограми).

Перевіреним способом ідентифікації власника банківської картки є використання секретного персонального ідентифікаційного номера PIN. Значення PIN має бути відомим тільки власнику картки. Довжина PIN повинна бути достатньо великою, аби ймовірність вгадування правильного значення шляхом повного перебору значень була незначною. З іншого боку, довжина PIN повинна бути достатньо короткою, щоб дати можливість власникам карток запам'ятати його значення. Рекомендована довжина PIN становить 4...8 десяткових цифр, але може досягати 12.

Припустимо, що PIN має довжину чотири цифри, тоді зловмисник, що намагається підібрати значення PIN до банківської картки, має вибрати одну з десяти тисяч можливостей. Якщо число спроб введення некоректного значення PIN обмежується п'ятьма спробами на картку в день, цей зловмисник має шанси на успіх менш ніж 1:2000. Але наступного дня супротивник може спробувати знову. Тоді його шанси збільшуються до 1:1000. Кожного наступного дня збільшується імовірність успіху зловмисника. Тому багато банків вводять абсолютну межу на число неправильних спроб введення PIN на картку, щоб виключити атаку такого роду. Якщо встановлену межу перевищено, вважається, що дана картка неправильна і її відбирають.

Значення PIN однозначно пов'язане з відповідними атрибутами банківської картки, тому PIN можна трактувати як підпис власника картки. Щоб ініціювати транзакцію, власник картки, який використовує POS-термінал, вставляє свою картку до спеціальної щілини зчитувача і вводить свій PIN, використовуючи спеціальну клавіатуру термінала. Якщо введене значення PIN і номер рахунка клієнта, записаний на магнітну смугу картки, узгоджуються між собою, тоді ініціюється транзакція.

Захист персонального ідентифікаційного номера PIN для банківської картки є критичним для безпеки всієї платіжної системи. Банківські картки можуть бути втрачені, вкрадені або підроблені. У таких випадках єдиним контрзаходом проти несанкціонованого доступу залишається секретне значення PIN. От чому відкрита форма PIN повинна бути відома тільки законному власнику картки. Вона ніколи не зберігається і не передається в рамках системи електронних платежів. Очевидно, значення PIN потрібно тримати у секреті протягом усього терміну дії картки.

Метод генерації значення PIN істотно впливає на безпеку електронної платіжної системи. Взагалі, персональні ідентифікаційні номери можуть формуватися банком або власниками карт. Зокрема, клієнт розрізняє два типи PIN:

- PIN, призначений йому банком, що видав картку;
- PIN, вибраний власником картки самостійно.

Якщо PIN призначається банком, банк зазвичай використовує один з двох варіантів процедур генерації PIN.

При першому варіанті PIN генерується криптографічний з номера рахунка власника картки. Процес генерації PINa, призначеного з номера рахунку, показаний на рис. 3.23.

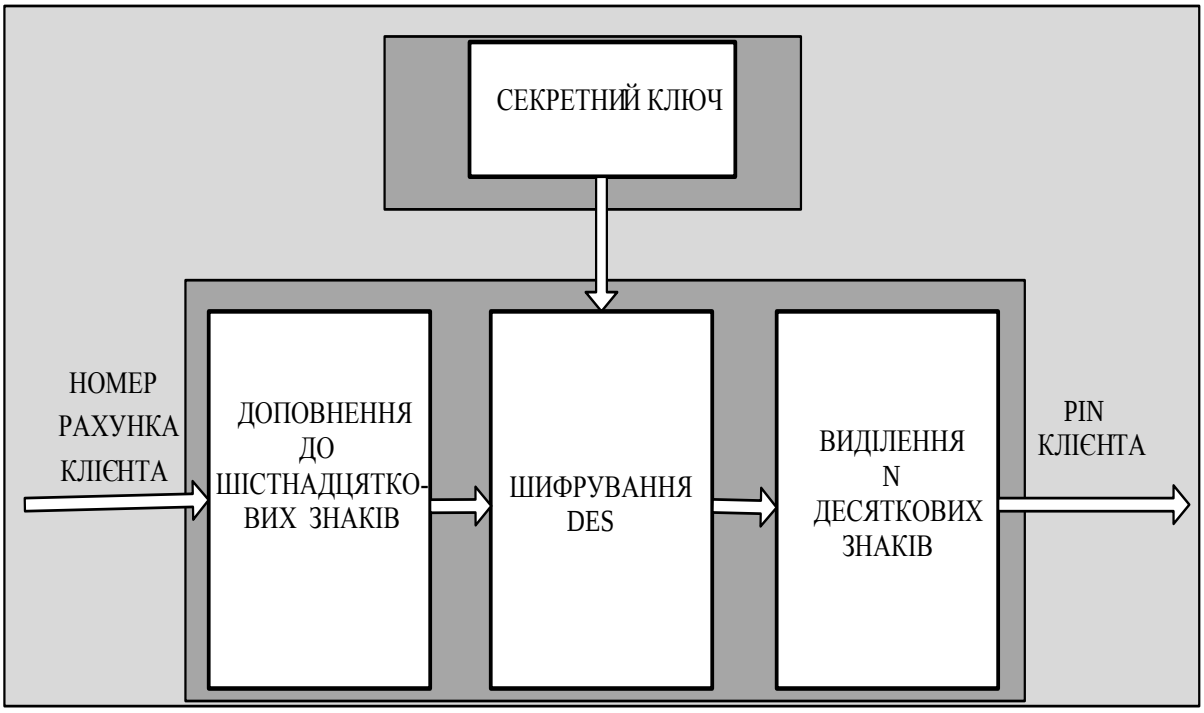


Рисунок 3.23 – Процес генерації PINa, призначеного з номера рахунка

Спочатку номер рахунка клієнта доповнюється нулями або іншою константою до 16 шістнадцяткових цифр (8 байт). Потім ці 8 байт шифруються за алгоритмом DES з використанням секретного ключа. З одержаного шифртексту завдовжки 8 байт по черзі виділяють 4-бітові блоки, починаючи з молодшого байта. Якщо число, що утворюється цими бітами, менше 10, то одержана цифра включається в PIN, інакше це значення не використовується. Таким шляхом обробляють усі 64 біт (8 байт). Якщо в результаті обробки не вдалося одержати відразу необхідну кількість десяткових цифр, то звертаються до невикористаних 4-бітових блоків, з яких віднімають 10.

Очевидна цінність цієї процедури полягає в тому, що значення PIN не потрібно зберігати усередині електронної платіжної системи. Недоліком підходу є те, що при необхідності зміни PIN потрібен вибір нового рахунка клієнта або нового криптографічного ключа. Банки вважають за краще, аби номер рахунка клієнта залишався фіксованим. З іншого боку, оскільки всі PIN обчислюють, використовуючи однаковий криптографічний ключ, зміна одного PIN при збереженні рахунка клієнта неминуче спричиняє зміну всіх

персональних ідентифікаційних номерів.

При другому варіанті банк вибирає значення PIN випадково, зберігаючи значення цього PIN у вигляді відповідної криптограми. Вибрані значення PIN банк передає власникам банківських карт, користуючись захищеним каналом.

Використання PIN, призначеного банком, незручне для клієнта навіть при невеликій його довжині. Такий PIN важко утримати в пам'яті, тому утримувач картки може записати його куди небудь. Головне – це не записати PIN безпосередньо на картку або будь-яке інше видне місце. В такому випадку завдання зловмисника буде сильно полегшене.

Для більшої зручності клієнта використовують значення PIN, вибране самим клієнтом. Такий спосіб визначення значення PIN дозволяє клієнту:

– використовувати один і той же PIN для різних цілей;

– задавати PIN як сукупність букв і цифр (для зручності запам'ятовування).

Коли PIN вибраний клієнтом, він повинен бути доведений до відома банку. PIN може бути переданий до банку замовленою поштою або відправлений через захищений термінал, розміщений в банківському офісі, який негайно його шифрує. Якщо банку необхідно використовувати вибраний клієнтом PIN, кожну цифру вибраного клієнтом PIN складають за модулем 10 (без урахування перенесень) з відповідною цифрою PIN, клієнта, що виводиться банком з рахунка. Отримане десяткове число називається "зміщення". Це зміщення запам'ятовується на картці клієнта. Оскільки PIN, що виводиться, має випадковий характер, то вибраний клієнтом PIN неможливо визначити за його "зміщенням".

Головна вимога безпеки полягає в тому, що значення PIN повинне запам'ятовуватися власником картки і ніколи не повинно зберігатися в будь-якій читабельній формі. Але люди дуже часто забувають значення PIN. Тому банки повинні наперед заготовити спеціальні процедури для таких випадків. Банк може реалізувати один з наступних підходів. Перший основано на відновленні забутого клієнтом значення PIN і відправці його назад власнику картки. При другому підході просто генерується нове значення PIN.

При ідентифікації клієнта за значенням PIN і пред'явленій картці використовуються два основні способи перевірки PIN: неалгоритмічний і алгоритмічний.

Неалгоритмічний спосіб перевірки PIN не вимагає застосування спеціальних алгоритмів. Перевірка PIN здійснюється шляхом безпосереднього порівняння введеного клієнтом PIN зі значеннями, що зберігаються в базі даних. Зазвичай база даних із значеннями PIN клієнтів шифрується методом прозорого шифрування, щоб підвищити її захищеність, не ускладнюючи процесу порівняння.

Алгоритмічний спосіб перевірки PIN полягає в тому, що введений

клієнтом PIN перетворять за певним алгоритмом з використанням секретного ключа і потім порівняють із значенням PIN, що зберігається в певній формі на картці. Переваги цього методу перевірки:

- відсутність копії PIN на головному комп'ютері виключає його розкриття персоналом банку;
- відсутність передачі PIN між банкоматом або POS-терміналом і головним комп'ютером банку виключає його перехоплення зловмисником або нав'язування результатів порівняння;
- спрощення роботи зі створення програмного забезпечення системи, оскільки вже немає необхідності дій в реальному масштабі часу.

3.4.2. Забезпечення безпеки електронних платежів через мережу Інтернет

Традиційним і перевіреним способом електронної торгівлі, який бере свій початок від звичайної торгівлі за каталогами, є оплата товарів і послуг кредитною карткою по телефону. В цьому випадку покупець замовляє на Web-сервері список товарів, які він хотів би купити, і потім повідомляє по телефону номер своєї кредитної картки продавцю комерційної фірми. Далі відбувається звичайна авторизація картки, а списування грошей з рахунку покупця проводиться лише у момент відправки товару поштою або з кур'єром.

Для того щоб покупець-власник кредитної картки міг без побоювань розплатитися за покупку через мережу, необхідно мати надійніший, відпрацьований механізм захисту передачі електронних платежів. Такий принципово новий підхід полягає в негайній авторизації і шифруванні фінансової інформації в мережі Інтернет з використанням схем SSL і SET.

Протокол "Безпечні електронні транзакції" SET (Secure Electronic Transactions), розроблений компаніями Visa і Master Card, припускає шифрування виключно фінансової інформації. Протягом тривалого часу протокол SET обговорювався ученими всього світу. Головна вимога, що ставилася до нього, – забезпечити повну безпеку і конфіденційність здійснення операцій. На сьогодні технічні умови протоколу, що забезпечують безпеку, визнані оптимальними. Введення цього протоколу в дію дасть власникам пластикових карток можливість використовувати комп'ютерні мережі при проведенні фінансових операцій, не побоюючись за подальшу долю своїх платіжних засобів.

Стандарт SET обіцяє істотно збільшити обсяг продажів за кредитними картками через Інтернет. Сукупна кількість потенційних покупців-власників карток Visa і MasterCard по всьому світу перевищує 700 мільйонів чоловік. Забезпечення безпеки електронних транзакцій для такого числа покупців може привести до помітних змін, що виражаються в зменшенні собівартості

транзакції для банків і процесингових компаній.

Для того щоб забезпечити повну безпеку і конфіденційність здійснення операцій, протокол SET повинен гарантувати неодмінне дотримання наступних умов.

Абсолютна конфіденційність інформації. Власники карток повинні бути упевнені в тому, що їх платіжна інформація надійно захищена і доступна тільки вказаному адресату. Це є неодмінною умовою розвитку електронної торгівлі.

Повне збереження даних. Учасники електронної торгівлі повинні бути упевнені в тому, що при передачі від відправника до адресата зміст повідомлення залишиться незмінним. Повідомлення, що відправляються власниками карток комерсантам, містять інформацію про замовлення, персональні дані і платіжні інструкції. Якщо в процесі передачі зміниться хоча б один з компонентів, то транзакція не буде оброблена належним чином. Тому, щоб уникнути помилок, протокол SET повинен забезпечити засоби, що гарантують збереження і незмінність повідомлень, які відправляються. Одним з таких засобів є використання цифрових підписів.

Автентифікація (встановлення достовірності) рахунка власника картки. Використання цифрових підписів і сертифікатів власника картки гарантує автентифікацію рахунка власника картки і підтвердження того, що власник картки є законним користувачем даного номера рахунка.

Власник картки повинен бути упевнений, що комерсант дійсно має право проводити фінансові операції з фінансовою установою. Використання цифрових підписів і сертифікатів комерсанта гарантує власнику картки, що можна безпечно вести електронну торгівлю.

Протокол SET змінює спосіб взаємодії учасників системи розрахунків. У даному випадку електронна транзакція починається з власника картки, а не з комерсанта або еквайєра.

Комерсант пропонує товар для продажу або надає послуги за плату. Протокол SET дозволяє комерсанту пропонувати електронні взаємодії, які можуть безпечно використовувати власники карток.

Еквайєром (одержувачем) є фінансова установа, яка відкриває рахунок комерсанту і обробляє авторизації і платежі по кредитних картках. Еквайєр обробляє повідомлення про платежі, переведені комерсанту за допомогою платіжного міжмережевого інтерфейсу. При цьому протокол SET гарантує, що при взаємодіях, які здійснює власник картки з комерсантом, інформація про рахунок кредитної картки залишатиметься конфіденційною.

Фінансові установи створюють асоціації банківських кредитних карток, які захищають і рекламують даний тип картки, створюють і вводять у дію правила використання кредитних карток, а також організовують мережі для зв'язку фінансових установ один з одним.

Системи кредитних карток затвердилися, значною мірою, як платіжний засіб для придбання товарів безпосередньо у продавця. Основна відмінність використання кредитних карток у мережі Інтернет полягає в тому, що відповідно до стандарту SET для захисту транзакцій електронної торгівлі використовуються процедури шифрування і цифрового підпису.

Мережа Інтернет розрахована на одночасну роботу мільйонів користувачів, тому в комерційних Інтернет-додатках неможливо використовувати тільки симетричні криптосистеми з секретними ключами (DES, ГОСТ28147-89). У зв'язку з цим застосовуються також асиметричні криптосистеми з відкритими ключами. Шифрування з використанням відкритих ключів припускає, що у комерсанта і покупця є по два ключі –один відкритий, який може бути відомий третім особам, а інший – приватний (секретний), відомий тільки одержувачу інформації.

Правила SET передбачають первинне шифрування повідомлення з використанням випадковим чином генерованого симетричного ключа, який, у свою чергу, шифрується відкритим ключем одержувача повідомлення. В результаті утворюється так званий електронний конверт. Одержувач повідомлення розшифровує електронний конверт за допомогою приватного (секретного) ключа, щоб одержати симетричний ключ відправника. Далі симетричний ключ відправника використовується для розшифрування присланого повідомлення.

Цілісність інформації і автентифікації учасників транзакції гарантується використанням електронного цифрового підпису.

Для захисту операцій від шахрайства і зловживань організовані спеціальні центри (агентства) сертифікації в Інтернет, які стежать за тим, щоб кожен учасник електронної комерції одержував унікальний електронний сертифікат. У цьому сертифікаті за допомогою секретного ключа сертифікації зашифрований відкритий ключ даного учасника комерційної операції. Сертифікат генерується на певний час, і для його отримання необхідно подати до центру сертифікації документ, який підтверджує особу учасника (для юридичних осіб – їх легальну реєстрацію), і потім, маючи "на руках" відкритий ключ центру сертифікації, брати участь в операціях.

Розглянемо приклад шифрування. Комерсант Ірина хоче направити зашифроване повідомлення про товар покупцю Івану у відповідь на його запит. Ірина пропускає опис товару через однонаправлений алгоритм, щоб набути унікального значення, відомого як дайджест повідомлення. Це свого роду цифровий зліпок з опису товару, який згодом буде використаний для перевірки цілісності повідомлення. Потім Ірина шифрує цей дайджест повідомлення особистим (секретним) ключем для підпису, щоб створити цифровий підпис.

Після цього Ірина створює довільний симетричний ключ і використовує його для шифрування опису товару, свого підпису і копії свого сертифіката,

який містить її відкритий ключ для підпису. Для того щоб розшифрувати опис товару, Івану знадобиться захищена копія цього довільного симетричного ключа.

Сертифікат Івана, який Ірина повинна була одержати до ініціації безпечного зв'язку з ним, містить копію його відкритого ключа для обміну ключами. Щоб забезпечити безпечну передачу симетричного ключа, Ірина шифрує його, користуючись відкритим ключем Івана для обміну ключами. Зашифрований ключ, який називається цифровим конвертом, прямує до Івана разом із зашифрованим повідомленням.

Нарешті, вона відправляє повідомлення Івану, що складається з таких компонентів:

– симетрично зашифрованого опису товару, підпису і свого сертифіката;

– асиметрично зашифрованого симетричного ключа (цифровий конверт).

Продовжимо попередній приклад і розглянемо процедуру розшифрування.

Іван одержує зашифроване повідомлення від Ірини і перш за все розшифровує цифровий конверт особистим (секретним) ключем для обміну ключами з метою витягування симетричного ключа. Потім Іван використовує цей симетричний ключ для розшифрування опису товару, підпису Ірини і її сертифіката. Далі Іван розшифровує цифровий підпис Ірини за допомогою її відкритого ключа для підпису, який одержує з її сертифіката. Тим самим він відновлює оригінальний дайджест повідомлення з описом товару. Потім Іван пропускає опис товару через той же однонаправлений алгоритм, який використовувався Іриною, і одержує новий дайджест повідомлення з розшифрованим описом товару.

Потім Іван порівнює свій дайджест повідомлення з дайджестом, одержаним з цифрового підпису Ірини. Якщо вони збігаються, Іван отримує підтвердження, що зміст повідомлення не змінився під час передачі і воно підписане з використанням особистого (секретного) ключа для підпису Ірини. Якщо ж дайджести не збігаються, це означає, що повідомлення було відправлено з іншого місця або було змінено після того, як було підписано. У цьому випадку Іван виконує певні дії, наприклад, повідомляє Ірину або відкидає отримане повідомлення.

Протокол SET вводить нове застосування цифрових підписів, а саме – використання подвійних цифрових підписів. У рамках протоколу SET подвійні цифрові підписи використовуються для зв'язку замовлення, відправленого комерсанту, з платіжними інструкціями, що містять інформацію про рахунок і відправленими банку.

Наприклад, покупець Іван хоче направити комерсанту Ірині пропозицію купити одиницю товару і авторизацію своєму банку на переказ грошей, якщо Ірина прийме його пропозицію. В той же час Іван не хоче, щоб у банку

прочитали умови його пропозиції, як і не хоче, щоб Ірина прочитала його інформацію про рахунок. Крім того, Іван хоче пов'язати свою пропозицію з переводом так, щоб гроші були перераховані тільки в тому випадку, якщо Ірина прийме його пропозицію.

Все вищесказане Іван може виконати за допомогою цифрового підпису під обома повідомленнями за допомогою однієї операції підпису, яка створює подвійний цифровий підпис. Подвійний цифровий підпис створюється шляхом формування дайджесту обох повідомлень, скріплення двох повідомлень разом, обчислення дайджесту підсумку попередніх операцій і шифрування цього дайджесту особистим ключем для підпису автора. Автор зобов'язаний включити також дайджест іншого повідомлення, з тим, щоб одержувач перевірив подвійний підпис.

Одержувач будь-якого з цих повідомлень може перевірити його достовірність, генеруючи дайджест з своєї копії повідомлення, пов'язуючи його з дайджестом іншого повідомлення (у порядку, передбаченому відправником) і обчислюючи дайджест для одержаного підсумку. Якщо знов освічений дайджест відповідає розшифрованому подвійному підпису, то одержувач може довіряти достовірності повідомлення.

Якщо Ірина приймає пропозицію Івана, вона може відправити повідомлення банку, вказавши на свою згоду і включивши дайджест повідомлення з пропозицією Івана. Банк може перевірити достовірність авторизації Івана на перелік і дайджесту повідомлення з пропозицією Івана, наданого Іриною, щоб підтвердити подвійний підпис. Таким чином, банк може перевірити достовірність пропозиції на підставі подвійного підпису, але банк не зможе прочитати умови пропозиції.

Використання сертифікатів. Альтернативою безпечній передачі ключа служить використання довіреної третьої сторони-центру сертифікації (агентства по сертифікатах) для підтвердження того, що відкритий ключ належить саме власнику картки.

Центр сертифікації створює повідомлення, що містить ім'я власника картки і його відкритий ключ, після пред'явлення власником картки доказів ідентифікації особи (водійські права або паспорт). Таке повідомлення називається сертифікатом. Сертифікат забезпечується підписом центру сертифікації і містить інформацію про ідентифікацію власника, а також копію одного з відкритих ключів власника.

Учасники протоколу SET мають дві пари ключів і мають у своєму розпорядженні два сертифікати. Обидва сертифікати створюються і підписуються одночасно центром сертифікації.

Сертифікати власників карток функціонують як електронний еквівалент кредитних карток. Вони забезпечуються цифровим підписом фінансової установи, і тому не можуть бути змінені третьою стороною. Ці сертифікати

містять номер рахунка і термін дії, які шифруються з використанням однонаправленого алгоритму хешування. Якщо номер рахунка і дата закінчення дії відомі, то зв'язок з сертифікатом можна підтвердити, проте цю інформацію неможливо одержати шляхом вивчення даного сертифіката. В рамках протоколу SET власник картки подає інформацію про рахунок у той платіжний міжмережевий інтерфейс, де проводиться даний зв'язок.

Сертифікат видається власнику картки тільки з дозволу фінансової установи-емітента картки. Запрошуючи сертифікат, власник картки вказує свій намір використовувати торгівлю електронними засобами. Ці сертифікати передаються комерсантам разом із запитами про покупку і зашифрованими платіжними інструкціями. Коли комерсант одержує сертифікат власника картки, він може не сумніватися в тому, що номер рахунка підтверджено фінансовою установою.

Сертифікати комерсантів є електронним аналогом фірмової картки, яка виставляється у вітрині електронного магазину. Ці сертифікати забезпечені цифровим підписом фінансової установи комерсанта і, отже, не можуть бути змінені третьою стороною. Сертифікати служать гарантією того, що комерсант має діючу угоду з еквайєром.

Комерсант повинен мати щонайменше одну пару сертифікатів для того, щоб брати участь в операційному середовищі SET, але в одного комерсанта може бути множина пар сертифікатів для кожного типу кредитних карток, які він приймає до оплати.

Сертифікати платіжних міжмережевих інтерфейсів видаються еквайєрам або їх обробникам для систем, які обробляють авторизації і одержують повідомлення. Ключ шифрування конкретного інтерфейсу, який власник картки одержує з цього сертифіката, використовується для захисту інформації про рахунок власника картки. Сертифікати платіжного інтерфейсу видаються еквайєру оператором карток певного типу.

Сертифікати еквайєрів видаються еквайєрам для того, щоб вони могли приймати і обробляти запити про сертифікати, ініційовані комерсантами. Еквайєри одержують сертифікати від кожної асоціації кредитних карток.

Сертифікати емітентів потрібні емітентам для того, щоб користуватися послугами центру сертифікації, який може приймати і обробляти запити про сертифікати безпосередньо від власників карток по відкритих і приватних мережах. Емітенти одержують сертифікати від асоціації кредитних карток.

Сертифікати SET перевіряються в ієрархії довіри (див. рис. 3.24). Кожен сертифікат пов'язаний з сертифікатом підпису того об'єкта, який забезпечив його цифровим підписом. Йдучи по "дереву довіри" до відомої довіреної сторони, можна бути впевненим у тому, що сертифікат є дійсним. Наприклад, сертифікат власника картки пов'язаний з сертифікатом емітента (або асоціації за дорученням емітента), який у свою чергу пов'язаний з кореневим ключем

через сертифікат асоціації.

Відкритий ключ для кореневого підпису відомий всім програмним засобам SET і може бути використаний для перевірки кожного з сертифікатів. Кореневий ключ розповсюджується в сертифікаті з автопідписом. Цей сертифікат кореневого ключа буде доступний постачальникам програмного забезпечення для включення в їх програмні засоби.

Протокол SET визначає множину протоколів транзакцій, які використовують криптографічні засоби для безпечного ведення електронної комерції. Серед цих протоколів транзакцій – реєстрація власника картки, реєстрація комерсанта, запит про покупку, авторизація платежу, отримання платежу.

Нові досягнення в галузі безпеки використання кредитних карток, реалізовані в стандарті SET, здатні задовольнити найдовірливіших клієнтів електронних платіжних систем, оскільки усуваються всі їх побоювання шляхом впровадження засобів шифрування для скремблірування кредитної картки в такому порядку, щоб її могли читати тільки продавець і покупець.

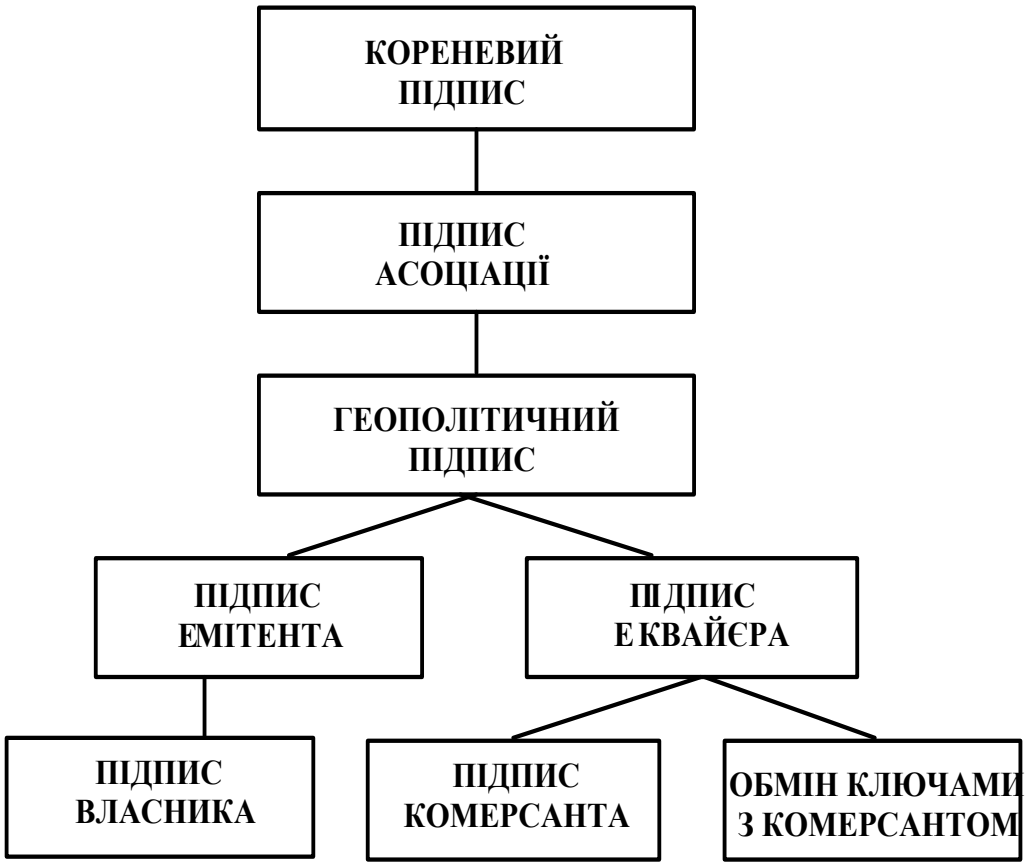


Рисунок 3.24 – Ієрархії довіри для перевірки сертифікатів SET

Таким чином, системи такого типу мають ряд переваг.

Гроші клієнта знаходяться під надійним наглядом банку. Якщо клієнт втратить картку, то його рахунок все одно буде пов'язаний з його ім'ям. На

відміну від систем з використанням готівки, у банку є можливість перевірити залишок на рахунку клієнта, тому гроші клієнта не втрачаються.

Відпадає необхідність у відкритті нового рахунку. У банку для обробки транзакцій даного типу клієнт може продовжувати користуватися діючим рахунком і кредитною карткою. Цей чинник має велике значення на початкових стадіях електронної торгівлі в WWW мережі Інтернет.

Проте є й недолік, причому істотний – відсутність конфіденційності. На відміну від транзакцій з електронною готівкою, які є анонімними, в транзакціях з кредитними картками ім'я клієнта жорстко пов'язане з рахунком.

У даний час найбільшого поширення набули два програмно-апаратних рішення, запропоновані компаніями Microsoft, VeriFone і Netscape.

Обидва вони припускають використання такого набору компонентів:

- клієнтського комп'ютера, що має доступ до Інтернет і Web-browser;
- W-сервера електронної торгівлі, на якому ведеться каталог товарів і приймаються зашифровані запити клієнтів на покупку тих або інших товарів;
- засобів для забезпечення взаємної конвертації протоколів Інтернет і стандартних протоколів авторизації (ISO 8583 та ін.).

Таким чином, мережа Інтернет – це об'єднання в масштабі всієї планети групи мереж, яке використовує єдиний протокол для передачі даних. Велике число організацій зараз приєднуються до Інтернету для того, щоб скористатися перевагами і ресурсами Інтернету. Бізнесмени й державні організації використовують Інтернет в найрізніших цілях, включаючи обмін електронною поштою, розповсюдження інформації серед зацікавлених осіб і проведення досліджень. Багато організацій приєднують існуючі локальні мережі до Інтернету, аби робочі станції цих локальних обчислювальних мереж могли дістати прямий доступ до сервісів Інтернету.

Приєднання до Інтернету може дати величезні переваги, хоча при цьому потрібно серйозно врахувати питання, пов'язані з безпекою з'єднання. Існують достатньо серйозні ризики безпеки, пов'язані з Інтернетом, які часто є неочевидними для користувачів-новачків. Зокрема, в світі спостерігається діяльність зловмисників, при цьому є багато вразливих місць, які можуть її полегшити. Дії зловмисників важко передбачити і деколи її буває важко виявити і припинити, тому багато організацій вже змарнували багато часу і зазнали значних фінансових втрат через діяльність зловмисників.

3.4.3. Управління доступом в мережевій технології «клієнт-сервер» для базових операційних систем

Правильне управління доступом до вмісту веб-і FTP-вузлів є основним елементом організації захищеного веб-сервера. Використовуючи можливості Windows і системи безпеки IIS, можна ефективно управляти доступом

користувачів до вмісту веб-і FTP-вузлів. Управління доступом може бути організовано на декількох рівнях, від усього веб-або FTP-вузла до окремих файлів.

Анонімний доступ. Анонімний доступ, який є найбільш широко використовуваним методом доступу до веб-вузлів, дозволяє будь-якому користувачеві відвідати загальні області на веб-сайті, але запобігає несанкціонованому доступу до важливих засобів адміністрування веб-сервера і приватним відомостям. Наприклад, якщо порівняти веб-сервер з музеєм, то дозвіл анонімного доступу аналогічно запрошення відвідувати загальні галереї та експозиції музею. Однак деякі кімнати повинні бути закриті, наприклад службові приміщення та лабораторії, які не повинні відвідуватися сторонніми. Аналогічно цьому, при налаштуванні анонімного доступу до веб-серверу слід задати дозволу NTFS, що забороняють звичайним користувачам доступ до особистих файлів і каталогів.

Для входу всіх користувачів на веб-сервер за замовчуванням використовується анонімний обліковий запис. При установці сервера створюється спеціальний обліковий запис анонімного користувача з ім'ям IUSR_імя Компьютера. Наприклад, для комп'ютера з ім'ям SalesDept1 обліковий запис анонімного користувача отримає ім'я IUSR_SalesDept1. Кожен веб-вузол на сервері може використовувати для входу анонімних користувачів або одну і ту ж, або різні облікові записи. Диспетчер локальних користувачів і груп Windows дозволяє створити новий обліковий запис для анонімного доступу.

WebDAV представляє собою розширення стандарту HTTP 1.1 для надання будь-якого носія інформації, наприклад файлової системи, при HTTP-з'єднанні. Використовуючи реалізацію WebDAV в IIS 5.0, можна надати можливість редагувати, переміщати, шукати або видаляти файли, каталоги і їх властивості на сервері. WebDAV конфігурується за допомогою установки дозволів веб-сервера.

Можна встановити дозволу WebDAV для:

– Пошуку каталогів і файлів і їх властивостей.

– Створення, зміни, видалення і перегляду каталогів і файлів і їх властивостей.

– Зберігання та вилучення настроюються властивостей файлів і каталогів.

– Блокування файлів в спільних робочих середовищах.

WebDAV функціонує в файлових системах і FAT, і NTFS.

WebDAV є реалізацією запропонованого проекту HTTP 1.1 і, отже, недоступний для служб, які не є службами HTTP, наприклад вузлів FTP.

Як працює управління доступом. Для того щоб управляти доступом користувачів до вмісту веб-сервера, слід задати правильну конфігурацію засобів безпеки Windows і веб-сервера. Коли користувач намагається отримати

доступ до веб-сервера, сервер виконує ряд операцій з перевірки користувача і визначенню дозволеного рівня доступу. На рис. 3.35 наведено структуру процесу управління доступу:

1. Клієнт запитує ресурс на сервері.
2. Сервер, якщо його конфігурація припускає це, запросить у клієнта відомості для перевірки автентичності. Оглядач або запропонує користувачеві ввести ім'я користувача та пароль, або передасть ці відомості автоматично.
3. IP- адресу клієнта перевіряється на обмеження IP- адрес , закладені в IIS. Якщо доступ для IP- адреси заборонено , запит не виконується і користувач отримує повідомлення про помилку "403 Access Forbidden".
4. IIS перевіряє допустимість облікового запису користувача. Якщо обліковий запис користувача не є припустимою , запит не виконується і користувач отримує повідомлення про помилку "403 Access Forbidden".
5. IIS перевіряє наявність у користувача веб- дозволів для запитуваного ресурсу . Якщо користувач не має відповідного дозволу , запит не виконується і користувач отримує повідомлення про помилку "403 Access Forbidden".
6. Будь-які модулі безпеки від незалежних розробників , додані адміністратором вузла веб , використовуються на цьому етапі.
7. IIS перевіряє дозволу NTFS для ресурсу. Якщо користувач не має дозволів NTFS для ресурсу , запит не виконується і користувач отримує повідомлення про помилку "401 Access Forbidden".
8. Якщо користувач має дозвіл NTFS , запит виконується.

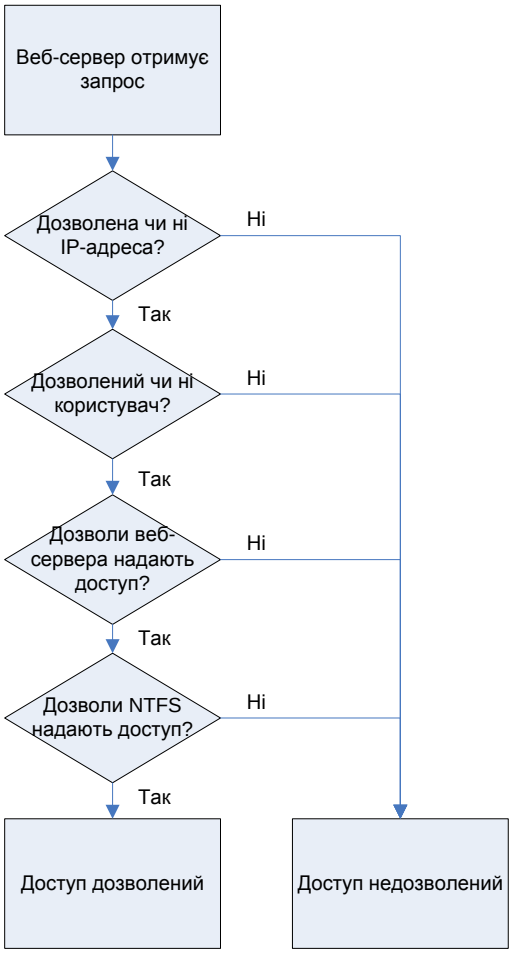


Рисунок. 3.25 – Структура процессу управління доступу

Обмеження доступу для IP-адрес. Веб-сервер може бути налаштований таким чином, щоб забороняти доступ до вмісту веб-вузла з боку певних комп'ютерів, груп комп'ютерів і цілих мереж. Коли користувач робить першу спробу отримати доступ до вмісту веб-сервера, сервер порівнює IP-адреса комп'ютера користувача зі списком обмежень IP-адрес.

Дозволи веб –сервера. У конфігурації веб-сервера є можливість задати дозволу для конкретних вузлів, каталогів і файлів. Ці дозволи будуть діяти для всіх користувачів, незалежно від наявних у них конкретних прав доступу. Наприклад, можна відключити дозвіл « Читання » для конкретного веб-вузла, щоб заборонити доступ користувачів під час оновлення вмісту сайту. В результаті, при спробі доступу до веб-вузла сервер буде повертати повідомлення про помилку "Access Forbidden". Після включення дозволу «Читання» всі користувачі отримають можливість переглядати веб-вузол, за винятком випадку, коли встановлені обмеження файлової системи NTFS на перегляд вузла користувачами.

Рівні веб-дозволів включають:

1. Читання (вибирається за замовчуванням). Користувачі можуть переглядати вміст файлу і його властивості.
2. Запис. Користувачі можуть змінювати вміст файлу і його властивості.
3. Доступ до тексту сценарію. Користувачі отримують доступ до

вихідних файлів. Якщо вибрано дозвіл «Читання», вихідний текст може бути прочитаний. Якщо встановлено дозвіл «Запис», можна проводити запис і в вихідні тексти. «Доступ до тексту сценарію» дозволяє доступ до вихідних текстів для файлів, наприклад сценаріями в додатку ASP. Ця можливість доступна лише при встановлених дозволах «Читання » або «Запис».

4. Огляд каталогів. Користувачі можуть переглядати списки і сімейства файлів.

5. Запис у журнал. Запис у журнал робиться для кожного відвідування вузла веб.

6. Індексація каталогу. Дозволяє службі індексації створити індекс для ресурсу.

Установки дозволів веб-сервера впливають на те , які команди HTTP можуть бути використані для вузла , віртуального каталогу або файлу.

Дозволи NTFS. IIS залежить від дозволів NTFS при забезпеченні безпеки окремих файлів і каталогів від несанкціонованого доступу. На відміну від дозволів веб-сервера , які застосовуються до всіх користувачів, дозволу NTFS дозволяють точно вказати, які користувачі можуть отримувати доступ до вмісту і як ці користувачі можуть обробляти вміст (табл. 3.13).

Рівні дозволів NTFS включають:

1. Повний доступ. Користувачі можуть змінювати , додавати , переміщати і видаляти файли, властивості, пов'язані з ними, і каталоги. Крім цього, можна змінити дозволи для всіх файлів і підкаталогів.

2. Зміна. Користувачі можуть переглядати і змінювати файли та їх властивості, включаючи видалення і додавання файлів в каталог або властивостей файлу до файлу.

3. Читання та виконання. Користувачі можуть запускати виконувані файли, включаючи сценарії.

4. Список вмісту папки. Користувачі можуть переглядати список вмісту папки.

5. Читання. Користувачі можуть переглядати файли та їх властивості.

6. Запис. Користувачі можуть записувати файл.Немає доступу Коли жоден з прапорців не встановлений, користувачі не мають ніякого доступу до ресурсу, навіть якщо користувач має доступ до каталогу більш високого рівня.

Установка дозволу «Ні доступу » до ресурсу для облікового запису IUSR_Імя Компьютера призведе до заборони доступу анонімних користувачів до цього ресурсу.

Таблиця 3.13 – Обліковий запис користувача

Обліковий запис користувача Windows 2000 або групи користувачів	Дозволи
MYSERVER/Administrators	Повний доступ
MYSERVER/JeffSmith	Зміни
MYSERVER/Guests	Немає доступу

Крім членів групи адміністраторів дозвіл на зміну цього файлу надано тільки обліковому запису з ім'ям JeffSmith. Для звичайних користувачів, які входять як члени групи гостей Windows, доступ до цього файлу заборонений в явному вигляді.

Після завдання дозволів NTFS необхідно вказати для веб-сервера спосіб перевірки ідентифікації або автентичності користувачів перед наданням їм доступу до файлів з обмеженим доступом. Є можливість задати в налаштуванні веб-сервера необхідність перевірки автентичності користувачів, при якій від користувачів для підключення потрібно припустиме ім'я облікового запису Windows і пароль.

Неправильна установка таблиць управління доступом NTFS може змусити оглядач на комп'ютері клієнта запитувати відомості про користувача. Наприклад, користувач не має доступу до файлу (згідно налаштуванням таблиць управління доступом), IIS видає повідомлення про заборону доступу, що змусить оглядач запросити у користувача інше ім'я користувача і пароль.

Список джерел інформації

1. Мельников В.В. Защита информации в компьютерных системах / В.В. Мельников. – М.: Финансы и статистика. Электроинформ, 1997. – 367 с.
2. Романец Ю.В. Защита информации в компьютерных системах и сетях / П.А. Тимофеев, В.Ф. Шаньгин. – М.: Радио и связь, 2001. – 376 с.
3. Стенг Д. Секреты безопасности сетей / Д. Стенг, С. Мун. – К.: Диалектика, 1995. – 544 с.
4. Столингс. В. Криптография и защита сетей. Принципы и практика / В. Столингс. – М.: «Вильямс», 2001. – 672 с.
5. Протоколы Интернет. Энциклопедия / Ю.А. Семенов. – М.: Горячая линия – Телеком, 2005. – 405 с.
6. Семенов Ю.А. Протоколы Internet для электронной торговки / Ю.А. Семенов.– М.: Горячая линия – Телеком, 2005. – 366 с.
7. Горбенко І.Д., Горбенко Ю.І., Прикладна криптологія. Теорія. Практика. Застосування: монографія. – Х.:Видавництво «Форт», 2012, 870 с.

Контрольні запитання

1. Наведіть приклади класів електронної комерції і основні засоби їх безпеки.
2. Наведіть приклади основних сервісів, що забезпечуються зв'язком з мережею Інтернет, і засоби їх безпеки.
3. Дайте визначення брандмауера.
4. Які основні різновиди заміни мережевої адреси використовуються в

брандмауерах?

5. Надайте структуру системи захисту інформації на фірмі, основним “інтернетівським” сервісом якої є електронна пошта, і ставлять високі вимоги до постійної доступності до сервісу.

6. Які основні функції системи PGP?

7. Опишіть алгоритм створення повідомлення системою PGP.

8. Опишіть алгоритм отримання повідомлення системою PGP.

9. Проілюструйте загальний формат повідомлення PGP.

10. Опишіть алгоритм визначення ступеня довіри ключам.

11. Укажіть причини, за якими алгоритми DES, “потрійний” DES з двома ключами RC2 і RC5 підходить або не підходить для PGP.

12. Опишіть загальні процедури підготовки повідомлень S/MIME. Які криптографічні алгоритми використовуються в S/MIME?

13. Дайте короткий опис типів, підтипів і вмісту S/MIME.

14. Яким чином користувачі оформляють і відновлюють сертифікати S/MIME?

15. Які механізми захисту повинні бути реалізовані для забезпечення функцій захисту інформації на окремих вузлах системи електронних платежів?

16. Опишіть набір функцій, реалізованих в смарт-картці.

4. ОСОБЛИВОСТІ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ІНФОРМАЦІЇ В СИСТЕМАХ МОБІЛЬНОГО ЗВ'ЯЗКУ СТАНДАРТУ GSM

Стільникові системи мобільного зв'язку нового покоління в змозі обслужити всіх потенційних користувачів, якщо буде гарантовано безпеку переданої інформації.

У стандарті GSM термін «безпека» розуміють як виключення несанкціонованого використання системи і забезпечення конфіденційності переговорів мобільних абонентів. Визначено такі механізми безпеки в стандарті GSM:

- автентифікація;
- конфіденційність передачі даних;
- конфіденційність абонента;
- конфіденційність напрямів з'єднання абонентів.

Захист сигналів керування і даних користувача здійснюється тільки радіоканалом. Режими безпеки в стандарті визначаються рекомендаціями, наведеними в таблиці. 4.1.

Таблиця 4.1. – Рекомендації щодо забезпечення інформаційної безпеки в мережах GSM

Стандарт	Режим безпеки	Механізм безпеки
GSM 02.09	Аспекти конфіденційності	Визначає характеристики безпеки, вживані в мережах. Регламентується їх застосування в мобільних станціях і мережах
GSM 03.20	Конфіденційність, пов'язана з функціями мережі	Визначає функції, необхідні для забезпечення характеристик безпеки, що розглядаються в Рекомендаціях GSM 02.09
GSM 03.21	Алгоритми забезпечення конфіденційності	Визначає криптографічні алгоритми в системі зв'язку
GSM 02.17	Модулі достовірності абонентів (SIM)	Визначає основні характеристики модуля SIM

Надалі розглянемо механізми безпеки в стандарті GSM, при цьому використовуватимемо терміни і позначення, прийняті в Рекомендаціях GSM.

4.1. Механізми автентифікації

Для виключення несанкціонованого використання ресурсів системи в

стандарті GSM реалізуються механізми автентифікації – перевірки достовірності абонента.

Кожен мобільний абонент на час користування системою зв'язку одержує стандартний модуль достовірності абонента (SIM-карту), який містить:

- міжнародний ідентифікаційний номер мобільного абонента (IMSI);
- свій індивідуальний ключ автентифікації (Ki);
- алгоритм автентифікації (A).

За допомогою закладеної в SIM інформації в результаті взаємного обміну даними між мобільною станцією і мережею здійснюється повний цикл автентифікації і вирішується доступ абонента до мережі.

Процедура перевірки мережею достовірності абонента реалізується таким чином.

Мережа передає випадковий номер (RAND) на мобільну станцію. Мобільна станція визначає значення відгуку (SRES), використовуючи RAND, K_i і алгоритм A3 :

$$SRES = K_i[RAND].$$

Мобільна станція посилає обчислене значення SRES у мережу, яка звіряє значення прийнятого SRES із значенням SRES, обчисленим мережею. Якщо обидва значення збігаються, мобільна станція зможе здійснювати передачу повідомлень. Інакше зв'язок перерветься, і індикатор мобільної станції покаже, що розпізнання не відбулося.

На користь забезпечення безпеки обчислення SRES відбувається в рамках SIM. Інформація, що не підлягає захисту (така, як K_i), не піддається обробці в модулі SIM.

Процедура автентифікації ілюструється на рис. 4.1.

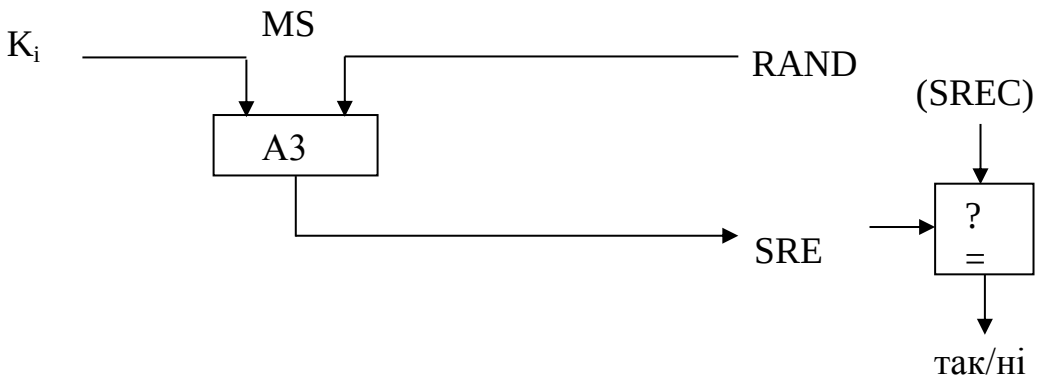


Рисунок 4.1 – Процедура автентифікації

4.2. Конфіденційність передачі мовної інформації

Усі конфіденційні повідомлення повинні передаватися в режимі захисту

інформації. Для забезпечення конфіденційності переданої радіоканалом інформації, використовується **ключ шифрування**. Алгоритм формування ключів шифрування (А8) зберігається в модулі SIM.

Після прийому випадкового номера RAND мобільна станція обчислює, окрім відгуку SRES, також і ключ шифрування ($К_с$), використовуючи RAND, $К_i$ і алгоритм А8 (рис. 4.2):

$$К_с = К_i [RAND].$$

Ключ шифрування $К_с$ не передається радіоканалом. Як мобільна станція, так і мережа обчислюють ключ шифрування, який використовується іншими мобільними абонентами. На користь забезпечення безпеки обчислення $К_с$ відбувається в SIM.

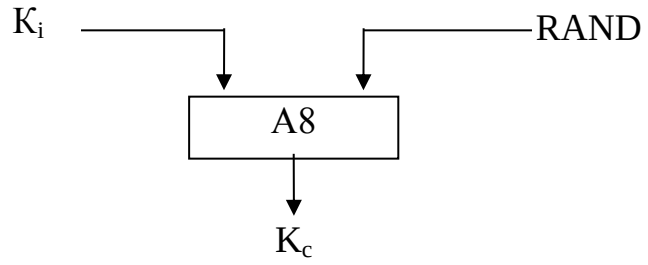


Рисунок 4.2 – Процедура формування ключа шифрування

Окрім випадкового числа RAND, мережа посилає мобільній станції **числову послідовність ключа шифрування**. Це число пов'язане з дійсним значенням $К_с$ і дозволяє уникнути формування неправильного ключа. Число зберігається мобільною станцією і міститься в кожному першому повідомленні, переданому в мережу. Деякі мережі ухвалюють рішення про наявність числової послідовності діючого ключа шифрування у випадку, якщо необхідно приступити до розпізнавання або якщо виконується попереднє розпізнавання, використовуючи правильний ключ шифрування. В деяких випадках це допущення реально не забезпечується.

Для **установлення режиму шифрування** мережа передає мобільній станції команду CMC (Ciphering Mode Command) на перехід у режим шифрування. Після отримання команди CMC мобільна станція, використовуючи ключ, що є у неї, приступає до шифрування і розшифрування повідомлень. Потік переданих даних шифрується біт за бітом або потоковим шифром, використовуючи алгоритм шифрування А5 і ключ шифрування $К_с$. Процедура встановлення режиму шифрування показана на рис. 4.3.

4.3. Принципи виконання алгоритмів сімейства А5/х

Спочатку французькими військовими фахівцями-криптографами був

розроблений потоковий шифр для використання виключно у військових цілях. У кінці 80-х років минулого століття для стандарту GSM було потрібно створення нової, сучасної системи безпеки. Як алгоритм шифрування даних була використана французька розробка. Цей шифр забезпечував достатньо хорошу захищеність, що забезпечувало конфіденційність розмови. Спочатку експорт стандарту з Європи не передбачався, але незабаром у цьому виникла необхідність. Саме тому A5 перейменували в A5/1 і стали поширювати в Європі і США. Для решти країн алгоритм модифікували, значно знизивши криптостійкість шифру. A5/2 був спеціально розроблений як експортний варіант для країн, що не входили в Євросоюз. Криптостійкість A5/2 була знижена додаванням ще одного регістра (завдовжки 17 біт), що керує зсувами інших. У A5/0 шифрування повністю відсутнє. На даний час розроблений також алгоритм A5/3, заснований на алгоритмі Касумі і затверджений для використання в мережах 3G. Модифікації мають позначення A5/x.

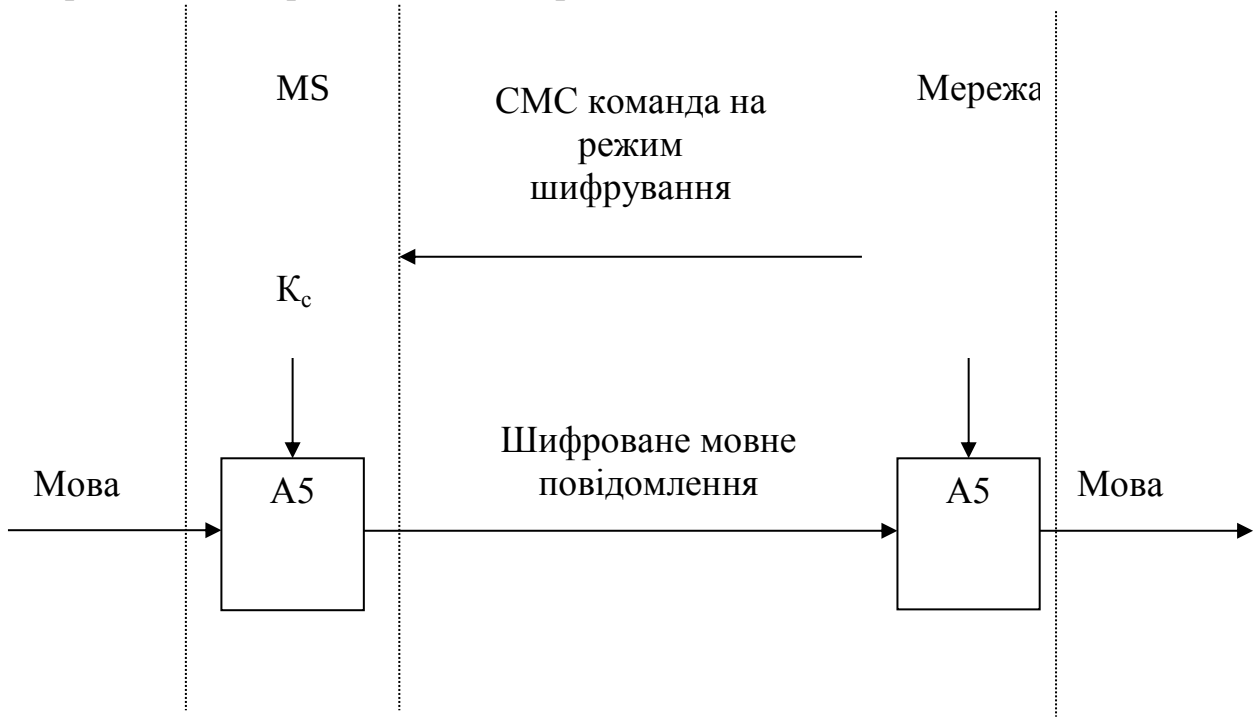


Рисунок 4.3 – Процедура встановлення режиму шифрування

Офіційно французька розробка не публікувалася, і її структура не розголошувалася. Це пов'язано з тим, що розробники покладалися на безпеку за рахунок невідомості. Дані надавалися операторам GSM тільки з потреби. Проте до 1994 року деталі алгоритму A5 стали відомі: британська телефонна компанія (British Telecom) передала всю документацію, що стосується стандарту, Бредфордському університету для аналізу, не уклавши угоду про нерозголошення інформації. Крім того, матеріали про стандарт з'явилися на одній конференції в Китаї. В результаті, чого схема поступово попала в широкі кола. У тому ж році кембріджські вчені Ross Anderson і Michael Roe опублікували відновлену за цими даними криптосхему. Остаточно алгоритм

був поданий в роботі Йована Голіча на конференції Eurocrypt'97.

Сімейство алгоритмів A5 є потоковими шифрами, побудованими на регістрах зсуву з лінійним зворотним зв'язком.

Алгоритм A5/1 має таку структуру:

- три регістри (R1, R2, R3) мають довжини 19, 22 і 23 біт;
- багаточлени зворотних зв'язків обчислюються за формулами 4.1, 4.2 і 4.3 для регістрів R1, R2 і R3 відповідно:

$$x^{18} + x^{17} + x^{16} + x^{13} + 1 \quad (4.1)$$

$$x^{21} + x^{20} + 1 \quad (4.2)$$

$$x^{22} + x^{21} + x^{20} + x^{7} + 1 \quad (4.3)$$

- блок керування тактуванням. Керування тактуванням здійснюється таким чином: у кожному регістрі є біти синхронізації –: 8 (R1), 10 (R2), 10 (R3). Обчислюється функція за формулою 4.4.

$$F = x \& y \mid x \& z \mid y \& z, \quad (4.4)$$

де & – булево AND, | – булево OR, x, y та z – біти синхронізації R1, R2 і R3 відповідно.

Зсуваються тільки ті регістри, у яких біт синхронізації рівний F. Тобто, фактично, здійснюється зсув регістрів, сінхробіт яких належить більшості.

Вихідний біт системи – результат операції XOR над вихідними бітами регістрів. Структурна схема алгоритму A5/1 зображена на рис. 4.4.

Передача даних здійснюється в структурованому вигляді – з розбиттям на кадри (114 біт). При ініціалізації алгоритму, на його вхід надходять сесійний ключ (64 біта), сформований A8, і номер кадру (22 біта). Далі послідовно виконуються наступні дії:

1. Ініціалізація:

- 64 такти без керування зсувами регістрів, при яких молодші біти складаються з відповідним бітом сесійного ключа;
- аналогічні 22 такти, тільки підсумовування проводиться з номером кадру;
- 100 тактів з керуванням зсувами регістрів, але без генерації послідовності.

2. 228 (114 + 114) тактів робочі, відбувається шифрування переданого кадру (перші 114 біт) і розшифрування (останні 114 біт) того, що приймається.

3. Далі ініціалізація проводиться наново, використовується новий номер кадру.

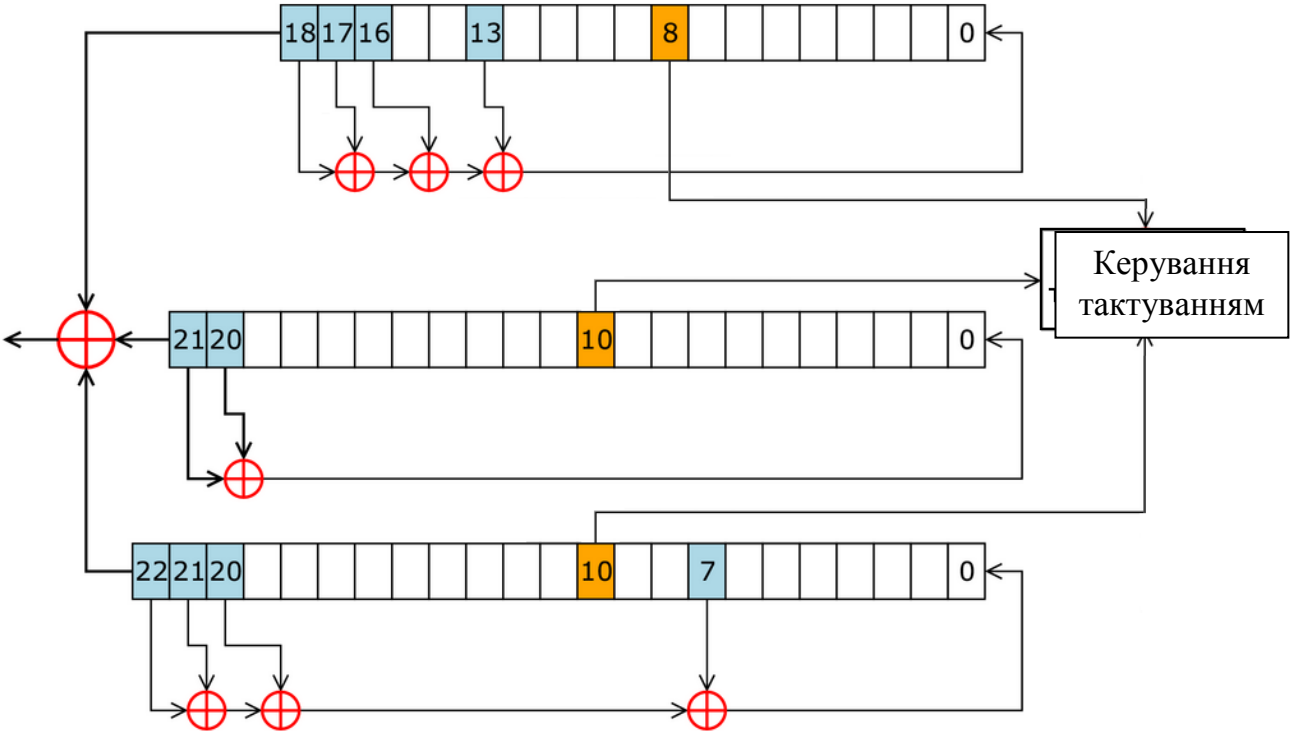


Рисунок 4.4 – Структурна схема алгоритму A5/1

В алгоритм A5/2 доданий ще один регістр на 17 біт (R4), що керує рухом інших. Структурна схема алгоритму A5/2 зображена на рис. 4.5.

Зміни структури такі:

- додано регістр R4 завдовжки 17 біт;
- багаточлен зворотного зв'язку для R4 обчислюється за формулою 4.5;

$$x^{16} + x^{11} + 1, \tag{4.5}$$

- керування тактуванням здійснюється з допомогою R4 таким чином:

1. У R4 біти 3, 7, 10 є бітами синхронізації.

2. Обчислюється мажоритарна функція за формулою 4.4, де x, y і z – біти синхронізації R4(3), R4(7) і R4(10) відповідно.

3. R1 здійснює зсув, якщо R4(10) = F.

4. R2 здійснює зсув, якщо R4(3) = F.

5. R3 здійснює зсув, якщо R4(7) = F;

- вихідний біт системи – результат операції XOR над старшими бітами регістрів і мажоритарних функцій від певних бітів регістрів:

1) для R1 – 12, 14, 15;

2) для R2 – 9, 13, 16;

3) для R3 – 13, 16, 18.

Зміни у функціонуванні не такі істотні і стосуються тільки ініціалізації:

- 64 + 22 такти заповнюються сесійним ключем і номером кадру також R4;
- 1 такт R4(3), R4(7) і R4(10) заповнюються 1;

- 99 тактів з керуванням зрушеннями регістрів, але без генерації послідовності.

Видно, що ініціалізація забирає такий же час (100 тактів без генерації розбиті на дві частини).

Алгоритм A5/3 розроблений в 2001 році і повинен змінити A5/1 у третьому поколінні мобільних систем. Також він називається алгоритмом Касумі. При його створенні за основу взятий шифр MISTY, корпорації Mitsubishi.

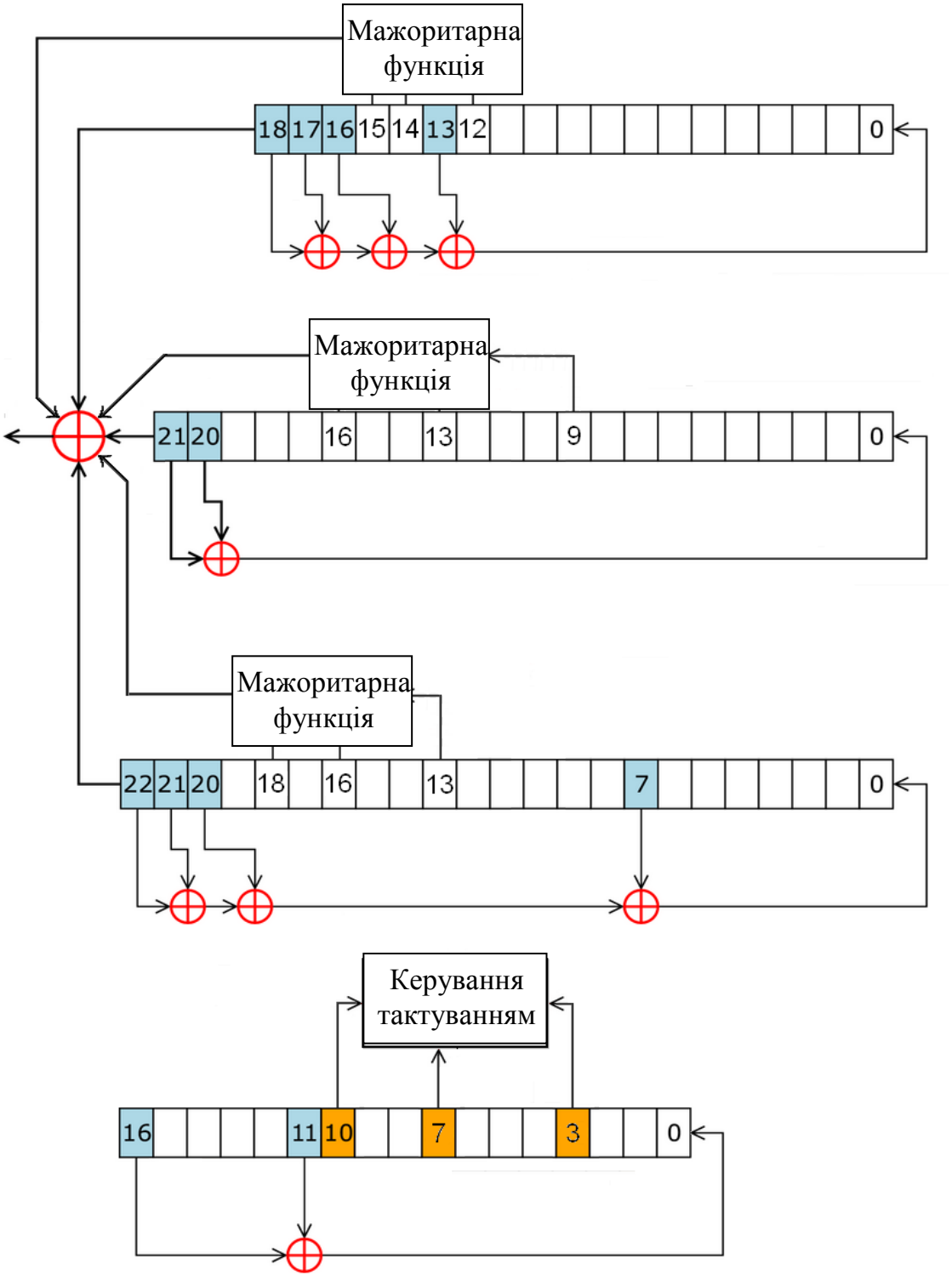


Рисунок 4.5 – Структурна схема алгоритму A5/2

4.4. Порівняльний аналіз криптостійкості алгоритмів A5/1 і A5/2

Перша атака на алгоритм A5/1 полягає в переборі всіх можливих початкових заповнень двох найменших регістрів завдовжки 19 і 22 біти, і відновленні початкового заповнення найбільшого регістру, використовуючи вихідну послідовність A5, – так звану атаку «грубою силою». Але трудомісткість даної атаки становитиме величину не 2(19+22)=241, а близько 245, оскільки послідовність керування синхронізацією залежить також і від найбільшого регістру, що вимагає додаткових обчислень.

У 1997 році Йован Голіч опублікував результати свого криптоаналізу A5/1. Він запропонував спосіб визначення первинного заповнення регістрів за відомим відрізком гамми завдовжки всього 64 біта. Цей відрізок одержують з нульових повідомлень. Атака має середню складність 240.

У грудні 1999 року Аді Шамір і Олексій Бірюков описують в своїй статті нетривіальний і ефективний спосіб злому алгоритму A5/1, публікуючи «Real Time Cryptanalysis of the Alleged A5/1 on a PC». У цій атаці Аді Шамір звертається до слабкості в структурі регістрів зсуву. У методі Аді Шаміра і Бірюкова є два види перевірених практично атак (спочатку проводиться нескладна підготовка даних): першій необхідний вихід алгоритму протягом перших двох хвилин розмови, і ключ обчислюється приблизно за 1 секунду; другій, навпаки, необхідна пара секунд розмови, а ключ обчислюється за декілька хвилин на звичайному ПК.

Алгоритм A5 відповідає всім відомим статистичним тестам. Єдиною його слабкістю є те, що регістри зворотного зв'язку дуже короткі, щоб запобігти пошуку ключа перебором. Варіанти A5 з довшими зсувними регістрами і щільнішими багаточленами зворотного зв'язку повинні бути безпечні.

У 1999 році Вагнеру і Голдбергу без великих зусиль вдалося продемонструвати, що для розкриття алгоритму A5/2 достатньо перебором визначити початкове заповнення четвертого додаткового регістру, що керує рухом інших трьох. Перевірка здійснюється за рахунок нульових кадрів. Складність цієї атаки дорівнює 217, таким чином, на сучасному комп'ютері розкриття шифру забирає декілька секунд.

Таким чином, порівнюючи криптостійкість алгоритмів A5/1 і A5/2, можна зробити висновок, що алгоритм A5/1 є стійкішим, оскільки для його злому у будь-якому випадку необхідно мінімум дві хвилини, а для злому A5/2 – декілька секунд.

Таким чином, можна зробити висновок, що алгоритми забезпечення конфіденційності акустичної інформації, автентифікації, генерації сеансового ключа в стандарті GSM на сьогодні є уразливими до багатьох видів атак. У технологію стільникового зв'язку GSM спочатку була закладена можливість прослуховування розмов спецслужбами. А завдяки роботам провідних світових

криптографів, з'явилися статті, в яких детально описані технології злому алгоритмів шифрування в GSM. Відповідно, можливість прослуховування розмов з'явилася не тільки у спецслужб, а і у осіб або компаній, що володіють набором спеціального устаткування. Тобто забезпечити захист від прослуховування переговорів по мережі GSM можливо тільки за допомогою спеціально розроблених і надійно протестованих крипто-GSM-телефонів (криптофонів), класифікацію яких буде наведено в наступних підрозділах.

Також у стандарті GSM існує ще одна проблема. Оскільки переговори між двома мобільними станціями ведуться через базову станцію і центр комутації, у обслуговуючого персоналу є можливість доступу до ключової інформації. Таким чином, GSM-зв'язок украй не рекомендується використовувати в сферах, де циркулює таємна інформація. Можливо, цю проблему розв’язало б удосконалення існуючого стандарту, пов'язане зі встановленням сеансового ключа безпосередньо між двома мобільними станціями. Проте існує ряд складнощів об'єктивного і суб'єктивного характеру, які можуть ускладнити цей процес. Наприклад, якщо встановлювати сеансовий ключ за допомогою алгоритму Діффі-Хеллмана, то потрібен захищений від модифікації канал, яким відбуватиметься сеанс зв'язку. Інакше у зловмисника з'являється можливість реалізації атаки «людина посередині».

Інший варіант – введення асиметричної криптографії. Наприклад, при продажі SIM-карт кожній карті присвоюється закритий ключ $К_т$, відповідний йому, відкритий ключ, на який видаватиметься сертифікат. Таким чином, за допомогою направленого шифрування алгоритмом RSA абоненти зможуть безпечно передавати один одному сеансовий ключ. При реалізації цього варіанта є дві проблеми. Перша полягає в тому, що асиметрична криптографія значно повільніша, і на встановлення ключа буде потрібний якийсь час. Друга полягає в тому, що спецслужби зацікавлені в прослуховуванні мобільних переговорів, і навіть при оптимальній реалізації цього можливого нововведення навряд чи ці розробки будуть уведені в дію, оскільки в офіційних джерелах асоціація GSM не визнає можливості практичної реалізації атак на алгоритми забезпечення безпеки інформації.

4.5. Забезпечення конфіденційності абонента

Для виключення визначення (ідентифікації) абонента шляхом перехоплення повідомлень, переданих радіоканалом, кожному абоненту системи зв'язку присвоюється «тимчасове посвідчення особи» – тимчасовий міжнародний ідентифікаційний номер користувача (TMSI), який дійсний тільки в межах зони розташування (LA). В іншій зоні розташування йому присвоюється інший TMSI. Якщо абоненту ще не присвоєний тимчасовий номер (наприклад, при першому включенні мобільної станції), ідентифікація

проводиться через міжнародний ідентифікаційний номер (IMSI).

Після закінчення процедури автентифікації і початку режиму шифрування тимчасовий ідентифікаційний номер – TMSI передається на мобільну станцію тільки в зашифрованому вигляді. Цей TMSI використовуватиметься при всіх подальших доступах до системи. Якщо мобільна станція переходить у нову область розташування, то її TMSI повинен передаватися разом з ідентифікаційним номером зони (LAI), в якій TMSI був присвоєний абоненту.

При виконанні процедури корегування місцезнаходження каналами керування здійснюється двосторонній обмін між MS і BTS-службовими повідомленнями, що містять тимчасові номери абонентів TMSI. В цьому випадку в радіоканалі необхідно забезпечити захист інформації про перейменування TMSI і їх належності конкретному абоненту.

Розглянемо, як забезпечується конфіденційність у процедурі корегування місцезнаходження у разі, коли абонент проводить сеанс зв'язку і при цьому здійснює переміщення з однієї зони в іншу.

У цьому випадку мобільна станція вже зареєстрована в режимі переміщення VLR з тимчасовим набором TMSI, відповідним колишній зоні розташування. При вході в нову зону розташування здійснюється процедура розпізнавання, яка проводиться по старому, зашифрованому в радіоканалі TMSI, переданому одночасно з найменуванням зони розташування LAI. LAI дає інформацію центру комутації і центру керування про напрям переміщення мобільної станції і дозволяє запитати колишню зону розташування про статус абонента і його дані, виключивши обмін цими службовими повідомленнями радіоканалами керування. При цьому повідомлення передається каналом зв'язку як зашифрований інформаційний текст з перериванням повідомлення в процесі «естафетної передачі» на 100–150 мс.

Процедура корегування місцезнаходження, що реалізовує функцію конфіденційності абонента, показана на рис. 4.6.

4.6. Перспективні напрями підвищення безпеки акустичної інформації в мережах стандарту GSM

Не дивлячись на те, що розробники мобільних систем стандарту GSM достатньо велику увагу приділяють розробці і впровадженню засобів захисту інформації попит, на нові, сучасні засоби, що забезпечують основні послуги безпеки, не зменшується. Останнім часом з'явилися багато нових, перспективних напрямів у розробці і впровадженні програмних і апаратних засобів захисту інформації. На рис. 4.7 наведено класифікацію сучасних програмно-апаратних засобів захисту інформації стандарту GSM.

4.6.1. Криптофони з додатковим криптопроцесором усередині GSM-телефону

В теперішній час найбільш поширеним криптофоном з додатковим криптопроцесором усередині GSM-телефону є криптосмартфон ANCORT A-7. Цей криптосмартфон спочатку планувався для криптографічного захисту. У телефоні є спеціалізований крипточип, спеціальні фільтри і металевий екран, які запобігають небезпечним випромінюванням. У криптосмартфоні відсутні такі високовипромінювальні елементи, як відеокамера, Bluetooth, інфрачервоний порт, знімна додаткова пам'ять, Wi-Fi. Присутня система контролю правильності роботи шифратора. Реалізація особливої системи синхронізації забезпечує надійну роботу криптосмартфону в роумінгу особливо тоді, коли роумінг доводиться здійснювати на значно віддалені відстані, де при передачі використовуються аналогові засоби передачі даних.

У цьому випадку в криптосмартфоні розроблена унікальна система відновлення криптосинхронізації, що забезпечує високу надійність з'єднання.

Особливості ANCORT A-7:

1. Шифрування SMS і E-mail. У ANCORT A-7 реалізоване повноцінне шифрування текстових повідомлень, що передаються каналами зв'язку протоколу GSM.

2. Виключення можливості атаки «людина посередині». Атакою "людина посередині" називається такий тип атаки, коли зловмисник дістає можливість читати, додавати і змінювати за своїм бажанням повідомлення та іншу інформацію. Причому жоден із абонентів знати про це не буде. Зловмисник повинен мати можливість відстежувати і перехоплювати повідомлення (інформацію) між абонентами. Така атака стає можливою при використанні обміну ключами за алгоритмом Діффі-Хеллмана, якщо обмін ключами відбувається без ідентифікації (перевірки достовірності джерела). Криптосмартфон Анкорт використовує вбудовані алгоритми і унікальну систему ідентифікації того, хто дзвонить, що, у свою чергу, виключає можливість атаки "людина посередині".

3. Захист від вірусів, які дозволяють прослуховувати телефон абонента, за допомогою будь-якого іншого телефону або спеціалізованого комп'ютера. Для захисту від даного типу вірусів при розробці криптосмартфону було розроблено додаткову апаратну частину, яка дозволяє уникнути дії цих вірусів на ефект несанкціонованого прослуховування розмови з інших мобільних телефонів і в безпосередній близькості від нього.

4. Захист від незаявлених можливостей. При розробці криптосмартфону Ancort були проведені необхідні дослідження, внаслідок чого було розроблено спеціальну апаратну частину крптосмартфону, що унеможливлює несанкціоноване прослуховування і видалене включення мікрофона. Цим

криптосмартфон Ancort відрізняється від інших смартфонів відомих виробників.

5. ANCORT A-7 захищений від пристроїв, що дозволяють прослуховувати стандартні GSM–телефони. З метою підвищення захисту на телефон були встановлені спеціальні екрани і фільтри.

6. Захист від прослухування інформації в результаті втрати, розкрадання, отримання тимчасового доступу. Розшифрувати раніше зашифровану інформацію неможливо, навіть якщо телефоном оволодів (втрата, крадіжка, дістали тимчасовий доступ) зловмисник. Тому що для кожного сеансу зв'язку створюється тимчасовий «сеансовий ключ», який надалі неможливо відновити. Це забезпечує збереження розмов, зашифрованих SMS і E-mail, навіть якщо телефон був загублений.

7. Низький час затримки в крипторежимі. Затримка становить близько 0,7 секунди, з яких 0,5 секунди – затримка унаслідок низького пріоритету каналу передачі даних, а 0,2 секунди, що залишилися, забирає процес шифрування. Затримка при звичайній розмові в мережі GSM дорівнює 0,08 секунди. У будь-якому випадку при розмові, коли два співбесідники знаходяться на деякій відстані один від одного, затримка не помітна.

Ключова потужність алгоритму – 1077.

4.6.2. Криптофони з додатковим чипом усередині GSM-телефону

Серед криптофонів з додатковим чипом усередині GSM-телефону можна виділити криптофон TopSecGSM.

TopSecGSM – мобільний телефон із захистом від перехоплення телефонних переговорів. TopSec GSM виготовлений на базі популярного дводіапазонного мобільного телефону S35i компанії Siemens. Проте серцевиною TopSecGSM є криптомодуль, який повністю інтегрований S35i. Шифрована мова передається прозорим каналом передачі даних мережі GSM. Висока безпека досягається комбінацією двох алгоритмів: асиметричного алгоритму з довжиною ключа 1024 біт для ключової угоди і симетричного алгоритму з довжиною ключа 128 біт для шифрування голосу. Режим криптозахисту має опціональне включення. При включеному режимі криптозахисту ініціалізується цифровий виклик і в межах 15 с відбувається обмін ключами. При натисненні клавіші "End" шифровані виклики можуть бути припинені і з'єднання відбудеться за звичайним шляхом. Після закінчення з'єднання ключі, що згенерували раніше, знищуються і в пам'яті не зберігаються.

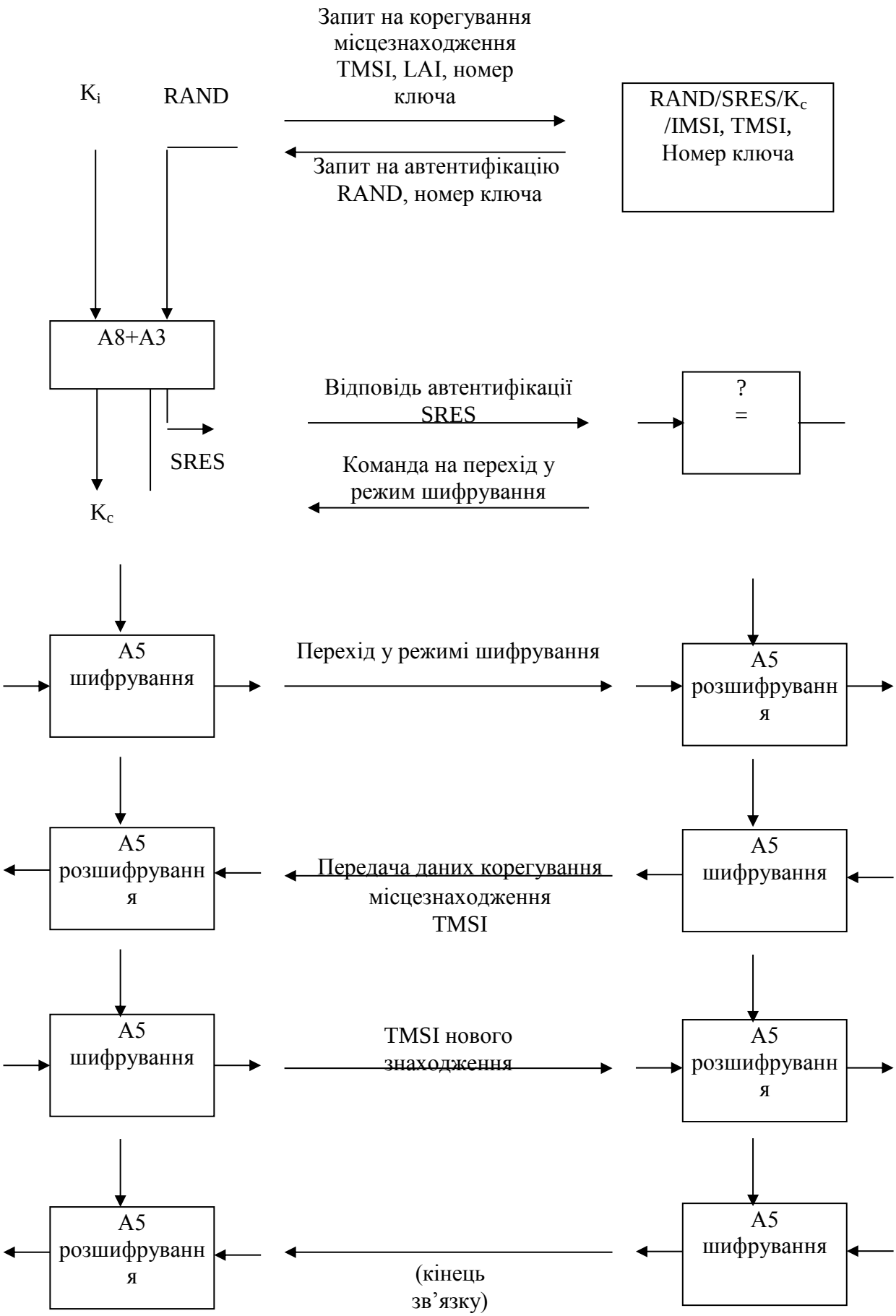


Рисунок 4.6 – Процедура корегування місцезнаходження

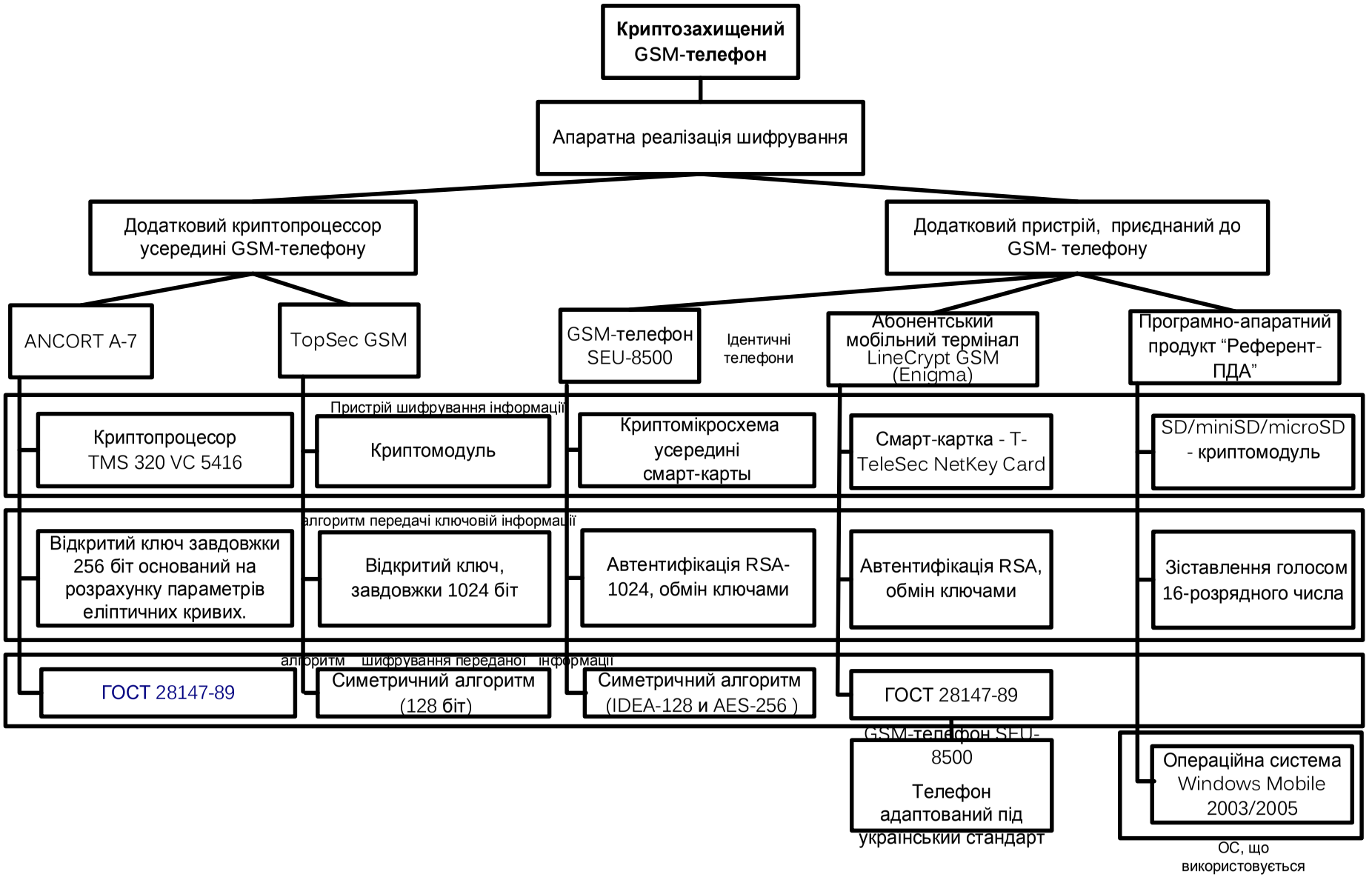


Рисунок 4.7 – Класифікація сучасних програмно-апаратних засобів захисту інформації стандарту GSM

У доповненні до функції криптозахисту TopSec GSM опціонально може забезпечувати автентифікацію. Спеціальне програмне забезпечення TopSec GSM дозволяє створювати закриті групи користувачів. Шифроване з'єднання можливе тільки тоді, коли телефони обох абонентів належать одній і тій же закритій, призначеній для користувача групі.

Для режиму конфіденційного зв'язку використовується канал передачі даних мережі GSM. Швидкість передачі даних становить 9600 біт/с. Передані дані не можуть бути розшифровані в межах мережі GSM.

При конфіденційному зв'язку відбувається стиснення мовного сигналу за допомогою Half-Rate GSM кодера. Стисла інформація доповнюється і кодується кодом з виправленням помилок. Перед передачею до неї додається службова інформація. На першому етапі встановлення з'єднання з використанням криптозахисту мобільні телефони обох абонентів генерують ключі за допомогою алгоритму Діффі-Хеллмана завдовжки 1024 біт. Результат роботи алгоритму використовується для обміну ключовою інформацією, за допомогою якої шифруватимуться дані. Для шифрування мови використовується симетричний алгоритм завдовжки 128 біт. Можлива кількість різних ключів становить 10^{38}.

TopSec GSM забезпечує шифрування голосу на ділянці абонент-абонент у мережах стільникового зв'язку GSM, працюючих у діапазонах частот 900 МГц і 1800 МГц. Конфіденційні переговори можуть бути проведені між двома мобільними телефонами TopSec GSM. Також шифровані виклики можуть бути проведені від TopSec GSM у проводові мережі тільки за умови викликів на ISDN станції абонентам, захищеним пристроєм TopSec 703+ з сімейства продуктів TopSec. Можливе також забезпечення зв'язку з кожним абонентом звичайним нешифрованим шляхом.

4.6.3. GSM-телефон SEU-8500

Шифрування здійснюється в режимі реального часу за допомогою алгоритмів IDEA з довжиною ключа 128 біт і AES з довжиною ключа 256 біт. Передача ключів шифрування здійснюється протягом сеансу установки з'єднання, як частина процедури автентифікації передавальної сторони.

Генерація ключів шифрування здійснюється за допомогою сертифікованого генератора випадкових чисел у мікросхемі смарт-карти.

Секретний ключ знаходиться в смарт-карті, і прорахувати його неможливо.

4.6.4. LineCrypt GSM (Enigma)

Enigma дозволяє встановити захищений голосовий виклик у мережі стільникового зв'язку загального користування із здійсненням безпосереднього шифрування між двома абонентами, що ведуть переговори. Щоб досягти цього, голосовий сигнал перш ніж буде переданий мережею, переводиться в цифрову форму і шифрується. В процесі встановлення зв'язку використовується ключ для шифрування і розшифрування даних.

У терміналі використовується європейський криптографічний алгоритм IDEA, при якому ключ є індивідуальним для кожної розмови і генерується під час процесу обміну на початку кожного захищеного виклику. В кінці виклику ці ключі знищуються, і при наступному виклику цей процес виконується наново. Крім того, для забезпечення додаткового захисту інформації, кожен термінал Enigma має індивідуальний цифровий сертифікат. Перед початком виклику відбувається обмін сертифікатами обох сторін і їх підтвердження з використанням алгоритму RSA. Процес шифрування відбувається в додатковій смарт-карті – T-TeleSec NetKey Card.

Під час спільної роботи державного підприємства «Українські спеціальні системи», Департаменту спеціальних телекомунікаційних систем і захисту інформації СБ України і компанії T-System International GmbH (Німеччина), яка тривала декілька років, Enigma була адаптована до вимог українського ринку і нормативно - правової базі України. В процесі адаптації термінал був перевірений на відсутність «закладок», було розроблене меню українською мовоюі, а також забезпечена сумісність з сертифікованим криптографічним алгоритмом ГОСТ 28147-89.

4.6.5. Додатковий пристрій, що виконує функцію шифрування, та приєднується до звичайного GSM-телефону

До додаткових пристроїв, що виконують функцію шифрування, слід віднести програмно апаратний комплект “Референт-PDA”.

Програмно-апаратний продукт “Референт-PDA” розроблений для пристроїв типу смартфон, що працюють під керуванням операційної системи Windows Mobile 2003/2005. Референт-PDA дозволяє запобігти прослуховуванню переговорів, що ведуться між двома смартфонами. Комплект складається з SD-модуля і програмного забезпечення.

Основною відмінністю виробу від аналогів є використання низькошвидкісного каналу передачі даних (до 1600 бод), що дозволяє

працювати при слабкому GSM–сигналі (у місцях поганого прийому), в роумінгу, при використанні різних операторів і т. ін.

Запуск програми здійснюється автоматично при підключенні SD-модуля “Референт-PDA”, при цьому на екрані смартофону в правому нижньому кутку з'являється значок індикації запуску програми у фоновий режим. Для здійснення і ухвалення викликів використовується програма “Референт-PDA”, яка з'явиться в смартфоні. Під час надходження дзвінка від іншого комплекту “Референт-PDA” замість програми «телефон» автоматично відкривається інтерфейс програми “Референт-PDA”.

У процесі встановлення з'єднання проводиться обмін спеціальною інформацією для взаємної автентифікації пристроїв і формування сеансового ключа. Даний обмін супроводжується статусним написом «Key exchange». Після закінчення обміну виводиться напис «Call established xxxx», де xxxx – шістнадцяткове число, що є результатом обчислення хеш-функції над сеансовою ключовою інформацією. Число xxxx повинне бути однаковим в обох абонентів. Абоненти повинні переконатися в цьому шляхом повідомлення голосом протилежному абоненту свого числа. Якщо числа вийшли різними, в цілях безпеки необхідно перервати поточний сеанс зв'язку і встановити з'єднання наново.

Смартофони, з якими комплект перевірений на сумісність: QTEK s110, I-MATE JAM, QTEK 2020i, I-MATE PDA 2K, QTEK s200, I-MATE JAMIN, I-MATE PDA 2K.

Смартфони, з якими даний комплект не працює: ASUS p505.

Порівнюючи вищеперелічені криптофони, можна зробити висновок про те, що ANCORT A-7 використовує найбільш сучасні засоби криптографічного захисту інформації. Внаслідок цього в криптофоні ANCORT A-7 забезпечується найбільша криптостійкість до атак. Також у ньому, на відміну від інших криптофонів, забезпечується шифрування SMS і E-mail.

4.7. Особливості захисту інформації в системах мобільного зв'язку стандарту IS-95

Стандарт IS-95 забезпечує високий ступінь безпеки переданих повідомлень і даних про абонентів. Перш за все він має складніший, ніж GSM, радіоінтерфейс, що забезпечує передачу повідомлень кадрами з використанням канального кодування і перемежування з подальшим «розширенням» переданих сигналів за допомогою складених широкосмугових сигналів, сформованих на основі 64 видів послідовностей Уолша і псевдовипадкових послідовностей з кількістю елементів 2^{15} і (2^{42}-1).

Безпека зв'язку забезпечується також застосуванням процедур автентифікації і шифрування повідомлень.

У стандарті IS-95 реалізовано процедуру автентифікації, відповідну стандарту IS-54B. Шифрування повідомлень, переданих каналами зв'язку, здійснюється також з використанням процедур стандарту IS-54B.

У мобільній станції зберігається один ключ А і один набір загальних секретних даних. Мобільна станція може передавати «цифровий підпис» для автентифікації, що складається з 18 біт. Ця інформація передається на початку повідомлення (відповідає мобільній станції на запит мережі при пошуку станції), додається до реєстраційного повідомлення або пакета даних, переданих по каналу доступу. В мобільних станціях передбачено можливість оновлення загальних секретних даних.

У стандарті IS-95 реалізовано режим «приватний характер зв'язку», що забезпечується використанням секретної маски у вигляді довгого коду. Цей процес також аналогічний процесу формування маски у вигляді довгого коду, описаного в стандарті IS-54B.

4.7.1. Особливості захисту інформації в прямому каналі зв'язку

Прямий канал зв'язку системи cdmaOne включає шістдесят чотири робочих канали (рис. 4.8): один пілотний канал (pilot channel), один канал синхронізації (synchronization clumnel) і шістдесят два канали передачі даних, з яких до семи каналів можуть бути використані як канали персонального виклику (paging clumnel), а інші (від 55 до 62 каналів) – для передачі прямого трафіку (forward traffic channel).

Для розділення каналів служать кодові бінарні фазоманіпулюванні (БФМ) послідовності, сформовані на базі ансамблю ортогональних функцій Уолша за таким алгоритмом:

$$H_1 = 0; \quad H_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}; \quad H_4 = \begin{pmatrix} 01 & 01 \\ 00 & 11 \\ 01 & 10 \end{pmatrix}; \ldots; \quad H_{2N} = \begin{pmatrix} H_N & H_N \\ H_N & H_N \end{pmatrix}, \quad N = 1,2,4,8,16,32.$$

Адресна послідовність W_o (константа) призначена для організації пілотного каналу, а адресна послідовність W_{32} (меандр) – для організації каналу синхронізації. Канали персонального виклику і прямого трафіку використовують інші шістдесят дві адресних послідовності. Зверніть увагу: у прямому каналі зв'язку функції Уолша використовуються для кодового розділення робочих каналів (у зворотному каналі зв'язку їх функціональне призначення принципово інше).

Як показано на рис. 4.9, в передавальному тракті BS перенесення сигналів у каналах на адресні несучі здійснюється в два етапи.

На першому етапі послідовністю інформаційних символів модулюють (операція "сума за модулем 2") адресну БФМ-послідовність, сформовану на основі функцій Уолша (W_j) з тактовою частотою 1,2288 Мбіт/с. На другому етапі одержаним потоком скремблюють ("сума за модулем 2") псевдовипадкові БФМ-послідовності PN_I і PN_Q (PN – pseudonoise) в синфазному (I) і квадратурному (Q) каналах. Ці псевдовипадкові послідовності (ПВП) не збігаються між собою і є М-послідовностями. Вони однакові для всіх 64 каналів і мають тактову частоту 1,2288 Мбіт/с.

У пілотному каналі використовується функція Уолша W_o, що визначає формування послідовності з 64 нулів. Тому результуюча адресна послідовність у пілотному каналі фактично визначається псевдовипадковими послідовностями PN_I і PN_Q квадратурних каналів I і Q, М-послідовностями довжиною в два символи (короткий код), сформованими на основі поліномів п'ятнадцятого степеня:

$$PNI(x)=x_{15}+x_{13}+x_9+x_8+x_7+x_5+1,$$
$$PNQ(x)= x_{15} + x_{12} + x_{11} + x_{10}+x_б + x_5 + x_4 + x_3+1.$$

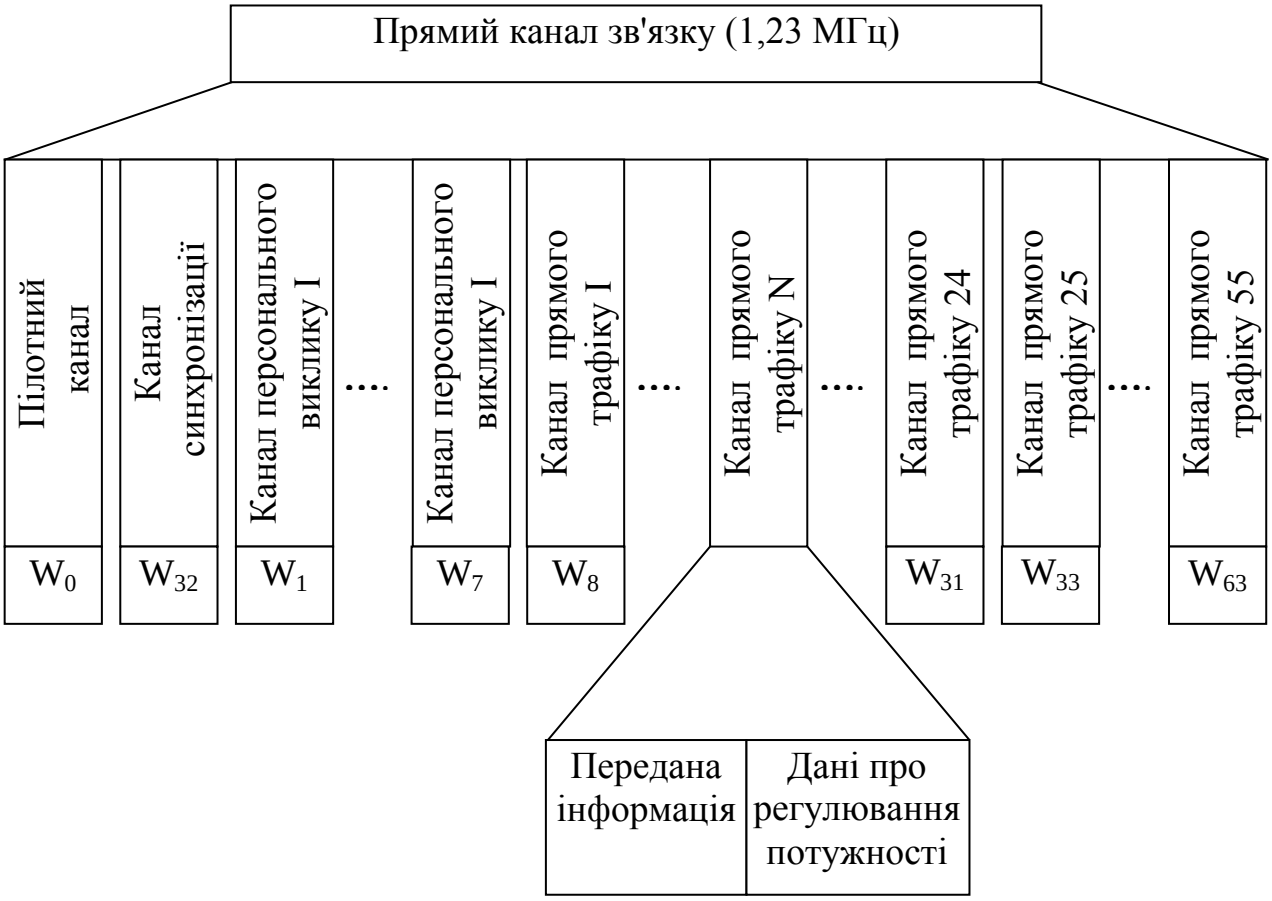


Рисунок 4.8 – Структура прямого каналу зв'язку системи cdmaOne

У канал синхронізації дані надходять із швидкістю 1200 біт/с. Після згортального кодування (9, 1/2) їх швидкість зростає в 2 рази – до 2400 бит/с. Потім інформаційна послідовність поступає на пристрій повторення, на виході якого формується інформаційний потік із швидкістю 4800 біт/с.

Далі йде процедура блокового перемежування в межах кадрів тривалістю 20 мс. Перемежування застосовується для перетворення пакетів помилок, що виникають при передачі в каналі зв'язку, в одиночні помилки. Це дозволяє істотно понизити імовірність помилки при декодуванні інформації. На виході блокового перемежувача швидкість інформаційного потоку не змінюється – 4800 біт/с.

Після перемежування інформаційна послідовність надходить на модулятор послідовностей Уолша. На вході модулятора кожен символ інформаційної послідовності має тривалість, рівну чотирьом періодам (по 64 біт) послідовності Уолша:

$$1{,}2288 \text{ Мбіт/с} = 64\text{x}4\text{x}4800 \text{ біт/с}.$$

На виході модулятора тактова частота інформаційного потоку 1,2288 Мбіт/с. У каналі синхронізації використовується функція Уолша W_{32}, що визначає формування послідовності типу «меандр».

У каналах прямого трафіку для передачі мови передбачено використання вокодеру CELP із змінною (залежно від параметрів мови абонента) швидкістю перетворення: 8550, 4000, 2000 або 800 біт/с. Інформація в каналах трафіку передається кадрами тривалістю 20 мс. При цьому швидкість передачі кодованої мовної інформації, що надходить у канал, постійна протягом кадру і становить 9600, 4800, 2400 або 1200 біт/с. Згортальний кодер з довжиною кодового обмеження 9 і швидкістю 1/2 подвоює швидкість інформаційного потоку на виході: 19200, 9600, 4800 або 2400 біт/с відповідно. Для вирівнювання швидкості потоків кодованої мовної інформації до швидкості 19200 біт/с застосовується пристрій повторення:

$$19200\text{x}1 = 19200 \text{ біт/с},$$
$$9600\text{x}2 = 19200 \text{ біт/с},$$
$$4800\text{x}4 = 19200 \text{ біт/с},$$
$$2400\text{ x}8 = 19200 \text{ біт/с}.$$

Чим більша кратність повторення символів, тим менша потужність використовується для їх передачі по каналу зв'язку. Це дозволяє понизити рівень взаємних перешкод у системі і збільшити пропускну спроможність мережі.

На MS при прийомі невідомі швидкість передачі інформації і кратність повторення символів в поточному кадрі. Тому декодер MS здійснює 4 варіанти

декодування сигналів, що приймаються, з різними поєднаннями швидкості передачі і кратності повторення. Дійсна швидкість передачі визначається за мінімумом виявлених помилок.

Довгий код, що формується на основі лінійного полінома 42 степеня:

$$P(x) = x_{42} + x_{35} + x_{33} + x_{31} + x_{27} + x_{26} + x_{25} + x_{22} + x_{21} + x_{19} + x_{18} + \\ + x_{17} + x_{16} + x_{10} + x_7 + x_6 + + x_5 + x_3 + x_2 + x + 1,$$

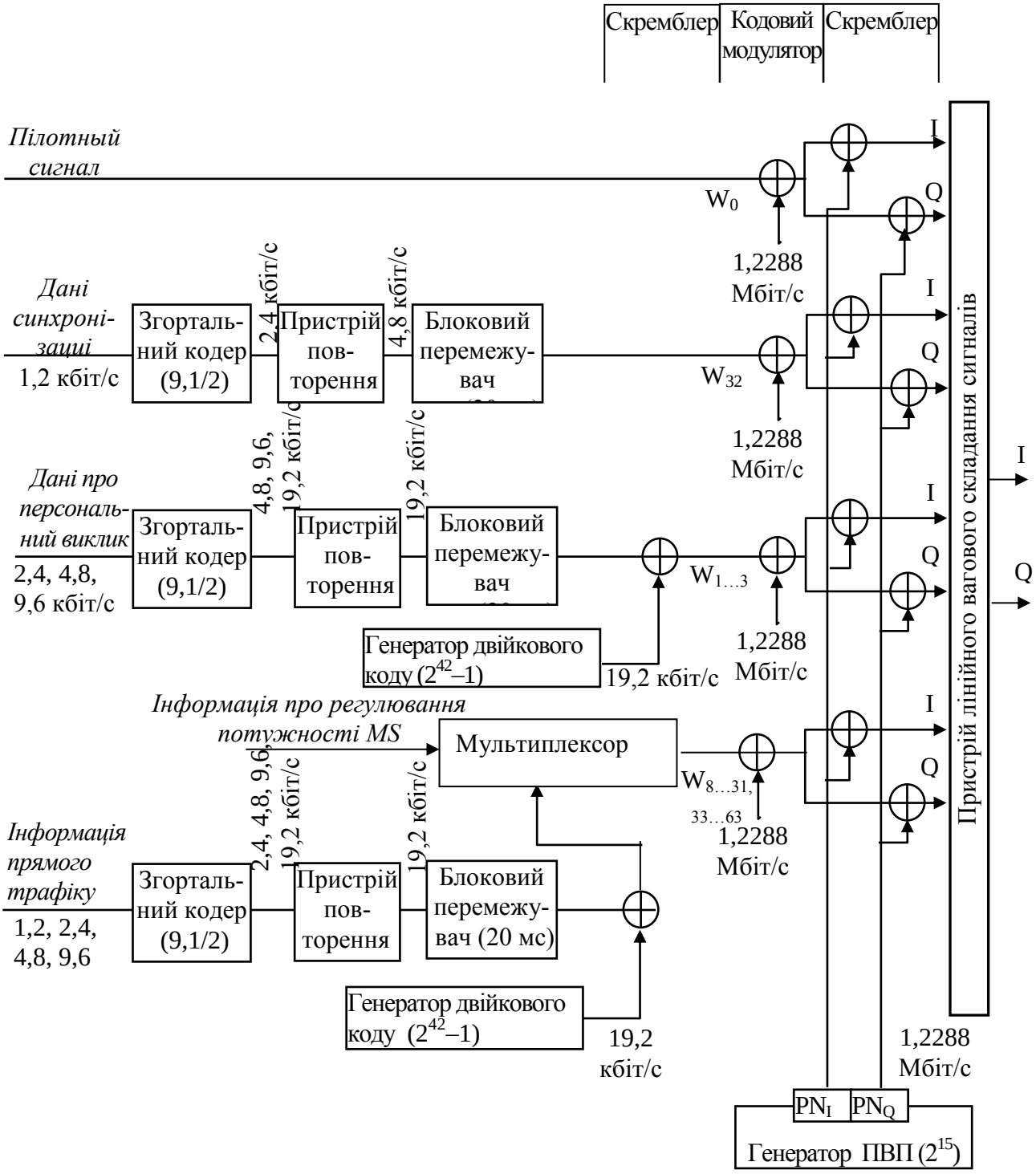


Рисунок 4.9 – Структурна схема передавального тракту базової станції

що є М-послідовністю завдовжки 2^{42}-1=14,4x10^{12} біт, несе інформацію про індивідуальний номер абонента в мережі. Необхідні для генерування довгого коду дані – маска (початкове заповнення генератора) записані в постійному запам'ятовувальному пристрої (ПЗП) MS. Вони містять 42 біт і включають фіксовану для всіх каналів прямого трафіку преамбулу з 10 біт і унікальний ідентифікатор обслуговуваної MS – 32-бітовий серійний номер (рис. 4.10).

Преамбула	Серійний номер		
1100011000	Кодовий номер 8 біт	Резервні біти 6 біт	Порядковий номер 18 біт

Рисунок 4.10 – Структура маски довгого коду каналу прямого трафіку

Генерування довгого коду здійснюється з тактовою частотою 1,2288 Мбіт/с, після чого пристрій децимації (на рис. 4.9 не показаний) знижує тактову частоту до 19200 біт/с, залишаючи кожен 64 символ початкової послідовності. Сформований таким чином довгий код надходить на один з входів скремблера. На другий вхід поступає інформаційна послідовність з виходу блокового перемежувача (також із швидкістю 19200 біт/с). Скремблер здійснює додавання за модулем 2 вхідних потоків. Скремблювання інформації довгим кодом є могутнім криптографічним засобом, що забезпечує високий ступінь конфіденційності переданих повідомлень.

Скремблювані дані мультиплексують з інформацією про регулювання потужності передавачів MS. Для цього окремі символи потоку даних, що надходять на вхід мультиплексора, замінюють символами команд регулювання потужності.

Після мультиплексора потік даних із швидкістю 19200 біт/с надходить на вхід кодового модулятора. Номер функції, що використовується в адресній послідовності Уолша, однозначно визначає номер каналу трафіку BS. З виходу модулятора складний сигнал з тактовою частотою 1,2288 Мбіт/с прямує в квадратурні канали I і Q, де здійснюється його скремблювання єдиними для всіх 64 каналів псевдовипадковими послідовностями PN_I, і PN_Q.

Канали персонального виклику служать для передачі MS системної інформації і команд керування. Структура каналу персонального виклику повторює структуру каналу прямого трафіку. Відмінність полягає в тому, що дані в канали виклику надходять із швидкістю 9600, 4800 або 2400 біт/с, інформація в них не мультиплексується з командами регулювання потужності і застосовується інша маска довгого коду (рис.4.11).

Преамбула	Резервні біти	Номер каналу персонального виклику	Резервні біти	Індекс часового зсуву ПВП пілот-сигналу
1100011000	00000	XXX	000000000000	XXXXXXXXX

Рисунок 4.11 – Структура маски довгого коду каналу персонального виклику

Сформовані в прямому каналі зв'язку квадратурні складові сигналів усіх 64 робочих каналів об'єднуються (підсумовуються з вагами) в режимі лінійного додавання. Сформований груповий сигнал, що має синфазний і квадратурний компоненти, фільтрується в основній смузі частот (1,25 Мгц) і надходить на схему чотирипозиційної фазової маніпуляції ФМ–4 {quaternary phase shift keying – QPSK), де здійснюється його перенесення на проміжну частоту. Далі груповий сигнал переноситься з проміжної частоти на несучу частоту, посилюється лінійним підсилювачем потужності і випромінюється передавальною антеною BS.

Існують **особливості використання пілотного каналу.** Пілотний сигнал, що безперервно випромінюється BS, забезпечує виконання декількох функцій. По-перше, рівень потужності сигналу, що випромінюється в пілотному каналі, постійний і на 4...6 дБ вищий, ніж у каналах трафіку. MS використовує пілотний сигнал для виділення опорного коливання, необхідного для когерентної обробки сигналів BS при прийомі. По-друге, результати вимірювання потужності пілотних сигналів використовуються MS при естафетній передачі і при регулюванні потужності передавача BS. По-третє, пілотний сигнал містить інформацію про єдиний час, що одержується BS від супутникової радіонавігаційної системи NAVSTAR/GPS (Global Positioning System) кожну парну секунду.

Адресна послідовність пілотного каналу (короткий код) є періодичними послідовностями PN_I і PN_Q, сформованими на основі поліномів п’ятнадцятого степеня:

$$PN_I(x) = x_{l5} + x_{13} + x_9 + x_8 + x_7 + x_5 + 1,$$
$$PN_Q(x) = x_{l5} + x_{12} + x_{11} + x_{10} + x_6 + x_5 + x_4 + x_3 + 1,$$

кожна з яких має довжину 2^{15}=32768 символів і період повторення 215 біт/1,2288 Мбіт/с=26,66 мс.

Всі BS у системі використовують один короткий код, але з різними циклічними зсувами. Сигнали, що випромінюються BS у різних стільниках і секторах, розрізняються за циклічному зсувом короткого коду. Циклічні зсуви мають рівномірний крок 26 (64) символи. Тобто, можливі 511 різних циклічних

зсувів короткого коду щодо положення з умовно нульовим зсувом. Це означає, що навіть у районах з мікростільниковою структурою існує тверда гарантія того, що сигнали різних BS будуть розпізнані при прийомі. Навіть якщо мережа містить більше 511 BS, то нескладно домогтися того, щоб BS з однаковими циклічними зсувами короткого коду не опинилися одночасно в зоні радіоприйому однієї MS.

Для забезпечення високої точності циклічних зсувів у стандарті cdmaOne реалізовано концепцію синхронізованих BS. Єдиний час у системі і висока стабільність тактових частот підтримується за допомогою супутникової системи радіонавігації NAVSTAR/GPS.

Приймач MS має в своєму складі три паралельних основних канали кореляційної обробки сигналів і один допоміжний скануючий канал. На першому етапі роботи, етапі пошуку сигналів BS, MS використовує допоміжний скануючий канал. MS після захоплення несучої частоти обробляє пілотний сигнал BS і виділяє з багатопроменевого сигналу, що приймається, три найбільш потужні компоненти. Подальша обробка сигналів трьох вибраних променів у гілках кореляційного приймача дозволяє MS відстежувати сигнали BS в умовах адитивних і мультиплікативних перешкод і оцінювати із заданою точністю їх амплітуди, фази і часові затримки.

Таке застосування триканального приймача і пілотного сигналу робить можливим когерентний прийом сигналів BS з триразовим часовим розсіянням і подальшим когерентним об'єднанням гілок. Це забезпечує істотний енергетичний виграш при прийомі і, як наслідок, високу перешкодостійкість системи.

Процедура синхронізації MS з BS фактично є процедурою доступу MS до ресурсів мережі через дану BS.

MS синхронізується з BS за коротким кодом. Для цього MS вимірює часові затримки сигналів у променях, що виділяються, і підстроює в кореляторах циклічні зсуви опорних ПВП. Далі MS починає сканувати канал синхронізації, що використовує той же короткий код, з тим же циклічним зсувом, що і пілотний канал.

Канали синхронізації всіх BS використовують одну функцію Уолша W_{32}, що забезпечує формування меандру. Швидкість передачі даних по каналу синхронізації складає 1200 біт/с, а довжина кадру дорінює періоду повторення короткого коду (26,66 мс). Оскільки канал синхронізації жорстко зв'язаний по тактовій частоті і по зсуву циклічного коду з пілотним каналом, MS дістає доступ до синхроінформації тієї BS, на пілотний канал якої вона настроїлася.

Повідомлення каналу синхронізації (Sync. Channel Massage) містить:

- дані про точний час у системі;
- циклічний зсув короткого коду даної BS;

- інформацію ідентифікації BS і MSC;
- потужність сигналу в пілотному каналі;
- параметри довгого коду;
- швидкість передачі даних у каналі персонального виклику.

Приймаючи повідомлення каналу синхронізації, MS одержує необхідну інформацію для початкової синхронізації з мережею, тобто для доступу в мережу.

Після закінчення процедури початкової синхронізації MS настроюється на канал персонального виклику або за командою з BS, або в результаті перебору наявних каналів (до семи каналів у смузі 1,25 Мгц). Тим самим вона дістає доступ до системної інформації і починає приймати команди керування. Якщо команди керування від BS не надходять, то MS переходить у режим очікування, продовжуючи прослуховувати канал персонального виклику і підтримуючи готовність до встановлення з'єднання.

Швидкість передачі інформації в каналі персонального виклику становить 9600, 4800 або 2400 біт/с. Маска довгого коду залежить від номера каналу персонального виклику і циклічного зсуву ПВП у пілотному каналі.

У каналах персонального виклику використовуються повідомлення чотирьох типів:

- заголовок (Overhead Message);
- пейджінг (Paging Message);
- ордер (Order Message);
- повідомлення про призначення каналів (Channel Assigment Message).

Розглянемо їх детальніше. Система мобільного зв'язку cdmaOne адаптивна. Її конфігурація вибирається з урахуванням конкретних умов розгортання мережі. Інформація про конфігурацію системи доводиться до абонентів за допомогою чотирьох типів повідомлень заголовка:

- повідомлення про параметри системи {System Parametr Message);
- повідомлення про параметри доступу {Access Parametr г Message);
- граничного списку (Neighbour List Message);
- списку каналів CDMA (CDMA Channel List Message).

Повідомлення про параметри системи містять інформацію про конфігурацію каналу персонального виклику, параметри реєстрації, допоміжні параметри при пошуку пілотного сигналу і т.і.

Повідомлення про параметри доступу містить відомості про конфігурацію каналу доступу MS і деякі параметри керування.

Граничний список містить дані, що дозволяють прискорити процес циклічного зсуву короткого коду в пілотному каналі і інші параметри BS сусідніх сотів.

Список каналів CDMA надає MS інформацію про знаходження каналів персонального виклику

Пейджінг є повідомленнями (сторінки), адресованими одній або декільком MS. Ці повідомлення зазвичай передають ті BS, які знаходяться в зоні пошуку MS при вхідному виклику в мережу. При швидкості передачі 9600 біт/с один канал персонального виклику забезпечує передачу до 180 сторінок протягом однієї секунди. Відповідно по семи каналах може бути передано до 1260 сторінок за секунду.

Ордер охоплює широкий клас повідомлень керування конкретними MS, використовуваних для підтвердження реєстрації MS, для блокування MS в стані збою і т. ін.

Повідомлення про призначення каналів надають MS інформацію про канал трафіку, що виділяється їй, про зміну каналу персонального виклику або про команди перемикання MS в аналогову систему мобільного зв'язку.

Інформація в каналі персонального виклику передається або загальним потоком, або в режимі часового розділення {time division multiplexing – TDM), коли повідомлення, адресовані конкретною MS, передаються в певних часових інтервалах – слотах. Період повторення слотів, виділених для однієї MS, становить від 2 до 128 с. Інформацію про виділений слот MS одержує при реєстрації на BS. Робота в TDM-форматі дозволяє MS сканувати тільки свої слоти, відключаючись в перервах між ними. Це забезпечує істотну економію джерела живлення MS у стані очікування.

Канали прямого трафіку призначені для передачі повідомлень трафіку (мова і дані абонентів) і службової інформації (сигналізація) з BS на MS.

Кодована мовна інформація надходить у канали трафіку кадрами по 20 мс. При цьому швидкість передачі постійна протягом кадру і становить 9600, 4800, 2400 або 1200 біт/с залежно від параметрів мови абонента. У паузах мови швидкість інформаційного потоку автоматично знижується до мінімального значення. Передача трафіку з адаптивною швидкістю мінімізує рівень внутрішньосистемних перешкод і підвищує пропускну спроможність мережі.

Службова інформація, що передана по каналу прямого трафіку, може бути чотирьох типів:

- повідомлення керування викликом;
- повідомлення керування естафетною передачею;
- команди регулювання потужності;
- інформація забезпечення безпеки зв'язку і автентифікації абонентів.

Стандартом передбачені два режими передачі службової інформації (сигналізації). При першому режимі (blank-and-burst) службові повідомлення передаються із швидкістю 9600 біт/с. При цьому кадри системної (службової) інформації заміщають кадри трафіку. Другий режим (dim-and-burst) забезпечує

передачу трафіку і службової інформації в одному кадрі: перетворення мови у вокодері при цьому здійснюють не швидше ніж 4000 біт/с (4800 біт/с в каналі), а ресурс, що залишився, використовується для сигналізації. Результуючий кадр, таким чином, складається з двох частин – трафіку і службової інформації, а швидкість передачі в ньому становить 9600 біт/с. Ускладнення структури кадру в режимі dim-and-burst

4.7.2. Особливості захисту інформації в зворотному каналі зв'язку

Зворотний канал зв'язку системи cdmaOne становлять канали доступу (access channel) (до 32) і канали зворотного трафіку (reverse traffic channel) (до 64) (див. рис. 4.12).

Канал доступу використовується спільно з каналом персонального виклику для реєстрації MS у мережі і виконання початкових процедур зі встановлення з'єднання, тобто до надання MS каналу зворотного трафіку. Швидкість передачі інформації в каналі доступу 4800 біт/с. Структурна схема передавального тракту каналу доступу наведена на рис. 4.1. Як видно з рисунку, алгоритми обробки інформації в каналі доступу і в каналі зворотного трафіку аналогічні.

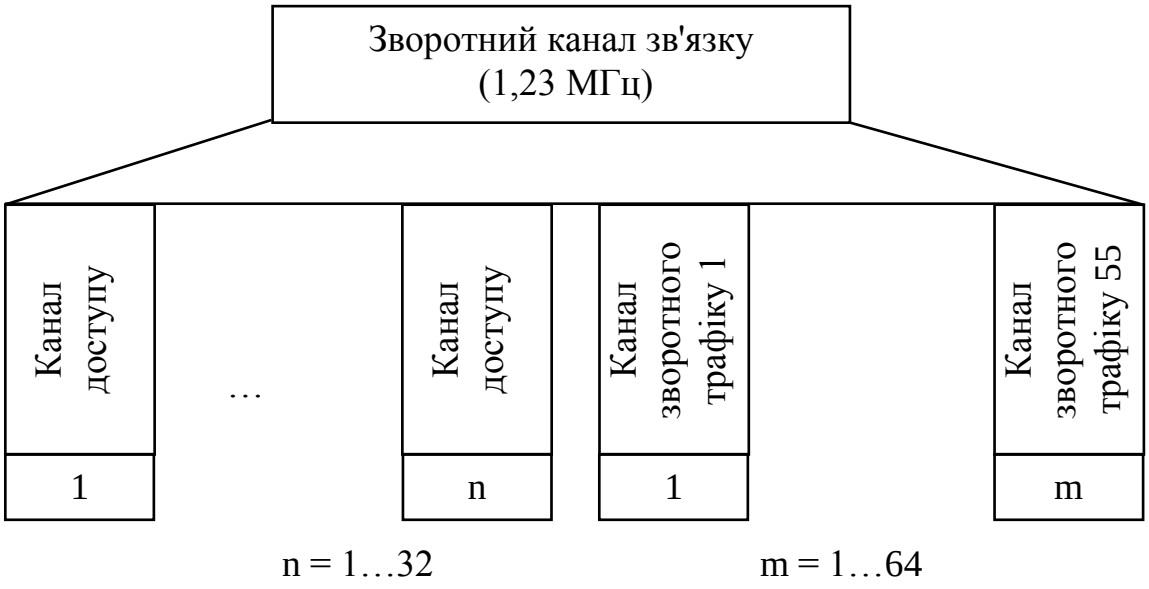


Рисунок 4.12 – Структура зворотного каналу зв'язку системи cdmaOne

Канал зворотного трафіку призначений для передачі мовних повідомлень і службової інформації з MS на BS. Мова абонента, перетворена CELP-вокодером, із швидкістю 9600, 4800, 2400 або 1200 біт/с надходить у тракт каналу зворотного трафіку на згортальний кодер, що має довжину кодового обмеження 9 і швидкість 1/3. В процесі кодування швидкість потоку даних

зростає (потроюється) до 28800, 14400, 7200 або 3600 біт/с відповідно. Потім, як і в прямому каналі, в пристрої повторення швидкість потоку доводиться до постійної величини, яка дорівнює 28800 біт/с:

дозволяє підвищити якість передачі мови в порівнянні з режимом blank-and-burst.

28800x1=28800 біт/с,
14400x2 = 28800 біт/с,
7200x4 = 28800 біт/с,
3600x8 = 28800 біт/с.

Після блокового перемежування інформації в межах кадрів тривалістю 20 мс потік із швидкістю 28800 біт/с надходить на кодер Ріда–Соломона 1 роду. Процес кодування здійснюється в два етапи. Спочатку вхідний потік двійкових символів розбивається на пакети по 6 символів. При цьому кожен пакет однозначно визначає число від 0 до 63 в десятковій системі числення. Після чого кожному пакету ставлять у відповідність послідовність, що має період 64 біт і сформовану з використанням однієї з 64 функцій Уолша:

$000000_2 = 0_{10} \rightarrow W_0$;
$000001_2 = 0_{11} \rightarrow W_1$;
……..
$101010_2 = 42_{10} \rightarrow W_{42}$;
$111111_2 = 63_{10} \rightarrow W_{63}$.

Після кодування швидкість інформаційного потоку на виході кодеру становить:

28800 біт/с • 64біт / 6біт = 307200 біт/с.

Важливо відзначити, що в зворотному каналі зв'язку послідовності, що сформовані з використанням функцій Уолша, застосовуються для перешкодостійкого кодування інформації, тоді як в прямому каналі вони застосовуються для розширення спектру частот і формування адресних послідовностей.

Після кодування дані надходять на кодовий модулятор і скремблер, де здійснюється розширення бази сигналу і формування його «адресних ознак».

У кодовому модуляторі застосовується довгий код ($L = 2^{42}-1$). Всі MS мережі в каналах зворотного трафіку використовують один і той же довгий код, але з різними значеннями циклічного зсуву. За величиною зсуву BS розрізняють сигнали обслуговуваних MS. Маски довгого коду в каналах прямого і зворотного трафіку збігаються. При цьому в каналі зворотного трафіку не застосовується дециматор, тому структура кодової послідовності зберігається і тактова частота залишається незмінною – 1,2288 Мбіт/с.

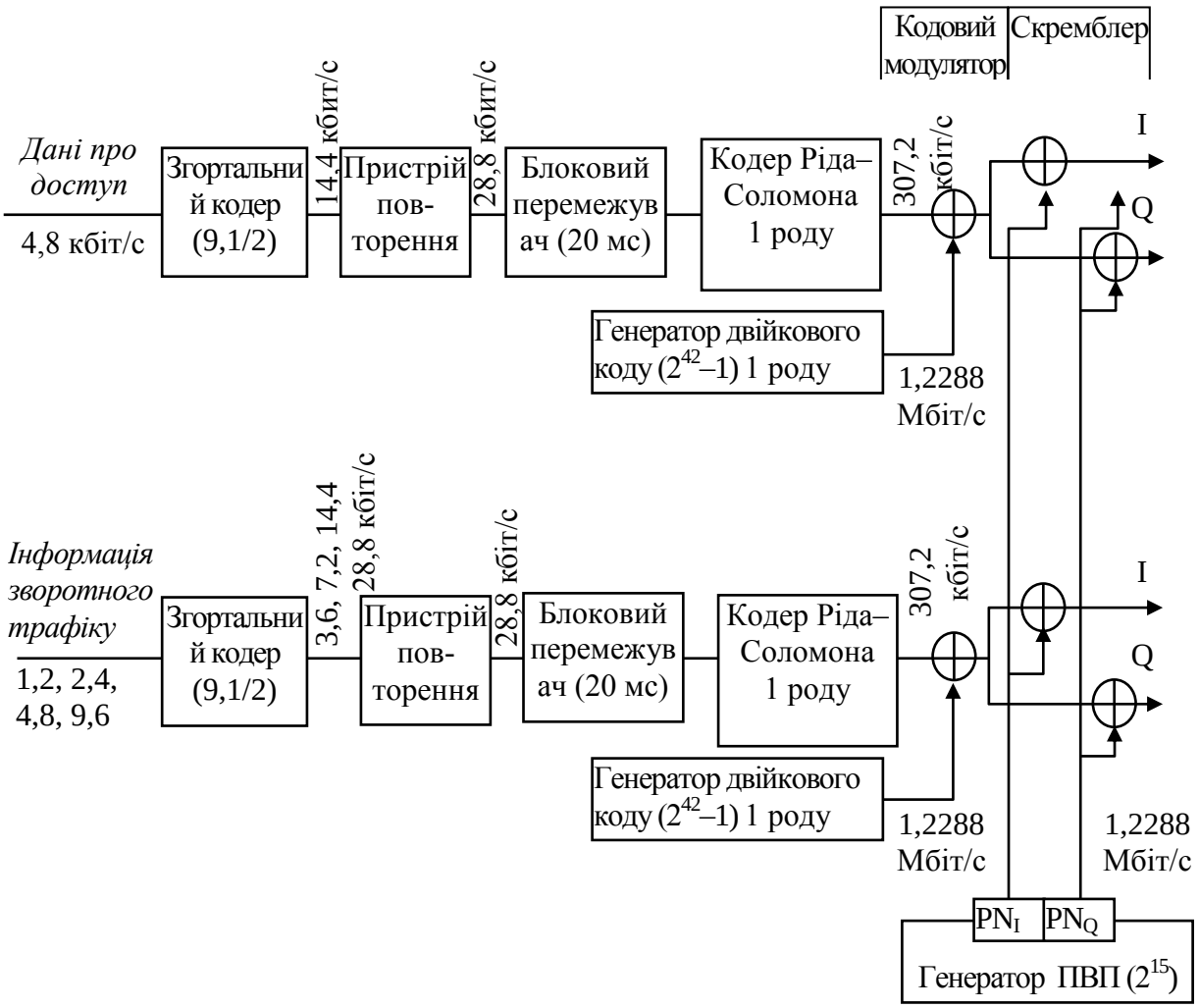


Рисунок 4.13 – Структурна схема передавального тракту мобільної станції

Складний сигнал з тактовою частотою 1,2288 Мбіт/с з виходу кодового модулятора надходить у квадратурні канали скремблера. Тут здійснюється «складанняння за модулем» сигналів з коротким кодом ($L = 2^{15}$) – ПВП PN_I і PN_Q. Всі MS у системі використовують один короткий код – той самий, що і в пілотних каналах BS. Проте циклічний зсув короткого коду фіксований і однаковий для всіх MS.

Далі результуючий сигнал фільтрується в основній смузі частот (1,25 Мгц) і піддається чотирипозиційній фазовій маніпуляції із зсувом – СФМ-4 (offset quaternary phase shift keying – OQPSK). Взаємний часовий зсув сигналів у квадратурних каналах, дорівнює половині біту, вводиться для того, щоб фаза маніпульованого сигналу змінювалася з кроком $\pm\pi/2$. В цьому випадку одночасна зміна символів у кожному з квадратурних каналів не спричиняє небажаних провалів огинаючої радіосигналу. Маніпульованне повідомлення переноситься з проміжної частоти на несучу частоту, посилюється за потужністю, піддається смуговій фільтрації і подається на антену MS.

Для підвищення якості прийому сигналів на BS реалізується просторове рознесення з кратністю, визначеною кількістю встановлених антен. RAKE-приймач BS включає чотири паралельних канали кореляційної обробки сигналів у кожній гілці просторового рознесення, що дозволяє здійснювати прийом сигналів з чотирикратним часовим рознесенням. У кожному каналі обробляється сигнал одного з виділених променів. Пошук найбільш потужних компонентів прийнятого багатопроменевого сигналу проводиться за допомогою двох додаткових скануючих каналів приймача.

Оскільки в стандарті cdmaOne MS не випромінюють пілотних сигналів, у зворотному каналі зв'язку при прийомі використовується некогерентна обробка сигналів. Подальше некогерентне складання гілок дозволяє одержати значний енергетичний виграш, що підвищує перешкодостійкість системи в цілому. Розглянемо особливості функціонування приймача BS (рис. 4.14).

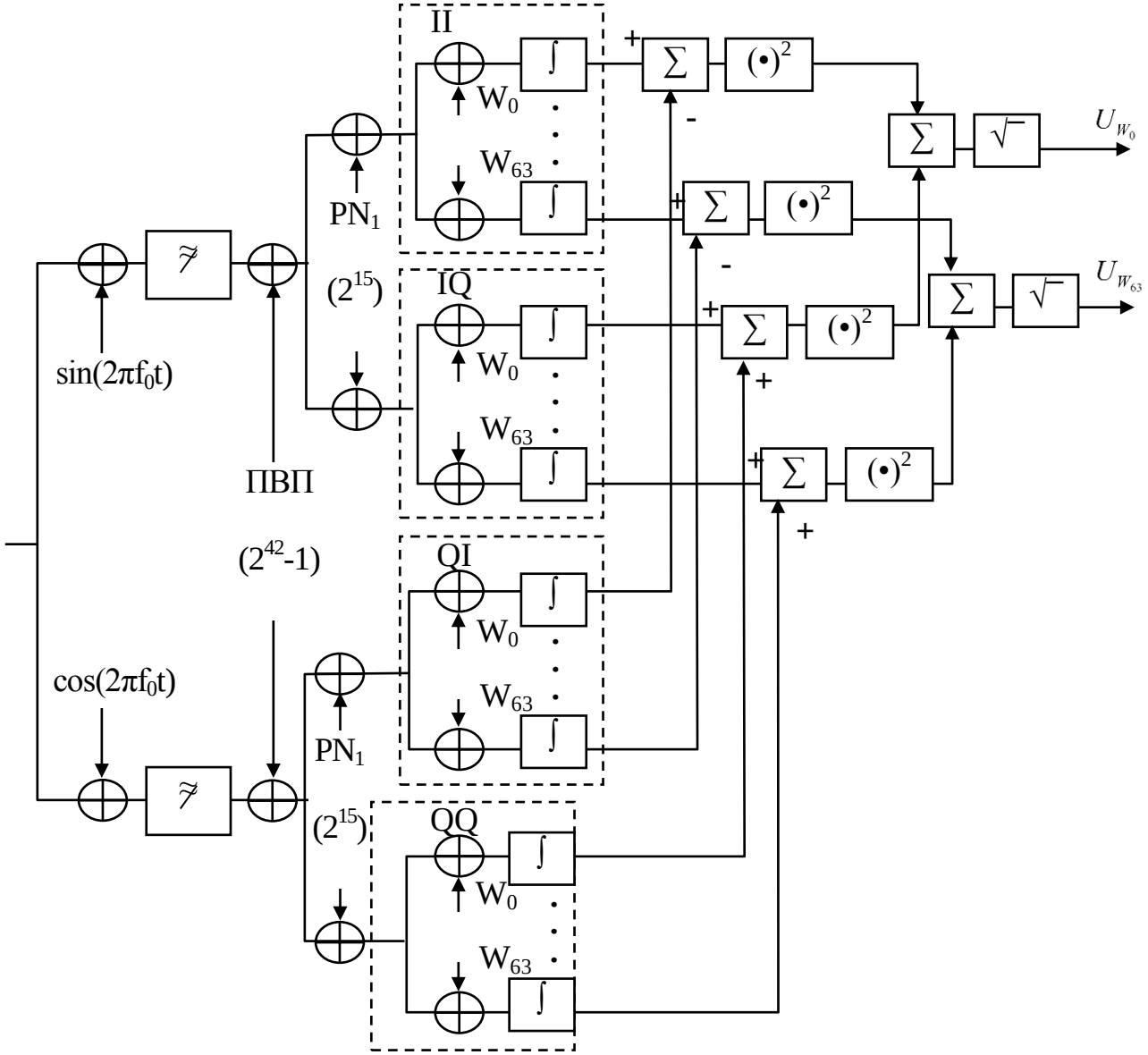


Рисунок 4.14 – Структурна схема каналу кореляційної обробки сигналів приймача BS

Після перетворення (перенесення спектру сигналу на проміжну частоту, балансна демодуляція і низькочастотна фільтрація) сигнал надходить у канали кореляційної обробки. Квадратурні складові сигналу «складаються за модулем 2» спочатку з довгим кодом (2^{42} - 1), а потім – з коротким (2^{15}). При цьому опорні кодові послідовності надходять на суматори з необхідними циклічними зсувом і часовими затримками. Результуючі послідовності поступають у блоки кореляторів. Кожен блок кореляторів складається з 64 паралельно включених кореляторів. На опорні входи кореляторів надходять послідовності, сформовані на основі функцій Уолша W_0 ... W_{63}. Після кореляційної обробки проводиться підсумовування вихідних сигналів однойменних гілок блоків кореляторів і зведення одержаної суми в квадрат. Ця процедура виконується окремо для синфазних і квадратурних складових сигналу. Потім за допомогою операцій підсумовування і добування квадратного кореня знаходяться амплітуди відгуків на виході кожного з 64 субканалів.

Ті ж операції виконуються і в трьох інших паралельних гілках приймача BS. Далі виходи однойменних субканалів кожній з гілок складають з рівними вагами (етап некогерентного об'єднання):

$$Z_j^k = \left|U_{wj}^{(1)}\right|^2 + \left|U_{wj}^{(2)}\right|^2 + \left|U_{wj}^{(3)}\right|^2 + \left|U_{wj}^{(4)}\right|^2, \quad j = 0,\ldots,63,$$

де j – номер субканалу, Z_j^k – статистика сигналу на виході RAKE–приймача в к-й гілці просторового рознесення. Вирішальна статистика Z_j обчислюється в результаті вагового складання статистик окремих гілок.

Порядковий номер j кожного субканалу (номер функції Уолша) відповідає пакету з шести двійкових символів. Вирішальний пристрій вибирає субканал з максимальним Z_j, визначаючи тим самим поточні шість біт інформаційної послідовності, і, таким чином, декодує код Ріда–Соломона 1 роду. Далі символи відновленої послідовності надходять на деперемножувач і згортальний декодер.

Таким чином, приймач BS здійснює «прийом в цілому», функціонуючи за оптимальним правилом: формує вирішальну статистику для всіх можливих типів посилок і вибирає ту посилку, якій відповідає максимальна статистика. Така обробка сигналів дозволяє звести до мінімуму імовірність помилки і забезпечити якісний зв'язок в умовах некогерентного прийому, при дії завмирань.

Канали доступу використовуються мобільними станціями для зв'язку з BS до виділення їм каналів зворотного трафіку

Швидкість передачі даних по каналах доступу становить 4800 біт/с. Маска довгого коду залежить від номера каналу доступу, номера поточного

каналу персонального виклику, інформації ідентифікації BS і циклічного зсуву ПВП у пілотному каналі BS.

MS по каналу доступу передає:

- запит на встановлення з'єднання;
- відповідь на пейджингове повідомлення в каналі персонального виклику;
- дані при реєстрації в системі.

Всі канали доступу, які може використовувати MS, приписані до певних каналів персонального виклику (до 32 каналів доступу на 1 канал персонального виклику). Використання BS і MS жорстко зв'язаних (асоційованих) каналів спрощує протоколи обміну.

Процес вибору каналу доступу мобільною станцією випадковий. MS конфігурує канал довільно з наявного набору масок і циклічних зсувів довгого коду. Це може призвести до того, що декілька MS, налаштованих на один канал персонального виклику, почнуть одночасно вести передачу, що, звичайно ж, спричинить перебої їх роботі. Для запобігання збоям і утриманню відношення сигнал/шум в зворотному каналі зв'язку в заданих межах BS постійно контролює поточну кількість MS, що використовують канали доступу. За необхідності окремим MS (наприклад, з низьким пріоритетом) дається команда звільнити канал доступу. Для керування процесом використання каналів доступу BS передає інформацію про параметри доступу (Access Parameter Message) в заголовку {Overhead Message) каналу персонального виклику.

Канали зворотного трафіку служать для передачі мови і даних абонентів з MS на BS, а також для сигналізації, коли MS вже виділений канал трафіку.

Стандарт cdmaOne дозволяє організувати до 62 каналів зворотного трафіку на 1 канал персонального виклику.

Особливістю є те, що при обробці в тракті каналу зворотного трафіку низькошвидкісні інформаційні кадри (4800, 2400 або 1200 біт/с) стискають і передають із швидкістю 9600 біт/с. Пропуски, що утворилися при цьому, розподіляються по псевдовипадковому закону.

Таким чином, стандарт IS-95 забезпечує високий ступінь безпеки переданих повідомлень і даних про абонентів. Перш за все він має складніший, ніж GSM, радіоінтерфейс, що забезпечує передачу повідомлень кадрами з використанням канального кодування і перемежування з подальшим "розширенням" переданих сигналів за допомогою складених широкосмугових сигналів, сформованих на основі 64 видів послідовностей Уолша і псевдовипадковими послідовностями з кількістю елементів 2^{15} і 2^{42}.

Безпека зв'язку забезпечується також застосуванням процедур автентифікації і шифрування повідомлень. Процедура автентифікації в стандарті IS-95 відповідає процедурі автентифікації стандарту D-AMPS (IS-

54В). Шифрування повідомлень, переданих по каналу зв'язку, здійснюється також з використанням процедур стандарту IS-54B.

У стандарті IS-95 використовується також режим "приватний характер зв'язку", що забезпечується за допомогою секретної маски у вигляді довгого коду. Цей процес також аналогічний процесу формування маски у вигляді довгого коду, який описаний в стандарті IS-54B.

Список джерел інформації

1. Мельников В.В. Защита информации в компьютерных системах / В.В. Мельников. – М.: Финансы и статистика. Электроинформ, 1997. – 367 с.
2. Романец Ю.В. Защита информации в компьютерных системах и сетях / Ю.В Романец., П.А. Тимофеев, В.Ф. Шаньгин. – М.: Радио и связь, 2001. – 376 с.
3. Стенг Д. Секреты безопасности сетей / Д. Стенг, С. Мун. – К.: Диалектика, 1995. – 544 с.
4. Столингс. В. Криптография и защита сетей. Принципы и практика / В. Столингс. – М.: «Вильямс», 2001. – 672 с.
5. Протоколы Интернет. Энциклопедия / Ю.А. Семенов. – М.: Горячая линия – Телеком, 2005. – 405 с.
6. Семенов Ю.А. Протоколы Internet для электронной торговки/ 6.Ю.А. Семенов.– М.: Горячая линия – Телеком, 2005. – 366 с.

Контрольні запитання

1. Назвіть основні механізми безпеки в стандарті GSM.
2. Що входить до складу стандартного модуля достовірності абонента (SIM-карти)?
3. Що використовується для забезпечення конфіденційності переданої радіоканалом інформації?
4. У чому суть алгоритму шифрування A5/1?
5. Порівняльний аналіз криптостійкості алгоритмів A5/1 і A5/2.
6. Що таке тимчасовий міжнародний ідентифікаційний номер користувача (TMSI)?
7. Що входить до складу криптофонів з додатковим криптопроцесором усередині GSM-телефону?
8. Які особливості захисту інформації в системах мобільного зв'язку стандарту IS-95?
9. Назвіть особливості захисту інформації в прямому каналі зв'язку.
10. Які додаткові пристрої, що виконують функцію шифрування, та приєднуються до звичайного GSM-телефону, ви знаєте?

Навчальне видання

СЕМЕНОВ Сергій Генадійович, ПОДОРОЖНЯК Андрій Олексійович,
БАЛЕНКО Олексій Іванович, ГАВРИЛЕНКО Світлана Юріївна

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ ТА МЕРЕЖАХ

Навчальний посібник
для студентів денної та заочної форм навчання напрямків
«Комп'ютерна інженерія»

Відповідальний за випуск проф. *Ф.А. Домнін*
Роботу до видання рекомендував проф. *В.Д. Дмитрієнко*
Редактор *О.С. Самініна*

План 2014 р., поз.34
Підписано до друку 20.05.14. Формат 60х84 1/16. Папір офісний.
Друк – ризографія. Гарнітура *Times New Roman*. Ум. друк. арк. 14,6.
Наклад 200 прим. Зам №.0834. Ціна договірна.

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21.

Надруковано у друкарні видавничого центру "Курсор"
Свідоцтво про державну реєстрацію № 21 від 24.03.2000 р.
61002, Харків, пр. Театральний, 11/13.Тел. (057)706-31-73.