Безпека операційних систем і комп'ютерних мереж

Лекція 21

Віртуальні приватні мережі – VPN (IPsec & SSL/TLS)

План

- IPSec
- SSL/TLS
- SOCKS

Захист віртуальних каналів на мережному рівні

- Утворення захищених віртуальних каналів на мережному рівні дозволяє досягти оптимального співвідношення між прозорістю і якістю захисту
  - Реалізація засобів захисту на цьому рівні робить їх прозорими для мережних застосувань, оскільки мережний рівень завжди буде відокремлений від застосування реалізацією транспортного рівня
  - З іншого боку, на мережному рівні можлива достатньо повна реалізація функцій захисту трафіка і керування ключами, оскільки саме мережний рівень відповідає за маршрутизацію пакетів

Засоби захисту віртуальних каналів на мережному рівні

- Стандартні засоби захисту на мережному рівні для IP мережі визначаються набором протоколів IPSec (англ. – Internet Protocol Security)
  - IPSec є складовою частиною IPv6
  - IPSec є сумісним з версією протоколу IPv4 (підтримка IPSec не є обов'язковою, але бажана, і в наш час, як правило, реалізована)
- IPSec вимагає підтримки стандарту IPSec лише від пристроїв по обидва боки з'єднання, що спілкуються між собою
  - Всі інші пристрої, що розташовані між ними, просто забезпечують передачу IP-пакетів

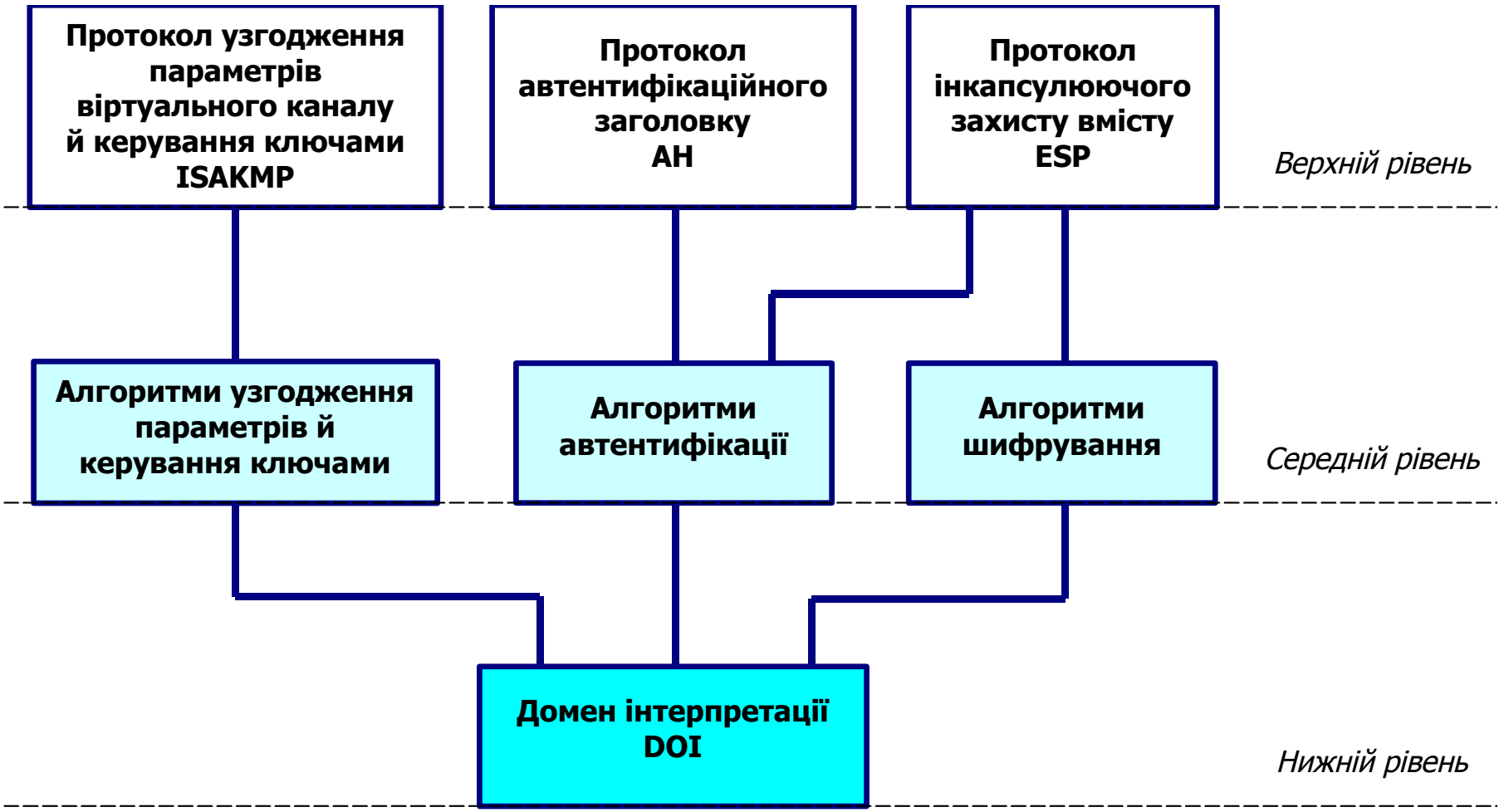
Архітектура засобів захисту IPSec

- Технологія IPSec охоплює кілька абсолютно різних областей, в число яких входять:
  - шифрування,
  - автентифікація
  - керування ключами
- Відповідно до IPSec, архітектура засобів безпеки інформаційного обміну поділяється на три рівня
  - RFC-4301, Security Architecture for the Internet Protocol / S. Kent, K. Seo. – December 2005

Рівні архітектури IPSec

- Верхній рівень – протоколи захисту віртуального каналу і узгодження параметрів захисту
  - Протоколи AH та ESP не залежать від конкретних алгоритмів шифрування й автентифікації. Можуть застосовуватись різні:
    - методи автентифікації
    - типи ключів
    - алгоритми шифрування та розподілу ключів
  - Протоколи AH та ESP зареєстровані організацією IANA (*Internet Address Naming Authority*) під номерами 51 та 50, відповідно
- Середній рівень – криптографічні алгоритми, що використовуються в протоколах AH та ESP, а також певні алгоритми узгодження і керування ключами, які використовує протокол ISAKMP
- Нижній рівень – так званий “домен інтерпретації” (*Domain of Interpretation*, *DOI*)
  - Це, фактично, база даних, яка містить інформацію про усі протоколи і алгоритми, що застосовуються в IPSec, а також про їхні параметри, ідентифікатори тощо
  - Наявність такої бази пояснюється тим, що відкрита архітектура IPSec припускає застосування протоколів і алгоритмів, які не розроблялись для неї чи з урахуванням її вимог
  - Необхідною умовою застосування сторонніх алгоритмів автентифікації або шифрування (наприклад, тих, що відповідають національним стандартам) є реєстрація їх у домені інтерпретації

Архітектура засобів захисту IPSec



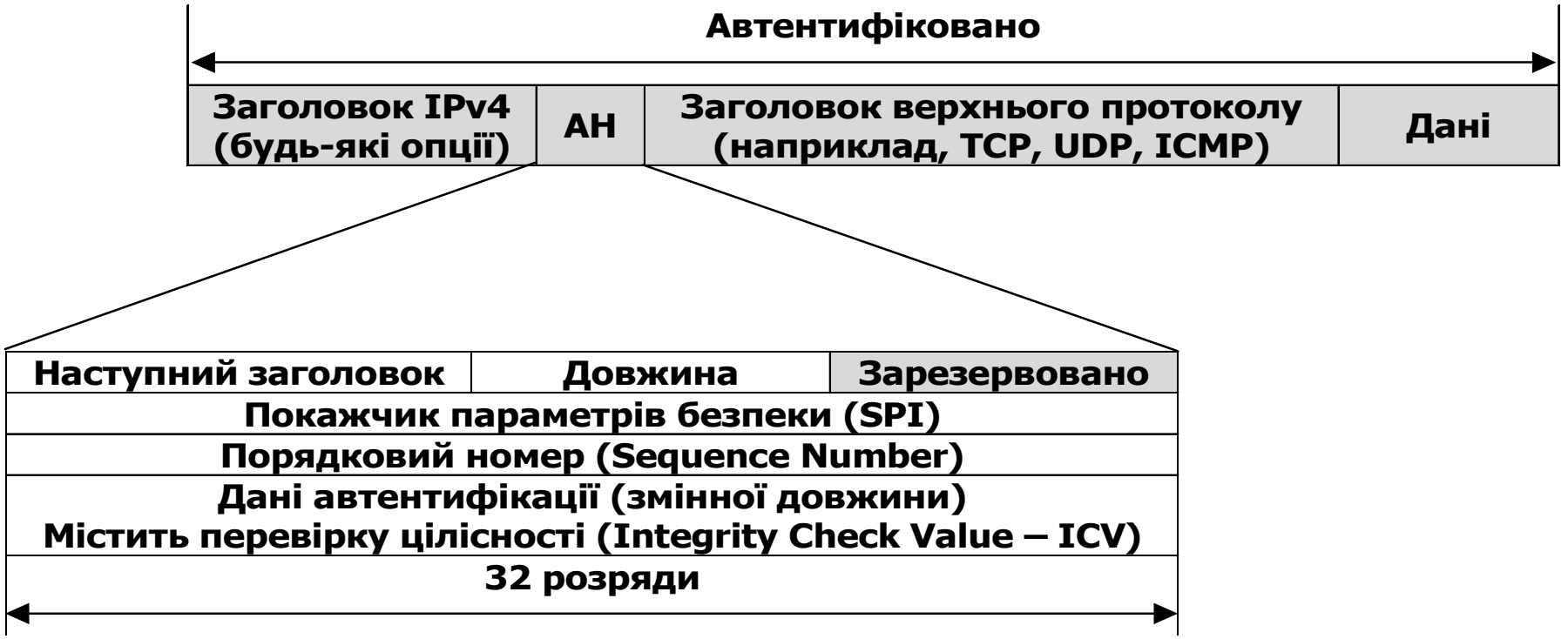
Верхній рівень IPSec

- Протокол автентифікаційного заголовку (*Authentication Header*, AH)
  - RFC-4302, IP Authentication Header / S. Kent. – December 2005
  - Протокол AH передбачає
    - автентифікацію джерела даних
    - перевірку їхньої цілісності і справжності після одержання
    - захист від нав'язування повторних повідомлень
- Протокол інкапсулюючого захисту вмісту (*Encapsulating Security Payload*, ESP)
  - RFC-4303, IP Encapsulating Security Payload (ESP) / S. Kent. – December 2005
  - Протокол ESP крім усіх функцій протоколу AH забезпечує ще й криптографічне закриття пакетів повідомлень
- Протокол узгодження параметрів віртуального каналу й керування ключами (англ. – Internet Security Association Key Management Protocol, ISAKMP)
  - RFC-4306, Internet Key Exchange (IKEv2) Protocol / C. Kaufman, Ed. – December 2005
  - Призначений для попереднього узгодження алгоритмів та їхніх параметрів сторонами, що взаємодіють за протоколами AH та ESP
  - Забезпечує створення сторонами, що взаємодіють, спільного контексту, елементи якого в подальшому вони можуть вільно використовувати.

Асоціації захисту (SA)

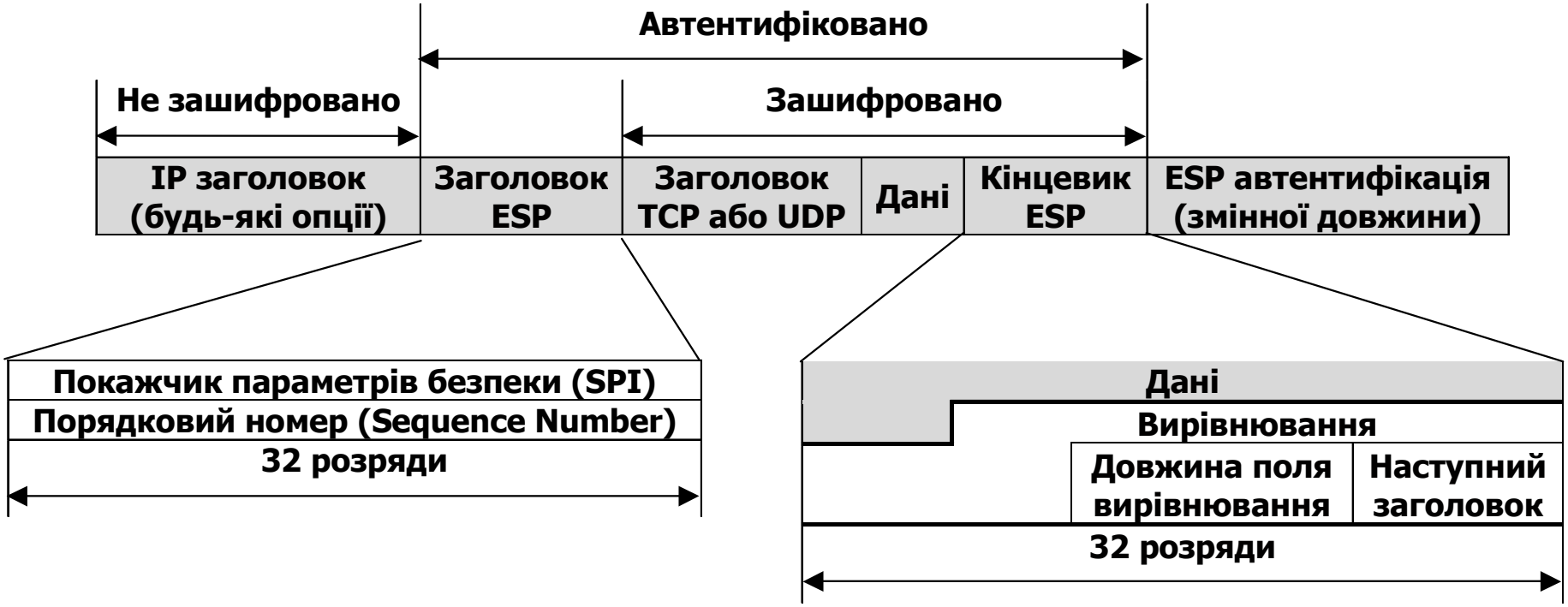
- Контекст, у якому взаємодіють сторони, що використовують технологію IPSec, визначають терміном “асоціація захисту” (*Security Association*, SA)
- Асоціація захисту функціює на основі угоди, що укладається сторонами
- Елементами асоціації захисту є:
  - учасники зв’язку: IP-адреси відправника й одержувача;
  - криптографічний алгоритм;
  - порядок обміну ключами;
  - розміри ключів;
  - термін дії ключів;
  - алгоритм автентифікації.
- Асоціації захисту утворюються відповідно до протоколу ISAKMP

Автентифікаційний заголовок (AH)



- Поле SPI (*Security Parameters Index*) – це “покажчик параметрів безпеки”
  - 32-розрядне число, що вказує на протоколи захисту, що використовуються
  - В це поле включені індекси алгоритмів і типи ключів
  - Фактично, воно визначає асоціацію захисту
- Порядковий номер (*Sequence Number*) визначає кількість пакетів, що відправлені, і забезпечує захист від хибного повторення даних

Протокол інкапсулюючого захисту (ESP)



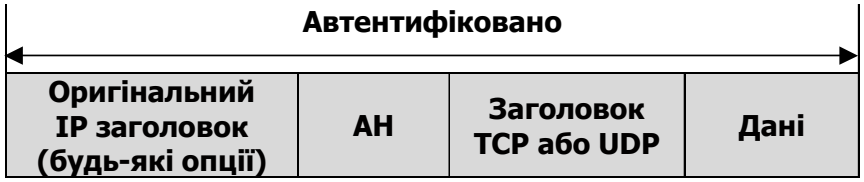
Протокол інкапсулюючого захисту (ESP)

- Протокол ESP забезпечує шифрування IP-інформації на рівні пакетів
  - Передбачено використання різних алгоритмів шифрування
- Протокол ESP забезпечує автентифікацію даних із застосуванням різних алгоритмів автентифікації
- Слід звернути увагу на таке:
  - заголовок ESP розташований між заголовком IP та рештою вмісту пакета;
  - поля покажчика SPI та порядкового номера виконують ту ж функцію, що й у заголовку AH;
  - поле заголовку TCP (або UDP, або іншого протоколу), дані та кінцевик (трейлер) ESP зашифровані;
  - поле вирівнювання має змінну довжину в діапазоні 0-255 біт, і забезпечує, по-перше, що поле “Наступний заголовок” закінчується на межі 32-розрядного слова, а по-друге, що розмір зашифрованої частини кратний розміру блоку застосованого алгоритму шифрування;
  - ESP забезпечує автентифікацію даних у тому ж порядку, що й AH.

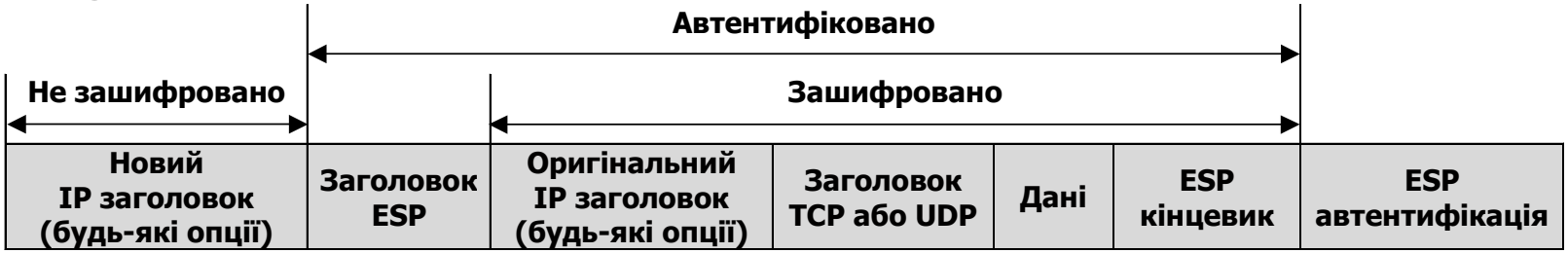
Режим тунелювання та транспортний режим

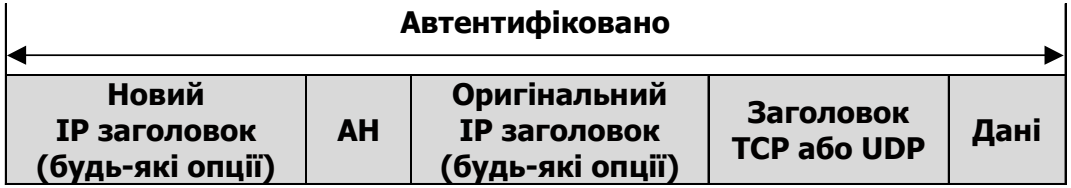
- Як для AH, так і для ESP існують два режими
- Транспортний режим (*Transport Mode*)
  - Призначений для забезпечення зв'язку між двома хостами
  - Не передбачає інкапсуляції IP-пакета в інший пакет
  - У випадку прослуховування трафіка, порушник зможе прочитати справжні IP-адреси відправника й одержувача
- Режим тунелювання (*Tunnel Mode*)
  - Весь IP-пакет поміщається в поле даних пакета IPSec. Далі для пакета вказується нові IP-адреси відправника та одержувача, і додаються захисні заголовки та автентифікаційні трейлери.
  - В новому заголовку адреси відправника й одержувача відрізняються від тих, що вказані у вихідному пакеті
  - Порушник, який несанкціоновано перехопив пакет, не зможе встановити, які саме станції спілкуються між собою
  - Заголовок ESP не шифрується, щоби станція, що приймає повідомлення, мала змогу зрозуміти, що одержаний пакет є пакетом IPSec ESP
  - Вихідний IP-заголовок, дані TCP, інформація, яку передають, та кінцевик ESP шифруються. Ці елементи складають вміст поля даних зовнішнього пакета

Транспортний режим



Режим тунелювання





Обмін ключами

- В IPSec застосовуються два способи передачі ключів:
- Вручну
  - Ключі вручну завантажуються у відповідні пристрої IPsec безпосередньо на об’єктах
  - Шифруванню ці ключі не піддаються, вони або передаються системному адміністратору особисто, або надсилаються поштою
  - Введення ключів вручну виправдано лише у невеликій мережі
- Шляхом обміну через IP-мережу (*Internet Key Exchange, IKE*)
  - Коли масштаби мережі зростають, виникає потреба в механізмі створення асоціацій захисту за вимогою (SA on Demand)
  - За створення асоціацій захисту відповідає протокол ISAKMP, який описує базові технології, але не специфікує конкретні алгоритми
    - Для обміну ключами можуть застосовуватись окремі протоколи
    - Був обраний протокол Oakley, що використовує алгоритм Діффі-Хелмана
  - Поєднання протоколів ISAKMP та Oakley було відомо як специфікації ISAKMP/Oakley, тепер воно отримало назву протоколу IKE

Протокол IKE

- Призначений для узгодження параметрів асоціацій захисту, що створюються, і для автентифікованого обміну ключами, якими будуть користуватись учасники цих асоціацій
- Дозволяє утворити між двома учасниками обміну (IKE SA) автентифікований захищений тунель, за яким будуть узгоджуватись параметри асоціації захисту, що створюється для IPSec
- Протокол на базі UDP, передбачає використання порту 500
- Може функціонувати у трьох режимах:
  - Основний режим (*Main Mode*)
    - застосовується, коли дві сторони вперше встановили зв'язок, щоби узгодити параметри асоціації захисту, яка забезпечить конфіденційність їх подальшого обміну
  - "Активний" режим (*Aggressive Mode*)
    - є скороченою версією основного режиму, має те ж призначення, що й основний режим, і може використовуватись замість нього
  - Швидкісний режим (*Quick Mode*)
    - застосовується, коли асоціація захисту вже створена в результаті використання основного або активного режиму, але існує необхідність в узгодженні функцій захисту або обміну новими ключами
    - оскільки захищений канал був утворений ще до застосування швидкісного режиму, останній забезпечує надійний захист без додаткових витрат, які притаманні основному або активному режиму

Автентифікація у протоколі IKE

- Протокол IKE передбачає кілька способів автентифікації
  - Коли спільно використовуються одні й ті ж ключі
    - Всі хост-системи (або шлюзи VPN) володіють одними й тими ж таємними ключами
    - IKE автентифікує різних учасників обміну по хешу ключа
  - При використанні криптографії з відкритим ключем
    - Кожна сторона генерує випадкове число і шифрує його відкритим ключем іншої сторони
    - Автентифікація відбувається, коли інша сторона може розрахувати хеш-функцію цього випадкового числа і надіслати результат першій стороні
  - Технології цифрового підпису
    - Кожний пристрій “підписує” набори даних, що відсилає іншій стороні
    - Цей метод подібний до шифрування відкритим ключем, але додатково забезпечує захист від відмовлення від авторства
- При використанні асиметричної криптографії (цифровий підпис, шифрування відкритим ключем), необхідно використання цифрових сертифікатів, що підтверджують взаємну відповідність і справжність відкритих та секретних ключів
  - Протокол IKE дозволяє отримати доступ до сертифікату в односторонньому порядку або у формі обміну при виконанні сторонами процедури IKE

Захист віртуальних каналів на сеансовому рівні

- Сеансовий рівень є найвищим рівнем моделі взаємодії відкритих систем, на якому можливо формування захищених віртуальних каналів
- Побудова VPN на цьому рівні дозволяє досягти:
  - найбільшої функціональної повноти захисту інформаційного обміну,
  - надійності контролю доступу,
  - простоти налаштовування системи безпеки.
- При побудові захищених віртуальних мереж на сеансовому рівні є можливість здійснити:
  - криптографічний захист інформації, включаючи автентифікацію
    - найбільшого поширення дістав протокол SSL/TLS (*Secure Sockets Layer / Transport Layer Security*), який був розроблений компанією Netscape Communications
  - деяких функцій посередництва між сторонами, що взаємодіють (саме сеансовий рівень відповідає за встановлення логічних з'єднань і керування ними)
    - IETF прийняла в якості стандарту протокол SOCKS

Прозорість захисту віртуальних каналів на сеансовому рівні

- Протоколи формування захищених віртуальних каналів на сеансовому рівні є прозорими для прикладних протоколів захисту, а також для протоколів прикладного рівня, таких як HTTP, FTP, POP3, SMTP
- З іншого боку, на сеансовому рівні існує залежність від програм, які реалізують високорівневі протоколи
  - На відміну від еталонної моделі OSI, у стеку протоколів TCP/IP розрізняють лише чотири рівні
  - Функції сеансового рівня можуть реалізовуватись або протоколами транспортного рівня (TCP), або протоколами верхнього (прикладного) рівня
  - Реалізація протоколів захисту інформаційного обміну, що відносяться до сеансового рівня, у багатьох випадках вимагає внесення змін у високорівневі мережні програмні засоби

Протокол SSL/TLS

- Протокол Secure Sockets Layer / Transport Layer Security орієнтований на захист інформаційного обміну між клієнтом і сервером комп'ютерної мережі
- Версія TLS 1.0 фактично є розвитком версії SSL 3.0 і мало відрізняється від неї, хоча розробники попереджають про відсутність сумісності
  - Специфікація SSL 3.0 – Netscape Communications, 1996
  - RFC-4346, The Transport Layer Security (TLS) Protocol Version 1.1 / T. Dierks, E. Rescorla. – April 2006
  - RFC-4366, Transport Layer Security (TLS) Extensions / S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, T. Wright. – April 2006
- SSL став загальновизнаним стандартом захисту в Інтернеті та інтранет-мережах і практично витіснив конкуруючі технології шифрування на прикладному рівні, такі як, наприклад, Secure HTTP (SHTTP)

Криптографічні основи SSL/TLS

- Основу протоколу складає технологія комплексного використання асиметричних і симетричних криптосистем
  - Як базові використовуються криптографічні алгоритми:
    - для асиметричного шифрування – RSA,
    - для симетричного шифрування – RC2, RC4, DES та потрійний DES (Triple DES),
    - для хешування – MD5 і SHA-1.
  - Починаючи з версії SSL 3.0 набір криптографічних алгоритмів може розширюватись
  - Для автентифікації сторін, що взаємодіють, і для криптографічного захисту ключа симетричного шифрування застосовуються цифрові сертифікати відкритих ключів, що відповідають стандарту X.509

Процедура встановлення сеансу SSL

- У відповідності до протоколу SSL криптозахищені тунелі утворюються між кінцевими точками віртуальної мережі. Ініціаторами кожного тунелю є клієнт і сервер. Протокол SSL передбачає дві стадії їхньої взаємодії:
  - Встановлення сеансу SSL
  - Захищена взаємодія
- Процедура встановлення сеансу SSL називається “рукостисканням”. Вона виконується за протоколом рукостискання (*Handshake Protocol*), який входить до складу SSL. В ході цієї процедури вирішуються такі завдання:
  - автентифікація сторін;
  - узгодження криптографічних алгоритмів та алгоритмів стискання, які будуть використовуватись при захищеному інформаційному обміні;
  - формування спільного секретного майстер-ключа;
  - генерація на основі майстер-ключа спільних секретних сеансових ключів для криптозахисту інформаційного обміну.
- У версії SSL 3.0 підтримуються три режими автентифікації:
  - взаємна автентифікація сторін;
  - одностороння автентифікація сервера без автентифікації клієнта;
  - повна анонімність.
    - В останньому режимі реалізується захищений обмін між клієнтом і сервером, але не надається жодних гарантій щодо автентичності сторін, що взаємодіють.

Послідовність процедури автентифікації (1/4)

- Наведено послідовність процедури автентифікації, що відповідає другому режиму (автентифікація сервера без автентифікації клієнта)
- Клієнт надсилає серверу запит на встановлення захищеного з'єднання. У запиті передається:
  - поточний час і дата;
  - випадкова послідовність RAND_CL;
  - набір алгоритмів симетричного шифрування та алгоритмів обчислення хеш-функцій, які підтримує клієнт;
  - набір алгоритмів стискання, які підтримує клієнт.
- Сервер надсилає у відповідь узгоджений набір параметрів, який містить:
  - ідентифікатор сеансу SSL;
  - обрані криптографічні алгоритми з числа тих, що запропонував клієнт (якщо запропоновані алгоритми чи їхні параметри з якихось причин не влаштовують сервер, то сесія закривається);
  - сертифікат сервера, завірений цифровим підписом центру сертифікації;
  - випадкову послідовність RAND_SERV.

Послідовність процедури автентифікації (2/4)

- Клієнт, використовуючи відкритий ключ центра сертифікації, здійснює перевірку одержаного сертифіката сервера
  - Якщо перевірка дає негативний результат, то сесія закривається
  - Якщо результат позитивний, то клієнт здійснює такі дії:
    - виробляє випадкову 48-байтну послідовність Pre-MasterSecret, зашифровує її на відкритому ключі сервера, який містився в сертифікаті сервера, і надсилає її серверу;
    - використовуючи обраний сервером алгоритм хешування, виробляє спільний таємний майстер-ключ (MasterSecret), використовуючи для цього послідовності Pre-MasterSecret, RAND_SERV і RAND_CL;
    - використовуючи MasterSecret, обчислює сеансові таємні ключі для симетричного шифрування і обчислення хеш-функцій;
    - переходить у режим захищеної взаємодії.

Послідовність процедури автентифікації (3/4)

- Сервер, отримавши зашифровану послідовність Pre-MasterSecret, розшифровує її, користуючись своїм таємним ключем, а далі виконує такі операції:
  - точно так, як і клієнт, використовуючи обраний алгоритм хешування, виробляє спільний таємний майстер-ключ (MasterSecret), використовуючи для цього послідовності Pre-MasterSecret, RAND_SERV і RAND_CL;
    - оскільки і алгоритм і вихідні послідовності ті ж самі, що й у клієнта, результат (MasterSecret) повинен бути ідентичним
  - точно так, як і клієнт, використовуючи MasterSecret, обчислює сеансові таємні ключі для симетричного шифрування і обчислення хеш-функцій;
    - знову ж, результати повинні бути ідентичні тим, що отримав клієнт
  - переходить у режим захищеної взаємодії.

Послідовність процедури автентифікації (4/4)

- Клієнт і сервер здійснюють перевірку ідентичності параметрів сеансу SSL (ключів):
  - клієнт формує тестове повідомлення із:
    - тих даних, що він відправляв серверу на кроці 1,
    - тих даних, що він одержав від сервера на кроці 2,
    - послідовності MasterSecret;
  - далі він формує код перевірки цілісності повідомлення (MAC), зашифровує повідомлення на спільному таємному сеансовому ключі і надсилає серверу;
  - сервер аналогічним чином формує тестове повідомлення і надсилає його клієнту;
  - кожна із сторін розшифровує одержане тестове повідомлення і здійснює перевірку цілісності.
- В разі успіху перевірки ідентичності параметрів сеанс SSL вважається встановленим і сторони розпочинають захищену взаємодію

Після процедури автентифікації

- Сторони підтримують захищену взаємодію
- В ході захищеної взаємодії
  - Кожна з сторін при відправленні кожного повідомлення формує MAC-код для перевірки цілісності повідомлення, а потім зашифровує повідомлення разом з MAC-кодом
  - При одержанні кожного повідомлення воно розшифровується і здійснюється перевірка його цілісності.
    - В разі виявлення порушення цілісності повідомлення сеанс SSL закривається

Протокол SOCKS

- Протокол SOCKS був розроблений у 1990 році для організації посередництва при взаємодії між клієнт-серверними застосунками на сеансовому рівні моделі OSI
- SOCKS може застосовуватись для реалізації багатьох функцій посередництва, таких як
  - трансляція мережних адрес (Network Address Translation, NAT),
  - контроль за напрямками інформаційних потоків,
  - розмежування доступу в залежності від атрибутів користувачів та інформації.
- У порівнянні з посередницькими функціями, що реалізуються на прикладному рівні, SOCKS пропонує більшу швидкодію та незалежність від високорівневих протоколів, таких як HTTP, FTP, POP3, SMTP тощо.
- Протокол SOCKS не залежить від операційних систем, а також не прив’язаний до протоколу IP

Особливості протоколу SOCKS

- На основі протоколів мережного і канального рівнів захищені тунелі формуються між комп'ютерами (або маршрутизаторами, брандмауерами), а на основі протоколу SOCKS захищені тунелі можуть утворюватись для кожного окремого застосування і сеансу
- Розрізняють SOCKS-сервер, який здебільшого встановлюють на шлюз або брандмауер мережі, та SOCKS-клієнти, які встановлюють на кожний комп'ютер користувача
  - SOCKS-сервер взаємодіє з будь-яким прикладним сервером від імені прикладного клієнта, що відповідає цьому серверу
  - SOCKS-клієнт перехоплює запити від справжніх клієнтів до прикладних серверів і передає ці запити SOCKS-серверу

Протокол SOCKS v5

- Версія 5 протоколу SOCKS (SOCKS v5) запропонована в якості стандарту Інтернет
  - RFC-1928, SOCKS Protocol Version 5 / M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones. – March 1996
  - RFC-1929, Username/Password Authentication for SOCKS V5 / M. Leech. – March 1996
- SOCKS v5 підтримує протоколи TCP та UDP. Таким чином, охоплюються майже всі прикладні протоколи. Але протокол ICMP не підтримується!
  - Разом із виключенням численних проблем з безпекою, недоступними становляться і діагностичні утиліти ping та tracert
- Передбачена автентифікація не лише SOCKS-клієнтів, але й користувачів, від імені яких ці клієнти звертаються. Також можлива двостороння автентифікація.
- SOCKS v5 припускає використання схем адресації як IPv4, так і IPv6
- Автентифікація користувача, яку виконує SOCKS-сервер, може бути основаною на сертифікатах X.509 або на паролях
- Для шифрування трафіка між SOCKS-клієнтом і SOCKS-сервером можуть застосовуватись будь-які протоколи, орієнтовані на сеансовий або нижчі рівні моделі OSI

Схема встановлення з'єднання за протоколом SOCKS

- Запит прикладного клієнта до певного прикладного сервера перехоплюється SOCKS-клієнтом, що встановлений на тому ж комп'ютері
- SOCKS-клієнт повідомляє SOCKS-серверу ідентифікатори усіх методів автентифікації, які він підтримує
- SOCKS-сервер приймає рішення, яким з методів автентифікації скористатись
  - Якщо жодний з запропонованих методів його не влаштовує, з'єднання розривається
- SOCKS-сервер автентифікує користувача, від імені якого виступає клієнт
  - У випадку невдалої автентифікації, сервер розриває з'єднання
- Після успішної автентифікації SOCKS-клієнт передає SOCKS-серверу DNS-ім'я або IP-адресу прикладного сервера, з яким необхідно встановити з'єднання
- SOCKS-сервер на основі правил розмежування доступу приймає рішення про можливість встановити з'єднання
- У випадку встановлення з'єднання прикладний клієнт і прикладний сервер взаємодіють один з одним через SOCKS-клієнт і SOCKS-сервер, причому трафік між останніми може шифруватись (для цього на етапі автентифікації клієнт і сервер виробляють сеансовий ключ)