НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1

«ДОСЛІДЖЕННЯ РЕАЛІЗАЦІЙ ПРОТОКОЛУ SSL»

Виконали:
Студенти групи ФІ-22 мн

Ковальчук Ольга
Коломієць Андрій
Толмачов Євгеній

Київ – 2024

Мережеві Моделі

Мережеві моделі— теоретичний опис принципів роботи набору мережевих протоколів, що взаємодіють один з одним. Зазвичай для зручності їх поділяють на рівні.

Розглянемо дві моделі: OSI та TCP/IP

Модель OSI є концептуальною моделлю, яка характеризує та стандартизує те, як різні програмні та апаратні компоненти, що беруть участь у мережній комунікації, повинні розділяти завдання та взаємодіяти один з одним. Вона має сім рівнів.

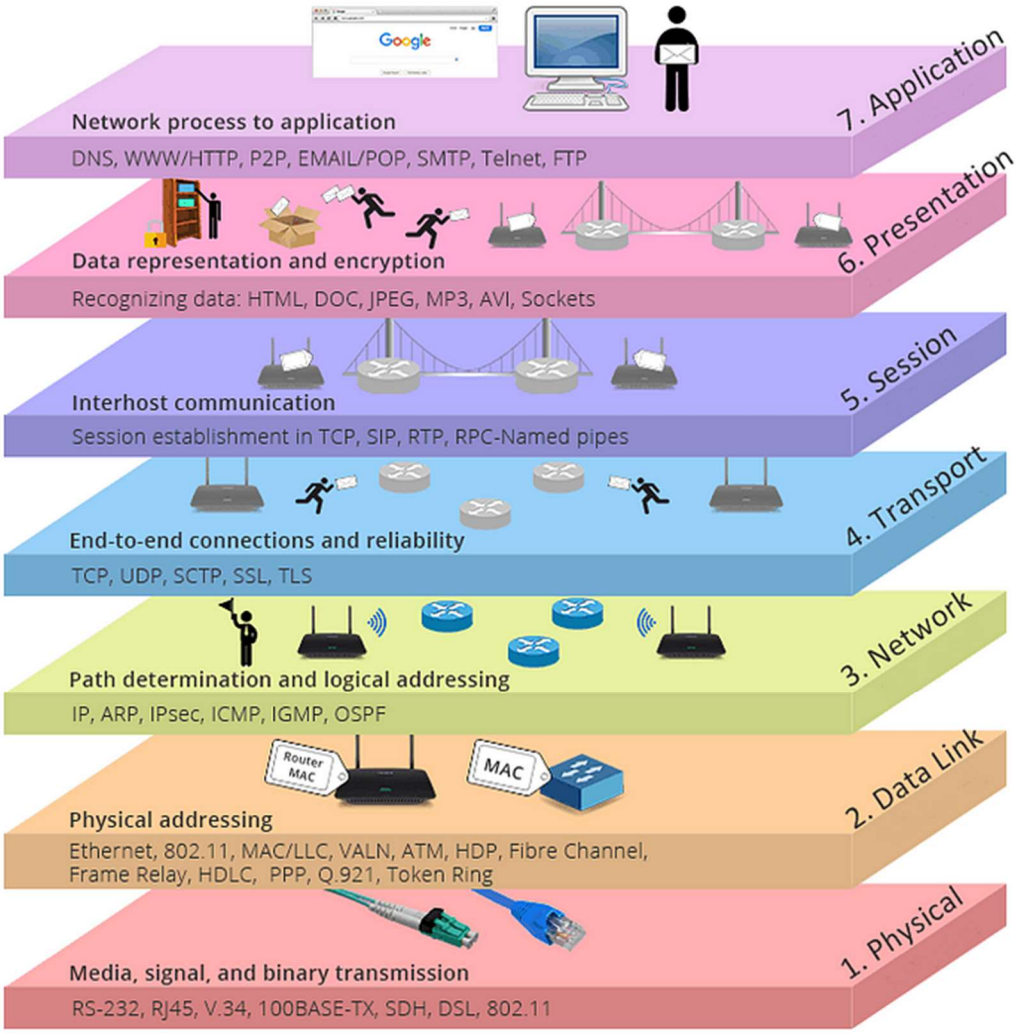

Рисунок 1 схема рівнів моделі OSI

TCP/IP- схожа модель, але вона має 4 рівні. Вона має таку завдяки її основним протоколам - TCP і IP, однак, не тільки ці два протоколи використовуються в цій моделі. Саме використання протоколів відрізняє її від OSI. Можна казати що TCP/IP – практична імплементація спрощеної OSI. В моделі наявні наступні рівні:

Прикладний рівень

Прикладний рівень моделі TCP/IP надає програмам можливість доступу до служб інших рівнів і визначає протоколи, які використовуються програмами обміну даними. Найбільш широко відомі протоколи прикладного рівня: HTTP, FTP, SMTP, Telnet, DNS, SNMP та протокол маршрутизації інформації (RIP).

Транспортний рівень

Транспортний рівень відповідає за надання на прикладному рівні служб зв'язку сеансів та датаграм. Основними протоколами цього рівня є TCP та UDP. Протокол TCP забезпечує надійну службу зв'язку один на один. Він відповідає за послідовність та підтвердження відправлених пакетів, а також відновлення пакетів, втрачених під час передачі. UDP надає ненадійну службу зв'язку один до одного або один до багатьох. UDP зазвичай використовується, коли обсяг даних, що передаються, невеликий (наприклад, дані поміщаються в один пакет).

Мережевий (міжмережевий) рівень

Мережевий рівень відповідає за адресацію хостів та функції маршрутизації. Основними протоколами мережевого рівня є IP, протокол дозволу адрес (ARP), протокол керуючих повідомлень Інтернету (ICMP) та протокол управління групами Інтернету (IGMP). На цьому рівні IP додає заголовок до пакетів, який відомий як IP-адреса.

Рівень доступу до середовища передачі (Network Access Layer)

Рівень доступу до середовища передачі (або канальний рівень) відповідає за розміщення пакетів TCP/IP на мережевому носії та отримання пакетів TCP/IP із мережевого носія. TCP/IP розроблено, щоб бути незалежним від методу

доступу до мережі, формату кадру та середовища. Таким чином, TCP/IP можна використовувати для підключення різних типів мереж, таких як Ethernet, Token Ring та асинхронний режим передавання (ATM).

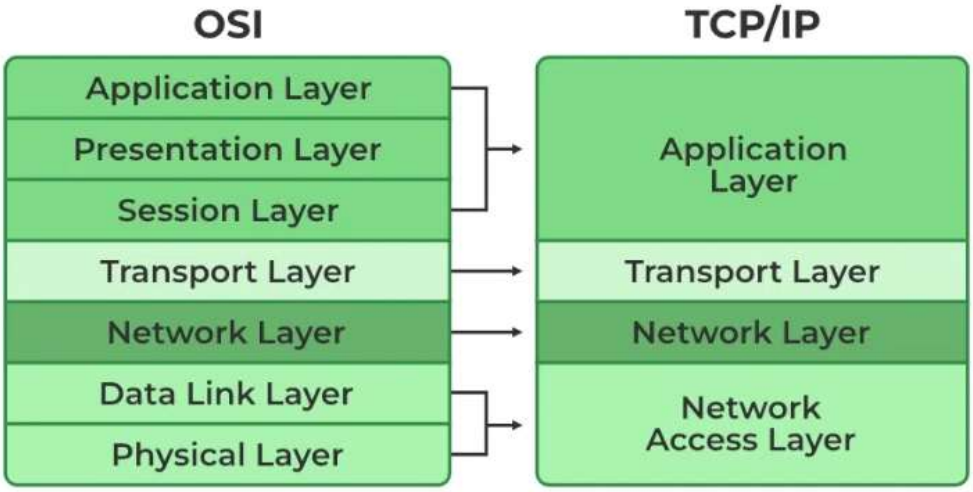

Рисунок 2 порівняння рівнів моделей OSI та TCP/IP

SSL

SSL, або шар захищених сокетів, був оригінальною назвою протоколу, який розробила компанія Netscape у середині 90-х. SSL 1.0 ніколи не був публічно доступним через наявність недоліків, а у версії 2.0 були знайдені серйозні вразливості. Протокол SSL 3.0, випущений в 1996, був повністю перероблений і поставив тон наступної стадії розвитку.

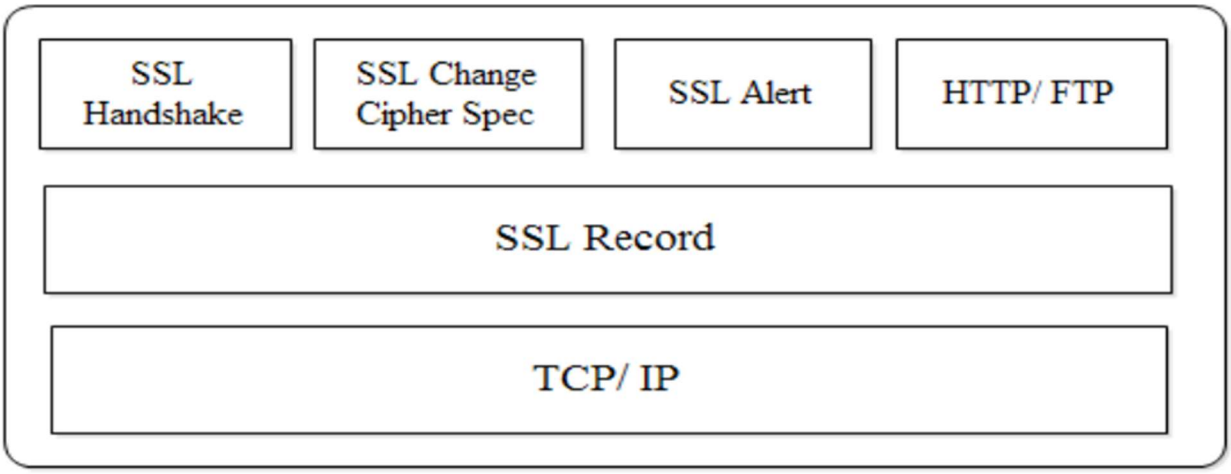

Рисунок 3 Структура шару протоколу SSL

Виконується він між транспортним та прикладним рівнем.

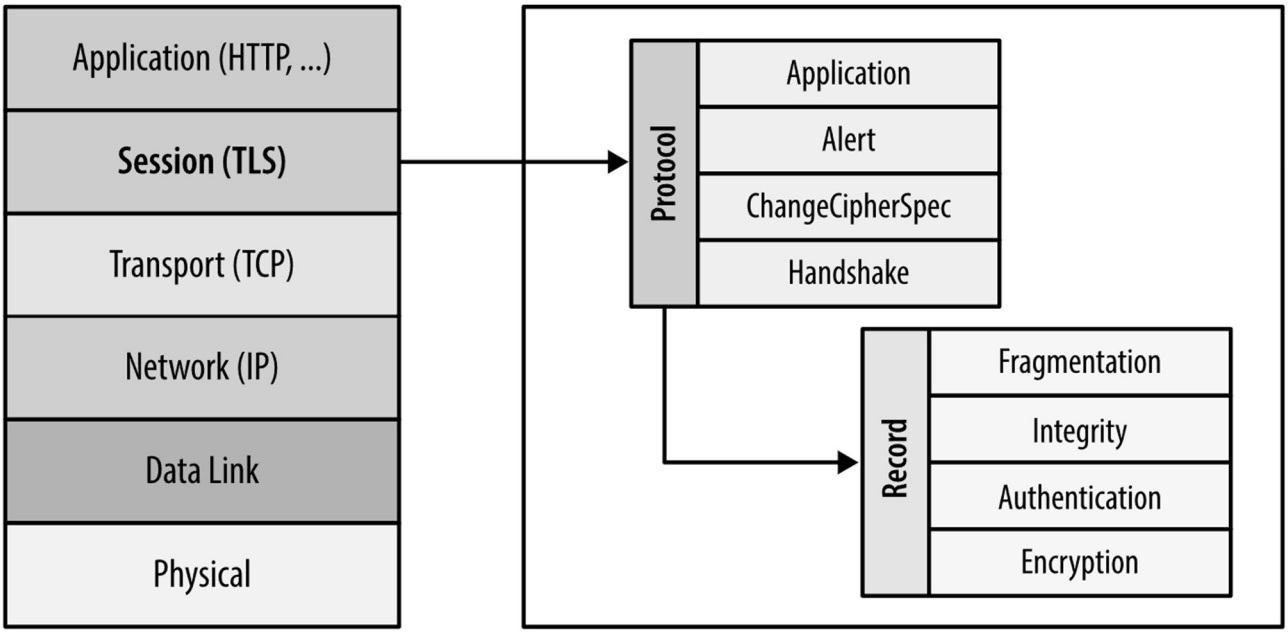

Рисунок 4 Схема зв'язку протоколу SSL/TLS з OSI

SSL Record Protocol (протокол запису)

Є базою решти основних SSL протоколів то надає конфеденційність і цілісність інформації на шарах вище. Сегментує інформацію на певну кількість блоків, стискає, розраховує MAC та шифрує отримані дані (повідомлення та MAC разом). В точці отримання запускається процес розшифрування та розстискання. Починаючи з SSL 3.0 за стандартними налаштуваннями стиснення не проводиться. Те саме правдиво і для всіх версій TLS. Процес роботи відбувається наступним чином:

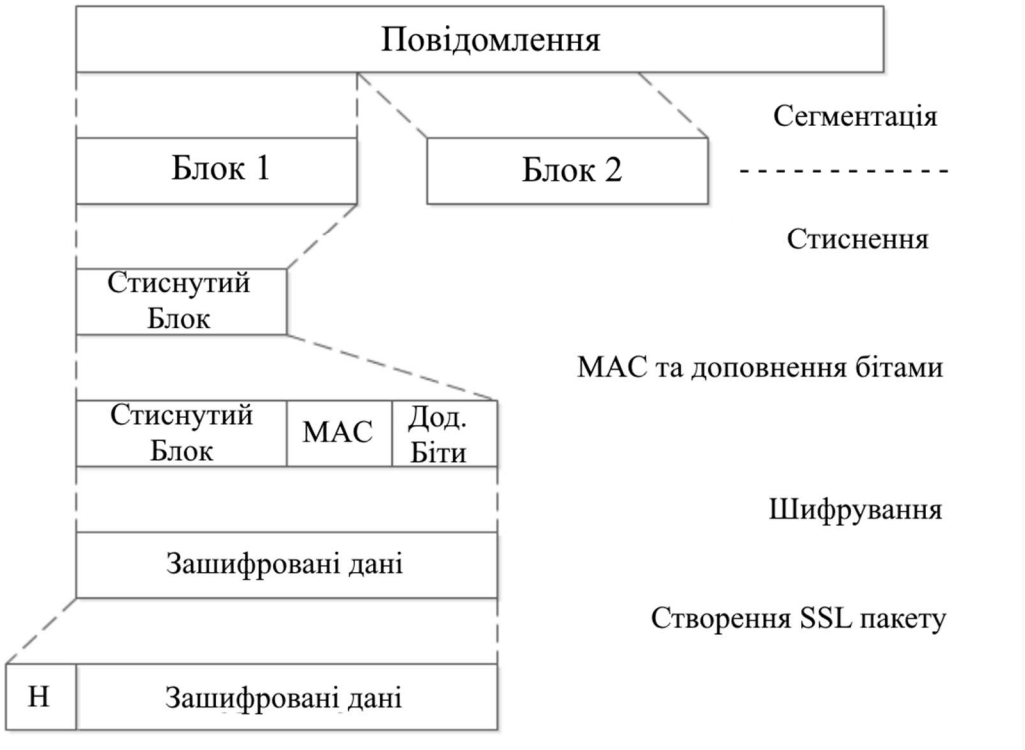

Рисунок 5 схема роботи протоколу запису

I. Алгоритми стиснення: після сегментації викликаються алгоритми стиснення без втрат аби стиснути інформацію без втрат. Наприклад, коди Хоффмана, LZ77, тощо.

II. Алгоритми гешування для додаткового захисту – MD5, SHA, тощо.

III. Алгоритми шифрування: симетричні шифри використовуються для створення даних. В випадках з потоковими шифрами шифрується лише стиснуте повідомлення та мак адреса, в випадках із блочними шифрами наявне доповнення додаткових бітів для створення необхідної кількості блоків певного розміру усіма даними.

SSL Change Cipher Spec Protocol

Найпростіший з протоколів SSL який дозволяє одним байтом сповістити чи має використовуватися нове шифрування. При новому рукостисканні зазвичай має змінитись і шифрування.

SSL Alert Protocol (протокол попереджень)

Протокол для попередження помилок що виникли під час з’єднання. Представляється у двох байтах що також стискаються та шифруються. Перший байт – рівень помилки, де 1 – попередження, 2 – критична помилка. Якщо виникла критична помилка то зв’язок обривається і нових з’єднань в цій SSL сесії не має виконуватись. Другий байт – код помилки.

Нижче наведена таблиця кодів та помилок.

Codes	Alerts	Representations	Types
0	close_notify	No more messages on this link to receiver	Warning
10	unexpected_message	Inappropriate message to receiver	Fatal
20	bad_record_mac	Incorrect MAC record to receiver	Fatal
21	decryption_failed	Invalid decryption due to improper chunk size	Fatal
30	decompression_failure	Decompression fail due to improper input	Fatal
40	handshake_failure	Negotiation fail due to improper security parameters set	Fatal
41	no_certificate	Reply to no proper certificate is available	Warning

42	bad_certificate	Corrupted certificate or contains invalid signature	Warning
43	unsupported_certificate	Sender certificate is unsupported	Warning
44	certificate_revoked	Certificate was withdrawn by signer	Warning
45	certificate_expired	Issued certificate is no longer valid	Warning
46	certificate_unknown	An uncertain problem causes certificate to be inappropriate while handling	Warning
47	illegal_parameter	Security parameter are inconsistent w.r.t. their field in handshake	Fatal

SSL Handshake Protocol (Протокол Рукостискання)

Найперший з протоколів що виконуються при встановленні з'єднання. Клієнт та сервер валідують одне одного та обмінюються необхідними параметрами захисту – тип шифру, ключі, метод стиснення, випадкові числа і тому подібне. Протокол складається з трьох полів – "Type" – 1-байтове поле для вказання типу пакету, "Length" – 3-байтове поле для вказання довжини пакету, "Content" – n-байтове поле із необхідними параметрами.

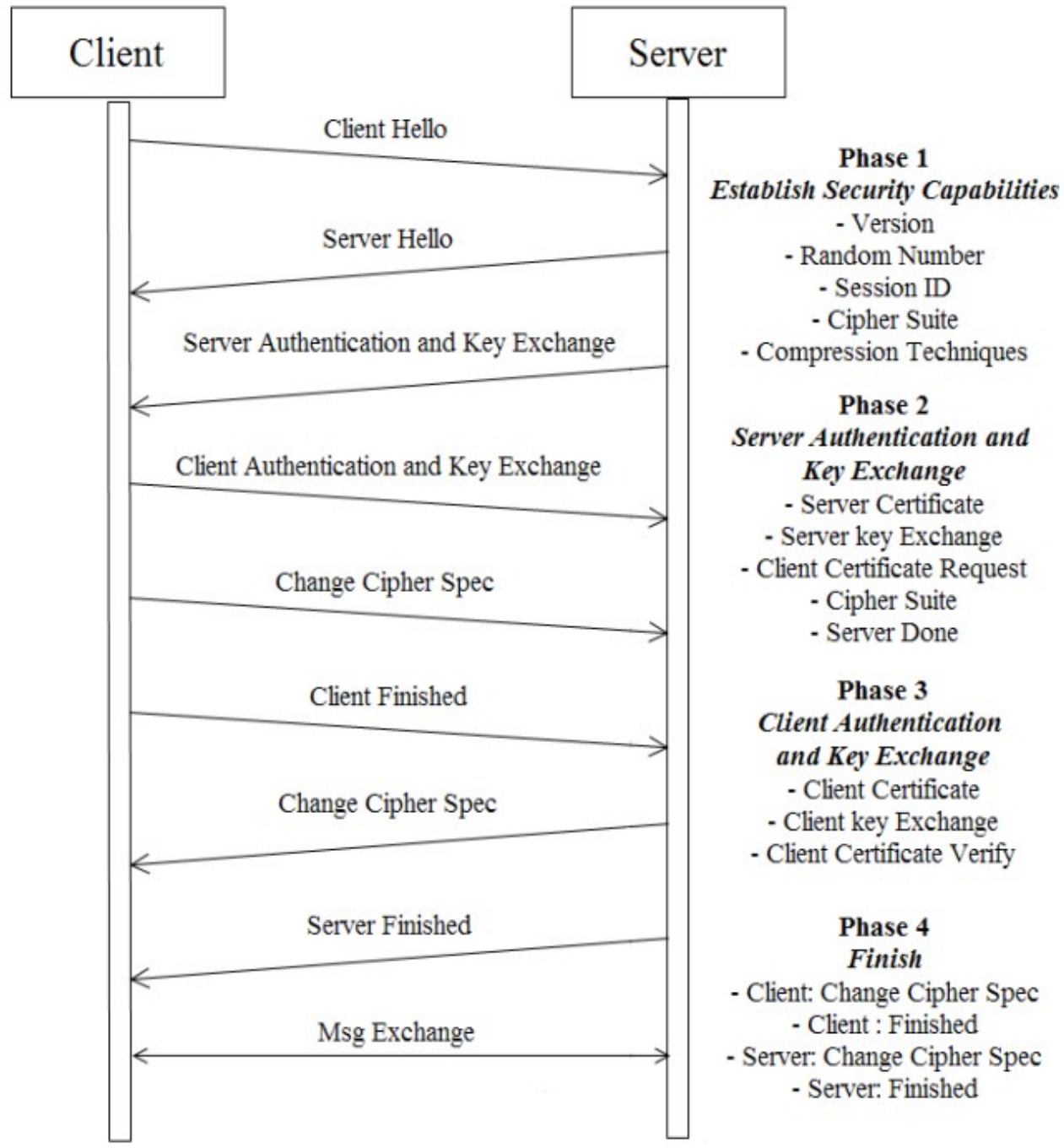

Рисунок 6 схема роботи протоколу рукостискання

В процесі виконання протоколу відбувається обмін повідомленнями.

Фаза 1: З серверу надається версія протоколу, тип шифру, спосіб стиснення, ID сесії та випадкове число.

Фаза 2: Сервер проводить автентифікацію сертифікатом та надає ключі, надає запит інформації для автентифікації від клієнта. Задається основа (параметри) шифру. Робота серверу завершується.

Фаза 3: Клієнт надає сертифікат, ключі та відбувається перевірка сертифікату клієнта.

Фаза 4: Клієнт та сервер завершують роботу протоколу. Перед цим змінюють параметри шифру.

Параметри:

Повідомлення

Codes	Messages	Parameters
0	MT_hello_request	Void
1	MT_client_hello	version,random_no, session_id, cipher_suite, compression_tech
2	MT_sever_hello	version,random_no, session_id, cipher_suite, compression_tech
11	MT_certificate	X.509 certificates chain
12	MT_server_key_exchange	msg_signature, public_parameters
13	MT_certificate_request	cert_authorities, cert_type
14	MT_server_done	Void
	MT_client_key_exchange	msg_signature, public_parameters
15	MT_certificate_verify	cert_signature
20	MT_finished	MD5_hash + SHA_hash

Параметри шифру

Parameters	Values
Key exchange algorithms	RSA, Diffie-Hellman, Fortezza
Cipher algorithm	RC4, RC2, DES, 3DES or IDEA, Fortezza
MAC algorithm	MD5 or SHA
Cipher type	Stream or Block
MAC size	MD5(0 or 16 bytes) or SHA (20 bytes)
IV size	Initialization vector size used in CBC

Алгоритми та параметри алгоритмів

Algorithms	Public Parameters
Ephemeral Diffie-Hellman	A prime no. and its primitive root
RSA	Public key (exponent and Modulo)

Аналогічно в клієнта, але замість публічного ключа – приватний.

TLS

Коли наступну версію протоколу випустили 1999 року, її стандартизувала спеціальна робоча група проєктування мережі Інтернет і дала їй нову назву: захист транспортного рівня, або TLS. Як говориться в TLS-документації, «різниця між цим протоколом та SSL 3.0 не є критичною». TLS і SSL формують серію протоколів, що постійно оновлюються, і їх часто об'єднують під назвою SSL/TLS. Тому очевидно, що в них є схожа архітектура. Зміни в основному в параметрах безпеки, способах обрахунку деяких функцій, електронного підпису та ключів. Також наявні додаткові попередження.

Протокол TLS шифрує інтернет-трафік будь-якого виду. Найпоширеніший вид — вебтрафік. Ви знаєте, коли ваш браузер встановлює з'єднання TLS — якщо посилання в адресному рядку починається з «https».

TLS також використовується іншими програмами, наприклад, у пошті та системах телеконференцій.

Зміни

TLS Record Protocol

Замість MAC використовується гешований код (Hash Message Authentication Code)

TLS Handshake Protocol

- Набір параметрів шифру такий самий як SSL, але без Fortezza. Наявна підтримка Еліптичних Кривих для Діффі-Хелмана.

- Сертифікат Підтвердження Повідомлень – гешує підпис для додаткової безпеки за допомогою MT_certificate_verify.cert_signature.MD5_hash = MD5, MT_certificate_verify.cert_signature.SHA_hash = SHA
- Finished Message геш інакше, передаються до PRF функції
- Master Secret розраховується за допомогою PRF функції
- Падинг відбувається інакше – перед шифруванням випадкова кількість бітів дописуються після MAC так аби зробити чанк розміром з шифроблок. В SSL кількість не випадкова, а мінімальна.

TLS Alert Protocol

Наявні додаткові попередження що роблять з'єднання більш захищеним, наявні в таблиці. Також існує підтримка усіх попереджень окрім коду 41.

Codes	Alerts	Representations	Types
22	Record overflow	Payload size exceeded more than 214 + 2048 bytes	Fatal
48	unknown ca	CA certificate cannot be trusted or discovered	Fatal
49	accessed denied	Negotiation failed due to access control provided by receiver	Fatal
50	decode error	Information could not be decoded properly due to incorrect message length	Fatal
51	decrypt error	Unable to decrypt the secret key, verify digital signature or authenticity of finished message	Warning/ Fatal

60	export restriction	Negotiation against export restriction are detected and terminated	Fatal
70	protocol version	Protocol version is not supported by server	Fatal
71	insufficient security	Handshaking fail due to stronger cipher suite required by server	Fatal
80	internal_error	Error associated to local system and not related to SSL.	Fatal
90	User_cancelled	Abnormal termination of session by user	Fatal
100	no_renegotiation	Client or server response w.r.t hello request is not suitable for renegotiation	Warning

Структура сертифікатів

Сертифікат SSL протоколів це по суті сертифікат X.509 – стандарту що визначає основну структуру. Він визначає основні поля. Існують різні формати такі як PEM, DER, PKCS#7 та PKCS#12. PEM та PKCS#7 мають Base64 ASCII формат кодування, а DER та PKCS#12 – бінарне кодування.

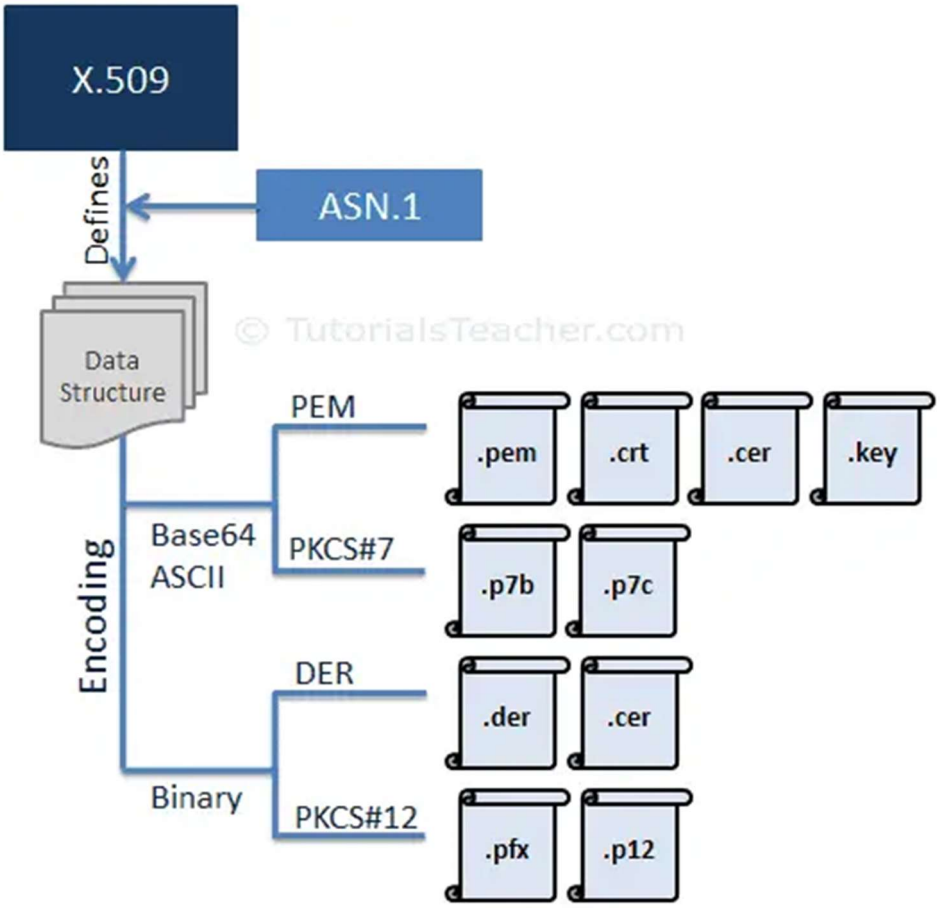

Рисунок 7 схема форматів сертифікатів X.509

Більшість CA надають PEM сертифікати. Вони бувають в форматах .pem, .crt, .cer, та .key. Формат .pem може одразу вміщати сертифікати серверу, проміжні сертифікати, запити та приватні ключі – і все в одному файлі. Їх також можна розділити в окремі файли форматів .crt, .cer та .key.

Оскільки кодування є в ASCII форматі, їх можна редагувати у будь-якому текстовому редакторі. Власне сертифікати між значеннями ---- BEGIN CERTIFICATE---- , ----END CERTIFICATE----; Приватний ключ - між ---- BEGIN RSA PRIVATE KEY----- та -----END RSA PRIVATE KEY-----. Запити – між -----BEGIN CERTIFICATE REQUEST----- та -----END CERTIFICATE REQUEST----- .

Формат PKCS#7

PKCS#7 є стандартом синтаксису криптографічних повідомлень. Сертифікат PKCS#7 використовує кодування Base64 ASCII із розширенням файлів .p7b або .p7c. В цьому форматі можна зберігати лише сертифікати. Сертифікати P7B зазначаються між значеннями -----BEGIN PKCS7----- та -----END PKCS7-----.

Формат DER

DER сертифікати записані у бінарному вигляді, файли мають формати .der і .cer. Ці сертифікати використовуються в основному в веб серверах на Java.

PKCS#12

PKCS#12 сертифікати записані у бінарному вигляді, файли мають формати .pfx та .p12. PKCS#12 сертифікати можуть зберігати власне сертифікат серверу, проміжні сертифікати в .pfx файлі з захистом на паролі. Використовуються в основному при роботі з системами на платформі Windows.

TLS Атаки

Типи атак – атаки на транспортний рівень захисту.

BEAST атака

BEAST атака – або ж Browser Exploit Against SSL/TLS – атака з використанням вразливості TLS 1.0 та була розроблена Т. Донгом та Дж. Райзо. Атака використовує особливості симетричної криптографії та режим CBC для того аби спробувати вгадати секретний ключ яким було зашифровано відкритий текст. В TLS 1.0 останній блок шифротексту є вектором ініціалізації для відкритого тексту. XOR операція між ініціалізованим вектором і відкритим текстом зашифровано симетричним ключем для створення відповідного шифротексту. Якщо аналітику вдається вгадати відкритий текст то він може відтворити секретний ключ. Цю атаку брутфорсом було адресовано в TLS 1.1, 1.2, 1.3.

CRIME атака

Скорочено від Compression Ratio Info Leak Mass Exploitation attack або ж атака за вразливістю інформації про стиснення. Відбувається при взломі сесії шляхом розшифрування cookies сесії в TLS 1.0 та розроблена Т. Донгом та Дж. Райзо. Використовує вразливість алгоритмів стиснення що використовуються в протоколах TLS та SPDY для стиснення заголовків. Вони використовують алгоритм DEFLATE що прибирає повторні строки шляхом стиснення та

шифрування. Ключ такого шифрування отримується шляхом посилання зловмисником різних фальшивих заголовків та отримання різних шифротекстів – і так поки не вийде знайти ключ за розмірами отриманих шифротекстів. Атаку було адресовано в TLS 1.1, 1.2, 1.3 шляхом вимкнення обов'язкового стиснення за стандартними налаштуваннями.

Атака за часом

Атака подібного характеру – розроблена Т. Бері та А. Шульманом – за різними повідомленнями, локаціями, суфіксами та префіксами аналітик отримує різний час відповіді на HTTP запит і таким чином відтворює побітово ключ.

LUCKY 13

Це одна з найбільших вразливостей SSL на даний момент. Атаку розробили Н. А. Фардан і К. Патерсон в 2013 році. Використовує техніку ораклу доповнювання – атаку стороннім каналом яка аналізує вплив доповнення блоків на шифротекста. Порівнює час відповіді сервера на різні шифротексти замінюючи по 1 біту. TLS пакети з справжнім доповненням швидше оброблюються. Якщо у відповід зловмисник отримає помилку, то її можливо буде використати для подальшої атаки. На атаку в середньому витрачається 2^{13} сесій із сервером.

Список Джерел

- https://aws.amazon.com/compare/the-difference-between-ssl-and-tls/
- Satapathy, Ashutosh & Livingston, Jenila. (2016). A Comprehensive Survey on SSL/ TLS and their Vulnerabilities. International Journal of Computer Applications. 153. 31-38. 10.5120/ijca2016912063. : https://www.researchgate.net/profile/Ashutosh-Satapathy/publication/310761924_A_Comprehensive_Survey_on_SSL_TLS_and_their_Vulnerabilities/links/58d1045e92851c1db43dfbfd/A-Comprehensive-Survey-on-SSL-TLS-and-their-Vulnerabilities.pdf
- https://dl.acm.org/doi/pdf/10.1145/3278532.3278568
- https://medium.com/@yangmuxizi/tcp-ip-vs-osi-%D0%BA%D0%B0%D0%BA%D0%B8%D0%B5-

%D1%80%D0%B0%D0%B7%D0%BB%D0%B8%D1%87%D0%B8%D1%8F-%D1%83-%D1%8D%D1%82%D0%B8%D1%85-%D0%B4%D0%B2%D1%83%D1%85-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B5%D0%B9-7f6e6c7c12ce