Міністерство освіти і науки України
Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Комп'ютерний практикум 1
з дисципліни
«Проектування, розробка і реалізація криптографічних систем»
Тема:
«Дослідження реалізацій протоколу SSL»

Виконав:
Студент групи ФІ-22мн
Кушнір Олександр
Перевірив:
Селюх П. В.

Київ – 2023

Набір інструментів OpenSSL включає:

- libssl реалізація всіх версій протоколу TLS до TLSv1.3 (RFC 8446), версій протоколу DTLS до DTLSv1.2 (RFC 6347) і протоколу QUIC (наразі лише на стороні клієнта) версії 1 (RFC 9000).
- libcrypto — повноцінна криптографічна бібліотека загального призначення. Він становить основу реалізації TLS, але також може використовуватися окремо.
- openssl інструмент командного рядка OpenSSL, швейцарський армійський ніж для криптографічних завдань, тестування та аналізу.

Його можна використовувати для:

- створення ключових параметрів
- створення сертифікатів X.509, CSR і CRL
- розрахунок дайджестів повідомлень
- шифрування та дешифрування
- Тести SSL/TLS/DTLS і клієнта і сервера
- QUIC клієнтські тести
- обробка підписаної або зашифрованої пошти S/MIME

Превалюючим стандартом структури цифрового сертифіката є X509, а сертифікат виробничого рівня видаються центром сертифікації (CA, certificate authority), таким як Verisign.

Цифровий сертифікат містить різну інформацію (наприклад, дати початку та закінчення терміну дії, а також доменне ім'я власника), включаючи ідентифікаційні дані видавця та цифровий підпис, який є зашифрованим криптографічним хеш-значенням. Сертифікат також має незашифроване хеш-значення, яке служить його ідентифікуючим відбитком (fingerprint).

Хеш-значення утворюється внаслідок зіставлення довільної кількості бітів з хеш-сумою (digest) фіксованої довжини. Що є біти (бухгалтерський звіт, роман чи, можливо, фільм) немає значення. Наприклад, хеш-алгоритм Message Digest версії 5 (MD5) відображає вхідні біти будь-якої довжини в 128-бітове хеш-значення, тоді як алгоритм SHA1 (Secure Hash Algorithm версії 1) відображає вхідні біти в 160-бітове значення. Різні вхідні біти призводять до різних – дійсно статистично унікальних – значень хеш-функції. У наступній статті ми розглянемо це докладніше і зосередимося на тому, що робить хеш-функцію криптографічної.

Цифрові сертифікати розрізняються за типом (наприклад, кореневий/root, проміжний/intermediate та кінцевий/end-entity сертифікати) і утворюють ієрархію, що відображає ці типи. Як випливає з назви, кореневий сертифікат знаходиться на вершині ієрархії, а сертифікати під ним успадковують будь-яку довіру, яка має кореневий сертифікат. Бібліотеки OpenSSL та більшість сучасних мов програмування мають тип X509, як і функції, які працюють із такими сертифікатами. Сертифікат від

Google має формат X509 і клієнт перевіряє, чи є цей сертифікат X509_V_OK.

Сертифікати X509 засновані на інфраструктурі відкритих ключів (PKI, public-key infrastructure), яка включає алгоритми (в основному домінує RSA) для створення пар ключів: публічний ключ і його парний приватний ключ. Публічний ключ – це ідентифікаційні дані (identity): публічний ключ Amazon ідентифікує його, а мій публічний ключ ідентифікує мене. Приватний ключ повинен зберігатись його власником у секреті.

Ключі у парі мають кілька стандартних застосувань. Публічний ключ можна використовувати для шифрування повідомлення, а приватний ключ з тієї ж пари потім можна використовувати для його розшифрування. Також можна використовувати приватний ключ для підпису документа або іншого електронного артефакту (наприклад, програми або електронного листа), а потім публічний ключ з пари можна використовувати для перевірки цього підпису. Розглянемо два приклади.

У першому прикладі Аліса відкриває свій публічний ключ усьому світу, включаючи Боба. Потім Боб шифрує повідомлення за допомогою публічного ключа Аліси, надсилаючи їй зашифроване повідомлення. Повідомлення, зашифроване публічним ключем Аліси, розшифровується її приватним ключем, який (за ідеєю) є тільки в неї.

Розшифрування повідомлення без приватного ключа Аліси в принципі можливе, але на практиці нереальне, враховуючи надійність криптографічної системи парних ключів, такий як RSA.

Тепер як другий приклад розглянемо підписання документа на підтвердження його справжності. Алгоритм підпису використовує приватний ключ із пари для обробки криптографічного хешу документа, що підписується.

Припустимо, що Аліса підписує цифровим підписом контракт, надісланий Бобу. Потім Боб може використовувати публічний ключ Аліси для перевірки підпису.

Неможливо підробити підпис Аліси без її приватного ключа: отже, на користь Аліси зберегти її приватний ключ у таємниці. Жоден із цих елементів безпеки, за винятком цифрових сертифікатів, не є явним у нашій програмі-клієнті.