УДК 004.49

Андрюшин В.С.
Центральноукраїнський національний технічний університет

Огляд криптографічного протоколу SSL

SSL (англ. Secure Sockets Layer — рівень захищених сокетів) — криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером. SSL спочатку розроблений компанією Netscape Communications. Згодом на підставі протоколу SSL 3.0 був розроблений і прийнятий стандарт RFC, що отримав ім'я TLS.

Протокол забезпечує конфіденційність обміну даними між клієнтом і сервером, що використовують TCP / IP, причому для шифрування використовується асиметричний алгоритм з відкритим ключем. При шифруванні з відкритим ключем використовується два ключі, причому будь-який з них може використовуватися для шифрування повідомлення. Тим самим, якщо використовується один ключ для шифрування, то відповідно для розшифрування потрібно використовувати інший ключ. У такій ситуації можна отримувати захищені повідомлення, публікуючи відкритий ключ, і зберігаючи в таємниці секретний ключ.

Протокол SSL складається з двох підпротоколів: протокол SSL запису і рукостискання. Протокол SSL запису визначає формат, який використовується для передачі даних. Протокол SSL включає рукостискання з використанням протоколу SSL запису для обміну серіями повідомлень між сервером і клієнтом, під час встановлення першого з'єднання. Для роботи SSL потрібно, щоб на сервері був SSL- сертифікат.

SSL надає канал, що має три основні властивості:

- Аутентифікація. Сервер завжди автентифікований, в той час як клієнт автентифікований в залежності від алгоритму.
- Цілісність. Обмін повідомленнями включає в себе перевірку цілісності.
- Конфіденційність каналу. Шифрування використовується після встановлення з'єднання і використовується для всіх наступних повідомлень.

У протоколі SSL всі дані передаються у вигляді записів-об'єктів, що складаються із заголовка і переданих даних. Передача починається із заголовка. Заголовок містить або два, або три байти коду довжини. Причому, якщо старший біт в першому байті коду дорівнює одиниці, то запис не має заповнювача і повна довжина заголовка дорівнює двом байтам, інакше запис містить заповнювач і повна довжина заголовка дорівнює трьом байтам. Код довжини запису не включає в себе число байт заголовка. Довжина запису 2-х байтового заголовка:

RecLength = (byte [0] & 0x7F <<8) | byte [1];
Тут byte [0] і byte [1] перший і другий отримані байти.
Довжина запису 3-х байтового заголовка:
RecLength = (byte [0] & 0x3F <<8) | byte [1];
Escape = (byte [0] & 0x40)! = 0;
Padding = byte [2];

Тут Padding визначає число байтів доданих відправником до початкового тексту, для того щоб зробити довжину запису кратною розміру блока шифру, при використанні блокового шифру. Тепер відправник «заповненого» запису додає заповнювач до наявних даних, і шифрує все це. Причому вміст заповнювача ніякої ролі не має. Через те, що обсяг переданих даних відомий, то заголовок може бути сформований з урахуванням Padding. У свою чергу одержувач запису дешифрує все поле даних і отримує повну вихідну інформацію. Потім обчислюється значення RecLength за відомим Padding, і заповнювач з поля даних видаляється. Дані запису SSL складаються з трьох компонент:

- MAC_Data [Mac_Size] — (Message Authentication Code) — код аутентифікації повідомлення;
- Padding_Data [Padding] — дані заповнювача;
- Actual_Data [N] — реальні дані.

Коли записи надсилаються відкритим текстом, очевидно, що ніякі шифри не використовуються. Тоді довжина Padding_Data і MAC_Data дорівнюють нулю. При використанні шифрування, Padding_Data залежить від розміру блоку шифру, а MAC_Data залежить від вибору шифру. Приклад обчислення MAC_Data:

MacData = Hash (Secret, Actual_Data, Padding_Data, Sequence_Number);

Значення Secret залежить від того, хто (клієнт або сервер) посилає повідомлення. Sequence_Number — лічильник, який інкрементується як сервером, так і клієнтом. Тут Sequence_Number є 32-х бітовий код, який передається хеш-функції у вигляді 4-х байт, причому першим передається старший байт. Для MD2, MD5 MAC_Size дорівнює 16 байтам (128 бітам). Для 2-х байтового заголовка максимальна довжина запису дорівнює 32767 байтам, а для 3-х байтового заголовка 16383 байти.

Протокол SSL був спочатку розроблений компанією Netscape. Версія протоколу 1.0 публічно не випускалася. Версія 2.0 була випущена в лютому 1995 року, але «містила багато недоліків з безпеки, які, в кінцевому рахунку, привели до створення версії 3.0», яка була випущена в 1996 році. Тим самим версія SSL 3.0 послужила основою для створення протоколу TLS 1.0, стандарт протоколу Internet Engineering Task Force (IETF) вперше був визначений в RFC 2246 в січні 1999 року. Visa, Master Card, American Express і багато інших організацій, що працюють з інтернет грошима, мають ліцензію на використання протоколу SSL, для комерційних цілей в мережі Інтернет.

SSL працює модульним способом. Тим самим SSL розширюваність згідно з проектом про підтримку передньої і зворотної сумісності та переговорів між сполуками в однорангової мережі.

Значне використання протоколу SSL призвело до формування протоколу HTTPS (Hypertext Transfer Protocol Secure), що підтримує шифрування. Дані, які передаються по протоколу HTTPS, «упаковуються» в криптографічний протокол SSL або TLS, тим самим забезпечуючи захист цих даних. Такий спосіб захисту широко використовується у світі Веб для додатків, в яких важлива безпека з'єднання, наприклад у платіжних системах. HTTPS підтримується всіма браузерами. На відміну від HTTP, для HTTPS за замовчуванням використовується TCP-порт 443.

Спочатку віртуальні приватні мережі (VPN) на основі SSL розроблялися як додаткова і альтернативна технологія віддаленого доступу на основі IPsec VPN. Однак, такі фактори як достатня надійність і дешевизна зробили цю технологію привабливою для організації VPN. Також SSL отримав широке застосування в електронній пошті.

Основні цілі протоколу. Криптографічна безпека: SSL встановлює безпечне з'єднання між двома сторонами. Відкритість: Програмісти, незалежно один від одного, можуть створювати додатки, що використовують SSL, які згодом будуть здатні успішно обмінюватися криптографічними параметрами без всякого знання коду чужих програм. Розширюваність: SSL прагне забезпечити робочий простір, в якому нові відкриті ключі і трудомісткі методи шифрування можуть бути додані при необхідності. Відносна ефективність: робота протоколу на основі SSL вимагає великих швидкостей від CPU, зокрема для роботи з відкритими ключами. Тому до SSL протоколу була включена необов'язкова схема кешування сесій для зменшення кількості з’єднань, які необхідно встановлювати з нуля. Крім того, велика увага приділяється тому, щоб зменшити мережеву активність.

В SSL використовуються алгоритми: для обміну ключами та перевірки їх достовірності застосовуються: RSA, Diffie-Hellman, ECDH, SRP, PSK; для аутентифікації: RSA, DSA, ECDSA; для симетричного шифрування: RC2, RC4, IDEA, DES, Triple DES або AES, Camellia; для хеш-функцій: SHA, MD5, MD4 і MD2.

Список використаних джерел

1. *Шнайер Б., Прикладная криптография. 2-е издание. Протоколы, алгоритмы и исходные тексты на языке С. – Пер. с англ.: М.: Издательство ТРИУМФ, 2002. 816 с.*
2. *Обуховська Т. І. Захист персональних даних в умовах розвитку інформаційного суспільства: передумови, принципи та міжнародне законодавство [Електронний ресурс] / Обуховська Т.І.// Вісник НАДУ. – 2014,. – №1 – С. 95-103. – Режим доступу: http://visnyk.academy.gov.ua/wp-content/uploads/2014/05/2014-1-17.pdf.*