TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
----------o0o----------

NHẬP MÔN

AN TOÀN THÔNG TIN

Đề tài: Tìm hiểu về SSL/TLS và các ứng dụng

GVHD: PGS.TS. Nguyễn Linh Giang

Nhóm SVTH:

Lương Thị Linh	20173236
Nông Thị Dương	20173063
Trần Thị Ánh Ngọc	20173288
Nguyễn Đức Mạnh Hoàng	20173139

TP. HÀ NỘI, THÁNG 06 NĂM 2020

Mục Lục

Danh mục từ viết tắt

SSL	SECURE SOCKET LAYER
TLS	TRANSPORT LAYER SECURITY
CA	Certificate Authority
MAC	Message Authentication Code
LDAP	Lightweight Directory Access Protocol

Giới thiệu

Việc kết nối giữa một Web browser tới bất kỳ điểm nào trên mạng Internet đi qua rất nhiều các hệ thống độc lập mà không có bất kỳ sự bảo vệ nào với các thông tin trên đường truyền. Không một ai kể cả người sử dụng lẫn Web server có bất kỳ sự kiểm soát nào đối với đường đi của dữ liệu hay có thể kiểm soát được liệu có ai đó thâm nhập vào thông tin trên đường truyền. Để bảo vệ những thông tin mật trên mạng Internet hay bất kỳ mạng TCP/IP nào, SSL (Secure Sockets Layer) đã kết hợp những yếu tố sau để thiết lập được một giao dịch an toàn: đó là khả năng bảo mật thông tin, xác thực và toàn vẹn dữ liệu đến người dùng. SSL được tích hợp sẵn vào các browser và Web server, cho phép người sử dụng làm việc với các trang Web ở chế độ an toàn.

Trong đề tài này, chúng em tiến hành tìm hiểu chi tiết về giao thức bảo mật SSL và TLS. SSL đã được chấp nhận phổ biến trên World Wide Web để liên lạc được xác thực và mã hóa giữa máy khách và máy chủ. Nội dung của bài báo cáo cụ thể gồm 4 phần:

Phần 1: Giao thức SSL/TLS

Phần 2: Các cơ chế bảo mật trong SSL/TLS

Phần 3: Xác thực trong giao thức

Phần 4: Ứng dụng của SSL/TLS

Phần 5: Demo

Chúng em hy vọng rằng với bài báo cáo này sẽ đem đến cái nhìn cụ thể hơn về SSL/TLS và bảo mật mạng, tầm quan trọng của nó cũng như ứng dụng trong thực tế.

I. Giao thức SSL/TLS

1. Tổng quan về SSL?

SSL là chữ viết tắt của **Secure Sockets Layer (Lớp socket bảo mật)**. Một loại bảo mật giúp mã hóa liên lạc giữa website và trình duyệt. Công nghệ này đang **lỗi thời** và được thay thế hoàn toàn bởi **TLS.**

1.1. Tại sao sử dụng SSL?

Ngày nay việc bảo mật thông tin là yếu tố quan trọng để quyết định sự sống còn của một tổ chức, một công ty hay doanh nghiệp. Với sự phát triển nhanh chóng của công nghệ đã mang lại nhiều tiện ích cho người dùng nhưng đồng thời cũng đặt ra một nhu cầu hết sức cấp thiết về sự an toàn và bảo mật. Và SSL chính là giải pháp tốt nhất hiện nay đáp ứng những nhu cầu đó và nó được coi như là "lá chắn cuối cùng" trong bảo mật thương mại điện tử.

Việc truyền các thông tin nhạy cảm trên mạng rất không an toàn vì những vấn đề sau:

1. Bạn không thể luôn luôn chắc rằng bạn đang trao đổi thông tin với đúng đối tượng cần trao đổi.
2. Dữ liệu mạng có thể bị chặn, vì vậy dữ liệu có thể bị 1 đối tượng thứ 3 khác đọc trộm, thường được biết đến như **attacker**.

3. Nếu attacker có thể chặn dữ liệu, attacker có thể sửa đổi dữ liệu trước khi gửi nó đến người nhận.

SSL giải quyết các vấn đề trên. SSL giải quyết vấn đề đầu tiên bằng cách cho phép 1 cách tùy chọn mỗi bên trao đổi có thể chắc chắn về định danh của phía đối tác trong 1 quá trình gọi là **authentication** (xác thực). Một khi các bên đã được xác thực, SSL cung cấp 1 kết nối được mã hóa giữa 2 bên để truyền bảo mật các message. Việc mã hóa trong quá trình trao đổi thông tin giữa 2 bên cung cấp sự riêng tư bí mật, vì vậy mà giải quyết được vấn đề thứ 2. Thuật toán mã hóa được sử dụng với SSL bao gồm hàm băm mã hóa, tương tự như 1 checksum. Nó đảm bảo rằng dữ liệu không bị thay đổi trong quá trình truyền dẫn. Hàm băm mã hóa giải quyết vấn đề thứ 3, tính toàn vẹn dữ liệu.

1.2. Cơ bản về SSL

SSL (Secure Sockets Layer) là tiêu chuẩn của công nghệ bảo mật, truyền thông mã hoá giữa máy chủ Web server và trình duyệt (browser). Tiêu chuẩn này hoạt động và đảm bảo rằng các dữ liệu truyền tải giữa máy chủ và trình duyệt của người dùng đều riêng tư và toàn vẹn. SSL hiện tại cũng là tiêu chuẩn bảo mật cho hàng triệu website trên toàn thế giới, nó bảo vệ dữ liệu truyền đi trên môi trường internet được an toàn.

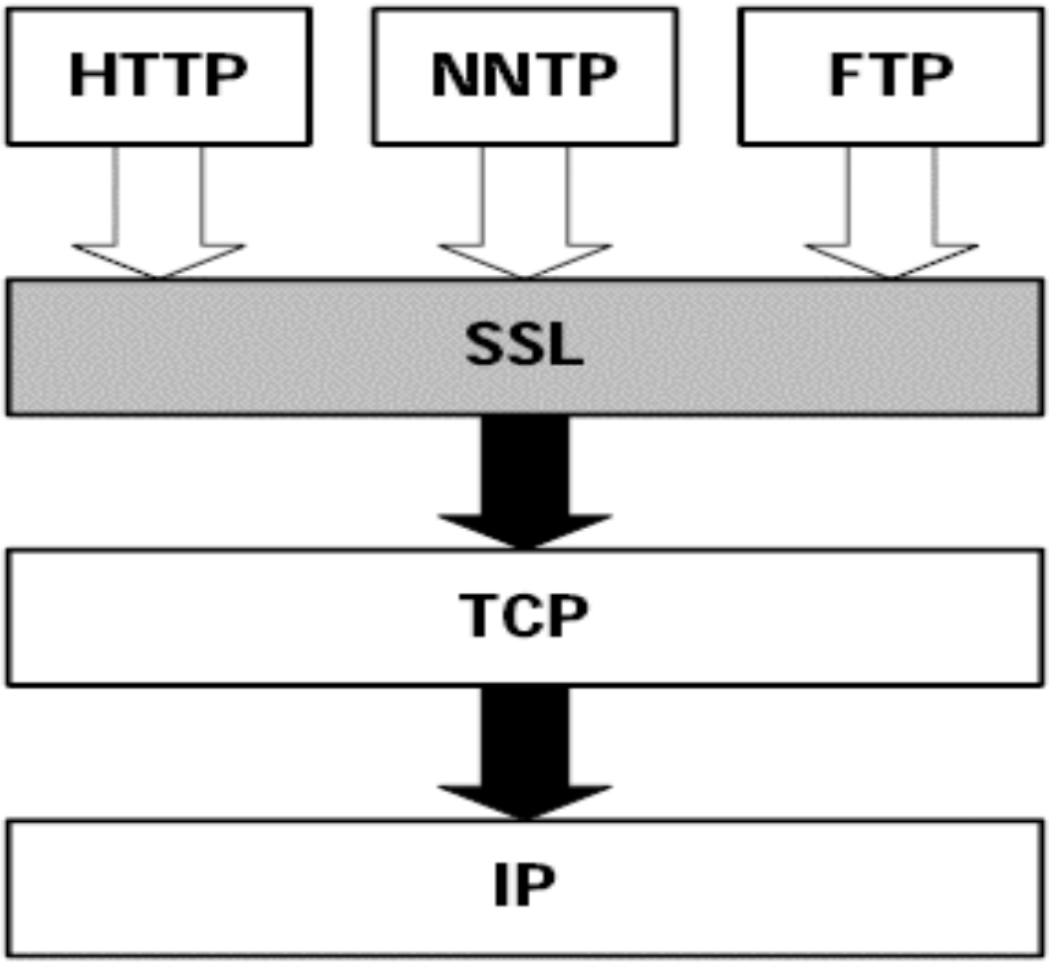
Nói cách khác, việc kết nối giữa một Web browser tới bất kỳ điểm nào trên mạng Internet đi qua rất nhiều các hệ thống độc lập mà không có bất kỳ sự bảo vệ nào với các thông tin trên đường truyền. Không một ai kể cả người sử dụng lẫn Web server có bất kỳ sự kiểm soát nào đối với đường đi của dữ liệu hay có thể kiểm soát được liệu có ai đó thâm nhập vào thông tin trên đường truyền. Để bảo vệ những thông tin mật trên mạng Internet hay bất kỳ mạng TCP/IP nào, SSL (Secure Sockets Layer) đã kết hợp những yếu tố sau để thiết lập được một giao dịch an toàn:

- Xác thực: đảm bảo tính xác thực của trang mà bạn sẽ làm việc ở đầu kia của kết nối. Cũng như vậy, các trang Web cũng cần phải kiểm tra tính xác thực của người sử dụng.
- Mã hoá: đảm bảo thông tin không thể bị truy cập bởi đối tượng thứ ba. Để loại trừ việc nghe trộm những thông tin “nhạy cảm” khi nó được truyền qua Internet, dữ liệu phải được mã hóa để không thể bị đọc được bởi người khác ngoài người gửi và người nhận.
- Toàn vẹn dữ liệu: đảm bảo thông tin không bị sai lệch và nó phải thể hiện chính xác thông tin gốc gửi đến.

SSL được tích hợp sẵn vào các browser và Web server, cho phép người sử dụng làm việc với các trang Web ở chế độ an toàn. Khi Web browser sử dụng kết nối SSL tới server, biểu tượng ổ khóa sẽ xuất hiện trên thanh trạng thái của cửa sổ browser và dòng “http” trong hộp nhập địa chỉ URL sẽ đổi thành “https”. Một phiên giao dịch HTTPS sử dụng cổng 443 thay vì sử dụng cổng 80 như dùng cho HTTP.

Giao thức điều khiển truyền/Giao thức Internet (TCP/IP) chi phối việc vận chuyển và định tuyến dữ liệu qua Internet. Các giao thức khác, chẳng hạn như Giao thức truyền tải siêu văn bản (HTTP), Giao thức truy cập thư mục nhẹ (LDAP) hoặc Giao thức truy cập nhắn tin Internet (IMAP), chạy "trên đầu" TCP/IP theo nghĩa là tất cả đều sử dụng TCP/IP để hỗ trợ các tác vụ ứng dụng điển hình như hiển thị trang web hoặc chạy máy chủ email.

Giao thức SSL chạy trên TCP/IP và bên dưới các giao thức cấp cao hơn như HTTP hoặc IMAP. Nó sử dụng TCP/IP thay mặt cho các giao thức cấp cao hơn và trong quá trình này cho phép máy chủ hỗ trợ SSL tự xác thực với máy khách hỗ trợ SSL, cho phép máy khách tự xác thực với máy chủ và cho phép cả hai máy thiết lập một kết nối được mã hóa.



Hình 1. Vị trí SSL trong mô hình OSI

Các khả năng này giải quyết các mối quan tâm cơ bản về giao tiếp qua Internet và các mạng TCP/IP khác:

- Xác thực máy chủ SSL: cho phép người dùng xác nhận danh tính của máy chủ. Phần mềm máy khách hỗ trợ SSL có thể sử dụng các kỹ thuật tiêu chuẩn về mật mã khóa công khai để kiểm tra xem chứng chỉ và ID công khai của máy chủ có hợp lệ không và đã được cấp bởi cơ quan cấp chứng chỉ (CA) được liệt kê trong danh sách CA đáng tin cậy của khách hàng. Xác nhận này có thể quan trọng nếu người dùng, ví dụ, đang gửi số thẻ tín dụng qua mạng và muốn kiểm tra danh tính của máy chủ nhận.
- Xác thực ứng dụng khách SSL: cho phép máy chủ xác nhận danh tính người dùng. Sử dụng các kỹ thuật tương tự như các kỹ thuật được sử dụng để xác thực máy chủ, phần

mềm máy chủ hỗ trợ SSL có thể kiểm tra xem chứng chỉ và ID công khai của khách hàng có hợp lệ không và đã được cấp bởi cơ quan chứng nhận (CA) được liệt kê trong danh sách CA đáng tin cậy của máy chủ. Xác nhận này có thể quan trọng nếu máy chủ, ví dụ, là một ngân hàng gửi thông tin tài chính bí mật cho khách hàng và muốn kiểm tra danh tính của người nhận.

- Kết nối SSL được mã hóa yêu cầu tất cả thông tin được gửi giữa máy khách và máy chủ phải được mã hóa bằng phần mềm gửi và được giải mã bằng phần mềm nhận, do đó cung cấp mức độ bảo mật cao. Bảo mật rất quan trọng đối với cả hai bên đối với bất kỳ giao dịch cá nhân nào. Ngoài ra, tất cả dữ liệu được gửi qua kết nối SSL được mã hóa được bảo vệ bằng cơ chế phát hiện giả mạo - nghĩa là để tự động xác định liệu dữ liệu có bị thay đổi khi truyền hay không.

Giao thức SSL bao gồm hai giao thức phụ: giao thức ghi SSL và giao thức bắt tay SSL. Giao thức bản ghi SSL xác định định dạng được sử dụng để truyền dữ liệu. Giao thức bắt tay SSL liên quan đến việc sử dụng giao thức bản ghi SSL để trao đổi một loạt tin nhắn giữa máy chủ hỗ trợ SSL và máy khách hỗ trợ SSL khi lần đầu tiên thiết lập kết nối SSL. Việc trao đổi tin nhắn này được thiết kế để tạo điều kiện cho các hành động sau:

- Xác thực máy chủ cho khách hàng.
- Cho phép máy khách và máy chủ chọn các thuật toán mã hóa hoặc mật mã mà cả hai đều hỗ trợ.
- Tùy chọn xác thực ứng dụng khách đến máy chủ.
- Sử dụng các kỹ thuật mã hóa khóa công khai để tạo ra các bí mật được chia sẻ.
- Thiết lập kết nối SSL được mã hóa.

Điểm cơ bản của SSL là được thiết kế độc lập với tầng ứng dụng để đảm bảo tính bí mật, an toàn và chống giả mạo luồng thông tin qua Internet giữa hai ứng dụng bất kì, thí dụ như webserver và các trình duyệt khách (browsers). Toàn bộ cơ chế và hệ thống thuật toán mã hóa sử dụng trong SSL được phổ biến công khai, trừ khóa phiên (session key) được sinh ra tại thời điểm trao đổi giữa hai ứng dụng là ngẫu nhiên và bí mật đối với người quan sát trên mạng máy tính. Ngoài ra, giao thức SSL còn đòi hỏi ứng dụng chủ phải được chứng thực bởi một đối tượng lớp thứ ba (CA) thông qua giấy chứng thực điện tử (digital certificate dựa trên mật mã công khai (ví dụ RSA).

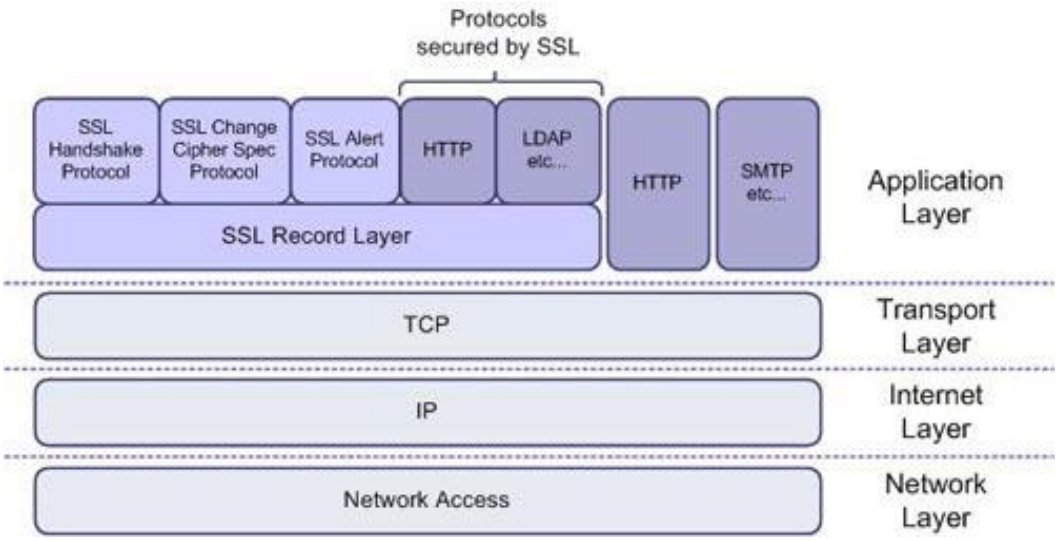
Các phiên bản:

- SSLv2: đây là phiên bản đầu tiên của giao thức SSL do Netscape Corporation thiết kế.
- SSLv3: đây là phiên bản SSL version 3.0 do Netscape Corporation thiết kế, đã có trợ giúp chain certificate (chứng chỉ nhóm) và được hỗ trợ cho tất cả các trình duyệt phổ thông.
- TLSv1: giao thức Transport Layer Security version 1.0 dựa trên cơ sở của SSLv3, được thiết kế bởi IETF.

2. Cấu trúc của giao thức SSL

Giao thức SSL gồm hai tầng. Tầng thấp nhất được đặt trên một số giao thức vận tải tin cậy (ví dụ TCP), là tầng SSL Record Protocol. SSL Record Protocol được sử dụng để đóng gói

một vài giao thức ở mức cao hơn. Một trong những giao thức được đóng gói là SSL Handshake Protocol.



Hình 2. Cấu trúc của SSL và giao thức SSL

Cấu trúc của SSL và giao thức SSL tương ứng được minh họa trong hình2. Theo hình minh họa trên, SSL ám chỉ một lớp (bảo mật) trung gian giữa lớp vận chuyển (Transport Layer) và lớp ứng dụng (Application Layer). SSL được xếp lớp lên trên một dịch vụ vận chuyển định hướng nối kết và đáng tin cậy, chẳng hạn như được cung cấp bởi TCP. Về khả năng, nó có thể cung cấp các dịch vụ bảo mật cho các giao thức ứng dụng tùy ý dựa vào TCP chứ không chỉ HTTP. Thực tế, một ưu điểm chính của các giao thức bảo mật lớp vận chuyển (Transport layer) nói chung và giao thức SSL nói riêng là chúng độc lập với ứng dụng theo nghĩa là chúng có thể được sử dụng để bảo vệ bất kỳ giao thức ứng dụng được xếp lớp lên trên TCP một cách trong suốt.

Giao thức SSL cung cấp giao thức bảo mật truyền thông có 3 đặc điểm nổi bật:

- Các bên giao tiếp (nghĩa là client và server) có thể xác thực nhau bằng cách sử dụng mật mã khóa chung
- Sự bí mật của lưu lượng dữ liệu được bảo vệ vì nối kết được mã hóa trong suốt sau khi một sự thiết lập quan hệ ban đầu và sự thương lượng khóa session đã xảy ra.
- Tính xác thực và tính toàn vẹn của lưu lượng dữ liệu cũng được bảo vệ vì các thông báo được xác thực và được kiểm tra tính toàn vẹn một cách trong suốt bằng cách sử dụng MAC.
-

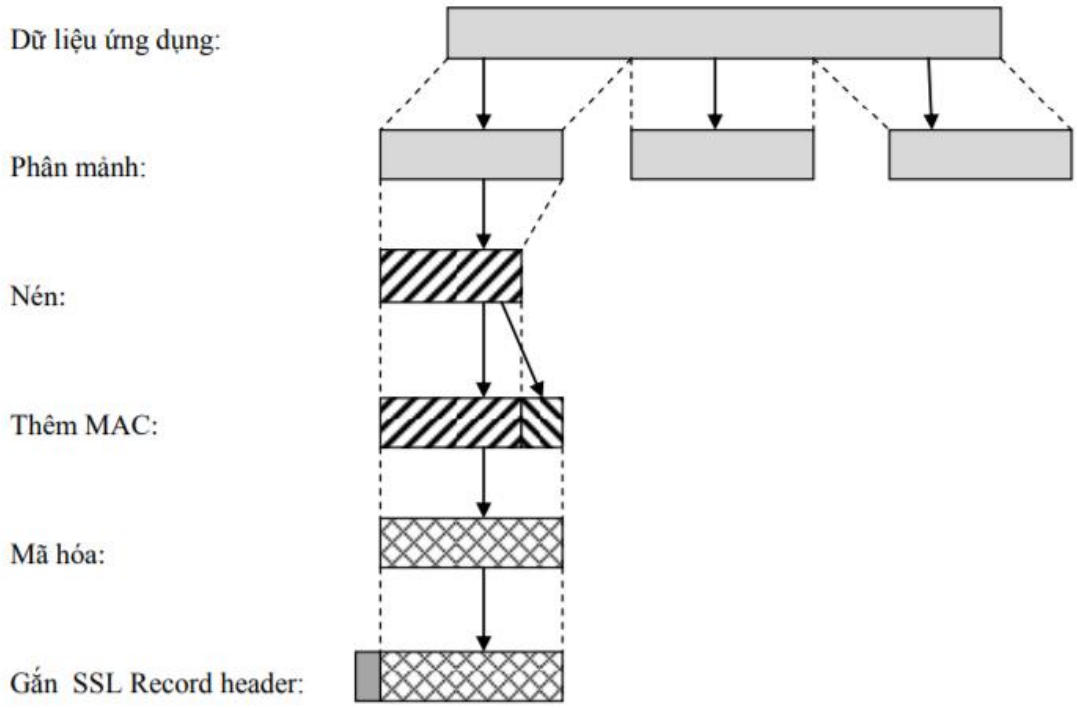
2.1. SSL Record Protocol

SSL Record Protocol cung cấp 2 dịch vụ cho kết nối SSL:

- Confidentiality (tính cẩn mật): Handshake Protocol định nghĩa 1 khóa bí mật được chia sẻ, khóa này được sử dụng cho mã hóa quy ước các dữ liệu SSL.

- Message integrity (tính toàn vẹn thông điệp): Handshake Protocol cũng định nghĩa 1 khóa bí mật được chia sẻ, khóa này được sử dụng để hình thành MAC (mã xác thực message).

Hình sau chỉ ra toàn bộ hoạt động của SSL Record Protocol. SSL Record Protocol nhận 1 message ứng dụng sắp được truyền đi, phân mảnh dữ liệu thành nhiều block, nén dữ liệu 1 cách tùy chọn, áp dụng vào 1 MAC, mã hóa, thêm vào header, và truyền khối kết quả thu được trong 1 segment TCP. Dữ liệu nhận được được giải mã, kiểm tra, giải nén, sắp xếp lại và phân phối đến người sử dụng ở lớp cao hơn.



Hình 3. Hoạt động của SSL Record Protocol

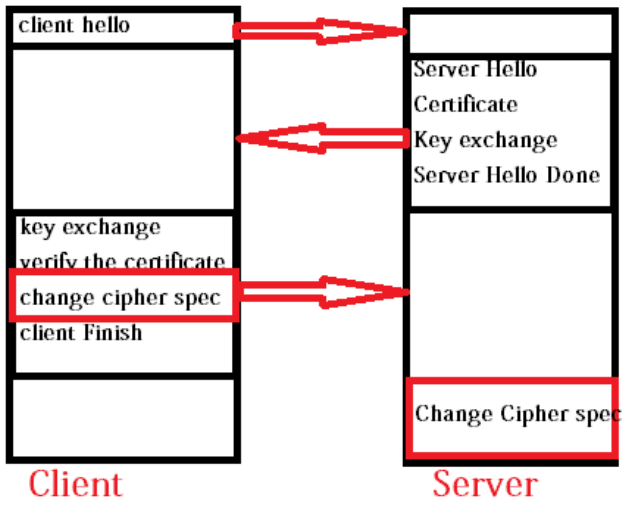
2.2. SSL Handshake Protocol

SSL Handshake Protocol là giao thức con SSL chính được xếp lớp trên SSL Record Protocol. Kết quả, các thông báo thiết lập quan hệ SSL được cung cấp cho lớp bản ghi SSL nơi chúng được bao bọc trong một hoặc nhiều bản ghi SSL vốn được xử lý và được chuyển như được xác định bởi phương pháp nén và thông số mật mã của session SSL hiện hành và các khóa mật mã của nối kết SSL tương ứng. Mục đích của SSL Handshake Protocol là yêu cầu một client và server thiết lập và duy trì thông tin trạng thái vốn được sử dụng để bảo vệ các cuộc liên lạc. Cụ thể hơn, giao thức phải yêu cầu client và server chấp thuận một phiên bản giao thức SSL chung, chọn phương thức nén và thông số mật mã, tùy ý xác thực nhau và tạo một khóa mật chính mà từ đó các khóa session khác nhau dành cho việc xác thực và mã hóa thông báo có thể được dẫn xuất từ đó.

Giao thức này cho phép server và client chứng thực với nhau và thương lượng cơ chế mã hóa, thuật toán MAC và khóa mật mã được sử dụng để bảo vệ dữ liệu được gửi trong SSL record.

Giao thức SSL Handshake thường được sử dụng trước khi dữ liệu của ứng dụng được truyền đi.

2.3. Giao thức SSL Change Cipher Spec Protocol



Hình 4. Giao thức SSL Change Cipher Spec Protocol

Giao thức SSL Change Cipher Spec là giao thức đơn giản nhất trong ba giao thức đặc trưng của SSL.

Giao thức này bao gồm một message đơn 1byte giá trị là 1. Mục đích chính của message này là sinh ra trạng thái tiếp theo để gán vào trạng thái hiện tại, và trạng thái hiện tại cập nhật lại bộ mã hóa để sử dụng trên kết nối này

2.4. Giao thức SSL Alert

Các hệ thống sử dụng giao thức Alert để báo hiệu một lỗi hoặc một cảnh báo xảy ra trong quá trình truyền thông giữa hai bên, và SSL gán cho kiểu giao thức của Alert là 21, Alert Protocol cũng giống như tất cả các giao thức SSL khác, sử dụng Record Layer định dạng thông điệp của nó. Các thông điệp alert truyền tải các thông báo lỗi hay cảnh báo trong quá trình thiết lập cũng như trao đổi dữ liệu của một phiên liên lạc.

Một số thông báo lỗi:

- bad_record_mac: MAC không chính xác
- unsupported_certificate: dạng certificate nhận được thì không hỗ trợ.
- certificate_revoked: certificate đã bị thu hồi bởi nhà cung cấp.
- certificate_expired: certificate đã hết hạn đăng ký.

Tóm lại, SSL Alert Protocol được sử dụng để chuyển các cảnh báo thông qua SSL Record Protocol. Mỗi cảnh báo gồm 2 phần, một mức cảnh báo và một mô tả cảnh báo.

SSL Handshake Protocol là giao thức con SSL chính được sử dụng để hỗ trợ xác thực client và server và để trao đổi một khóa session. Do đó SSL Handshake Protocol trình bày tổng quan và được thảo luận trong phần tiếp theo.

Sau cùng, SSL ChangeCipherSpec Protocol được sử dụng để thay đổi giữa một thông số mật mã này và một thông số mật mã khác. Mặc dù thông số mật mã thường được thay đổi ở cuối

một sự thiết lập quan hệ SSL, nhưng nó cũng có thể được thay đổi vào bất kỳ thời điểm sau đó.

Ngoài những giao thức con SSL này, một SSL Application Data Protocol được sử dụng để chuyển trực tiếp dữ liệu ứng dụng đến SSL Record Protocol.

II. Các cơ chế bảo mật trong SSL/TLS

Giao thức SSL hỗ trợ rất nhiều hệ mã hoá sử dụng cho các hoạt động chứng thực server và client, cho quá trình truyền thống chứng chỉ số và trong quá trình thành lập khoá phiên, Client và server có thể có nhiều bộ mã hoá khác nhau, tuỳ thuộc vào phiên bản SSL hỗ trợ, các chính sách công ty chấp nhận các hệ mã hoá, và các hạn chế của chính phủ trong việc sử dụng các phần mềm hỗ trợ SSL.

1. Hai loại mật mã

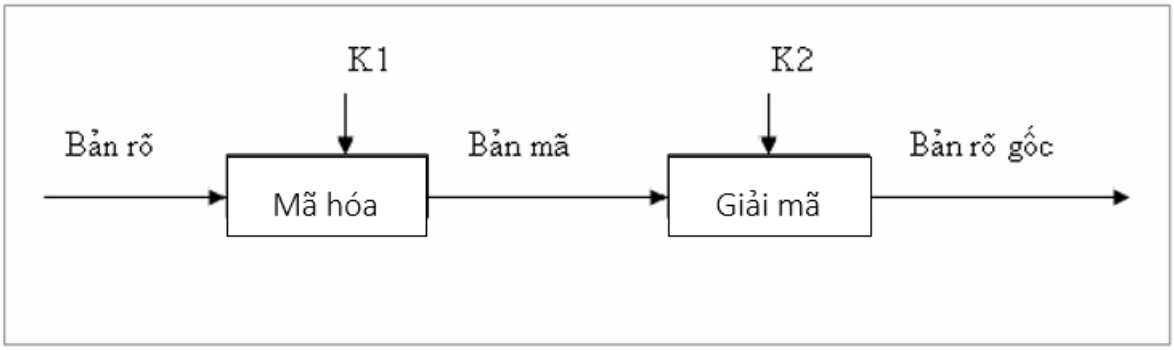
Đối với tất cả các ý định và mục đích, khi chúng ta thảo luận mật mã khi chúng liên quan đặc biệt đến mã hóa SSL, có hai loại thuật toán: đối xứng và không đối xứng. Điều này thực sự đi kèm với loại mã hóa bạn sẽ thực hiện, một lần nữa, đối xứng hoặc không đối xứng.

1.1. Mã hóa đối xứng

Hình 5. Mô hình mã hóa đối xứng

Mã hóa đối xứng liên quan đến hai bên giống nhau. Cả hai bên có thể thực hiện cả hai chức năng: mã hóa và giải mã. Bạn thấy điều này trong khi kết nối web được mã hóa giữa trình duyệt và máy chủ. Sau khi chứng chỉ SSL đã được xác thực và bắt tay SSL hoàn tất, trình duyệt và trao đổi máy chủ "khóa phiên" đối xứng cho phép chúng giao tiếp an toàn trong suốt thời gian truy cập. Trong khi các khóa phiên này đang hoạt động, chúng đang sử dụng mật mã đối xứng.

1.2. Mã hóa bất đối xứng



Hình 6. Mô hình mã hóa bất đối xứng

Ngược lại, với mã hóa bất đối xứng, bạn đang nói về các khóa khác nhau với các khả năng khác nhau. Ví dụ rõ ràng nhất về điều này là cặp khóa công khai / riêng được sử dụng trong quá trình bắt tay SSL. Trong trường hợp này, một khóa mã hóa (K1) và khóa khác giải mã (K2). Loại mã hóa này yêu cầu một loại mật mã khác – một thuật toán bất đối xứng.

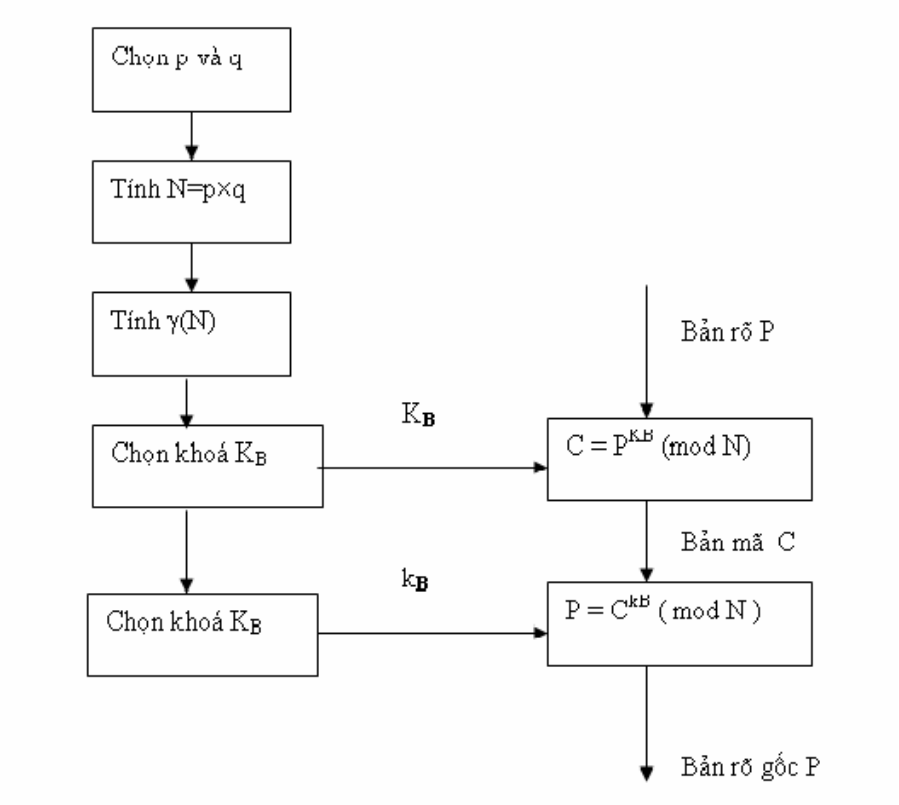
2. Mật mã chung

Có rất nhiều thuật toán mã hóa khác nhau thường được sử dụng trong mã hóa kết hợp với nhau. Đó là bởi vì, cụ thể vì nó liên quan đến SSL, bạn không chỉ sử dụng một thuật toán đơn lẻ mà là một tập hợp các thuật toán được nhóm lại với nhau trong những gì được gọi là "Bộ mã hóa" (Cipher Suite).

Bây giờ chúng ta đã hiểu được hai loại thuật toán – đối xứng và không đối xứng – chúng ta có thể xem xét một số thuật toán mã hóa khác nhau và các chức năng mà chúng phục vụ – sau đó chúng ta sẽ nói về xây dựng một bộ mã hóa.

3. Trao đổi khóa

3.1. RSA



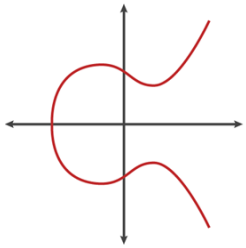
Hình 7. Thuật toán RSA

RSA được đặt theo tên của những người đã tạo ra nó: Rivest, Shamir và Adleman. Đây là một hệ thống mã hóa không đối xứng khá phổ biến, sử dụng số nguyên tố và có nhiều ứng dụng.

3.2. Diffie-Hellman

Được đặt tên theo Whitfield Diffie và Martin Hellman, đây là một giao thức khóa công khai được sử dụng chủ yếu để trao đổi các khóa mật mã trên các kênh công cộng. Trước các phương pháp như DH, các khóa phải được truyền dưới dạng vật lý.

3.3. Elliptic Curve Diffie-Hellman



Hình 8. Đồ thị Elliptic Curve Diffie-Hellman

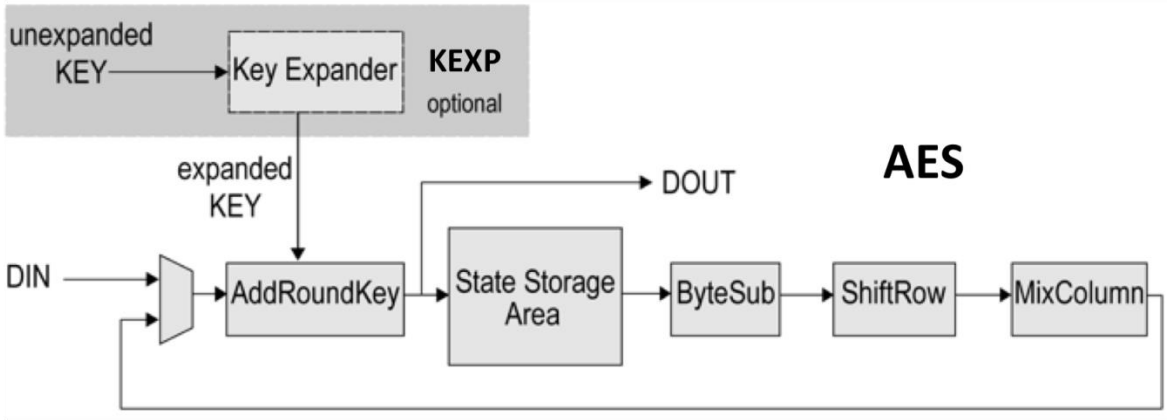
Một giao thức thỏa thuận khóa, cung cấp cho hai bên với cặp khóa công khai để thiết lập bí mật được chia sẻ (được sử dụng trực tiếp như một khóa) một cách an toàn trên kênh công khai.

3.4. PSK

Thường được viết dưới dạng TLS-PSK, đây là một thuật toán mã hóa cung cấp thông tin liên lạc an toàn dựa trên các khóa đối xứng được chia sẻ trước giữa các bên.

4. Mã hóa mật mã (Encryption Ciphers)

4.1. AES



Hình 9. Thuật toán AES

Chuẩn mã hóa nâng cao, còn gọi là Rijndael, là một mật mã mã hóa được NIST chấp thuận với kích thước khối 128bit và các khóa đối xứng có độ dài 128, 192 hoặc 256 bit.

4.2. Camellia

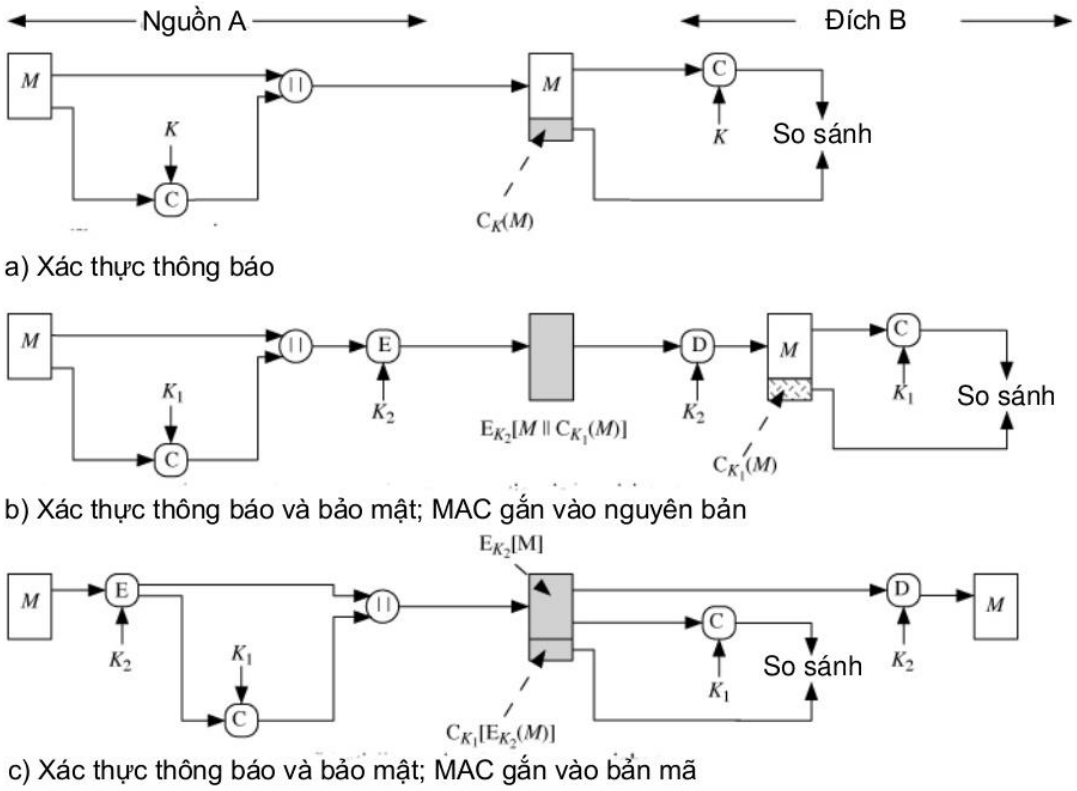
Một mật mã khối khóa đối xứng với các khả năng và kích thước khóa tương tự như AES. Nó được phát triển tại Nhật Bản bởi NTT và Mitsubishi và được chấp thuận bởi ISO / IEC, EU và dự án CRYPTREC của Nhật Bản.

4.3. ARIA

Một mật mã khối khác tương tự như AES, ARIA được phát triển bởi một nhóm các nhà nghiên cứu ở Hàn Quốc vào năm 2003.

5. Tính toàn vẹn / xác thực dữ liệu

5.1. Mã xác thực thư dựa trên hàm băm (HMAC)



Hình 10. Mô hình xác thực HMAC

Đây là một loại xác thực thông điệp sử dụng các mật mã băm để xác thực cả một thông điệp và đảm bảo tính toàn vẹn dữ liệu, hãy nghĩ đến SHA-256.

5.2. Mã hóa được xác thực

AE hoặc AEAD đảm bảo tính bảo mật, tính toàn vẹn và xác thực về dữ liệu trong một giao diện lập trình đơn. Thường được sử dụng kết hợp với mật mã khối.

Rõ ràng, đây là một danh sách không đầy đủ, có hàng tá mật mã khác. Nhưng điều này ít nhất sẽ cung cấp cho bạn thêm một số bối cảnh khi chúng ta bắt đầu thảo luận về các bộ mã hóa trong phần tiếp theo.

6. Bộ mật mã là gì?

A Cipher Suite là một sự kết hợp của các thuật toán được sử dụng để thương lượng các thiết lập bảo mật trong quá trình bắt tay SSL / TLS. Sau khi ClientHello và ServerHello tin nhắn được trao đổi, khách hàng sẽ gửi một danh sách ưu tiên các bộ mã hóa mà nó hỗ trợ. Máy chủ sau đó sẽ phản hồi với bộ mã hóa mà nó đã chọn từ danh sách.

Các bộ mã hóa được đặt tên là các kết hợp của:

- Thuật toán trao đổi khóa (RSA, DH, ECDH, PSK)
- Thuật toán xác thực (RSA, DSA)

- Thuật toán mã hóa hàng loạt (AES, Camellia, ARIA)
- Thuật toán mã xác thực thư (SHA-256)

Ví dụ 1 bộ mã hóa:

```
TLS _ECDHE_ RSA _ WITH_AES_128_GCM _ SHA256
```

III. Cơ chế xác thực SSL/TLS

SSL Handshake Protocol là một phần quan trọng của SSL, nó cung cấp ba dịch vụ cho các kết nối SSL giữa client và server. Handshake Protocol cho phép client/server thống nhất về phiên bản giao thức, xác thực mỗi bên bằng cách thi hành một MAC và thoả thuận về một thuật toán mã hoá và các khoá lập mã cho việc bảo vệ các dữ liệu gửi đi trong một SSL record trước khi giao thức ứng dụng truyền đi hay nhận được byte dữ liệu đầu tiên.

Quá trình handshake SSL/TLS cho phép máy client và máy server SSL/TLS thiết lập các khóa bí mật mà chúng sử dụng để giao tiếp với nhau.
Các bước để máy client và máy chủ SSl/TLS liên lạc với nhau:

a) Thống nhất về phiên bản của giao thức sẽ sử dụng.
b) Chọn thuật toán mật mã.
c) Xác thực lẫn nhau bằng cách trao đổi và xác nhận chứng thư số.
d) Sử dụng các kỹ thuật mã hóa bất đối xứng để tạo khóa bí mật dùng chung để tránh việc phân phối khóa. Sau đó SSL/TLS sử dụng khóa chung để mã hóa đối xứng các thông điệp. Việc sử dụng mã hóa đối xứng nhanh hơn mã hóa bất đối xứng.

1. Các bước handshake trong SSL:

Bước 1. Toàn bộ kết nối / bắt tay bắt đầu với máy khách gửi một thông điệp “hello client” đến máy chủ. Thông báo này bao gồm các thông tin mật mã như giao thức và CipherSuites được hỗ trợ. Nó cũng bao gồm một giá trị ngẫu nhiên hoặc chuỗi byte ngẫu nhiên.

Bước 2. Để trả lời thông điệp “hello client” của máy khách, máy chủ phản hồi với thông báo “hello server”. Thông điệp này bao gồm CipherSuite mà máy chủ đã chọn trong số những người được cung cấp bởi client. Máy chủ cũng gửi chứng chỉ của nó cùng với ID phiên và một giá trị ngẫu nhiên khác.

Bước 3. Máy khách SSL/TLS xác minh chứng thư số của máy chủ

Bước 4. Máy khách SSL/TLS gửi chuỗi byte ngẫu nhiên cho phép cả máy khách và máy chủ tính toán khóa bí mật sử dụng để mã hóa dữ liệu thông điệp tiếp theo. Chuỗi byte ngẫu nhiên được mã hóa bằng khóa chung của máy chủ.

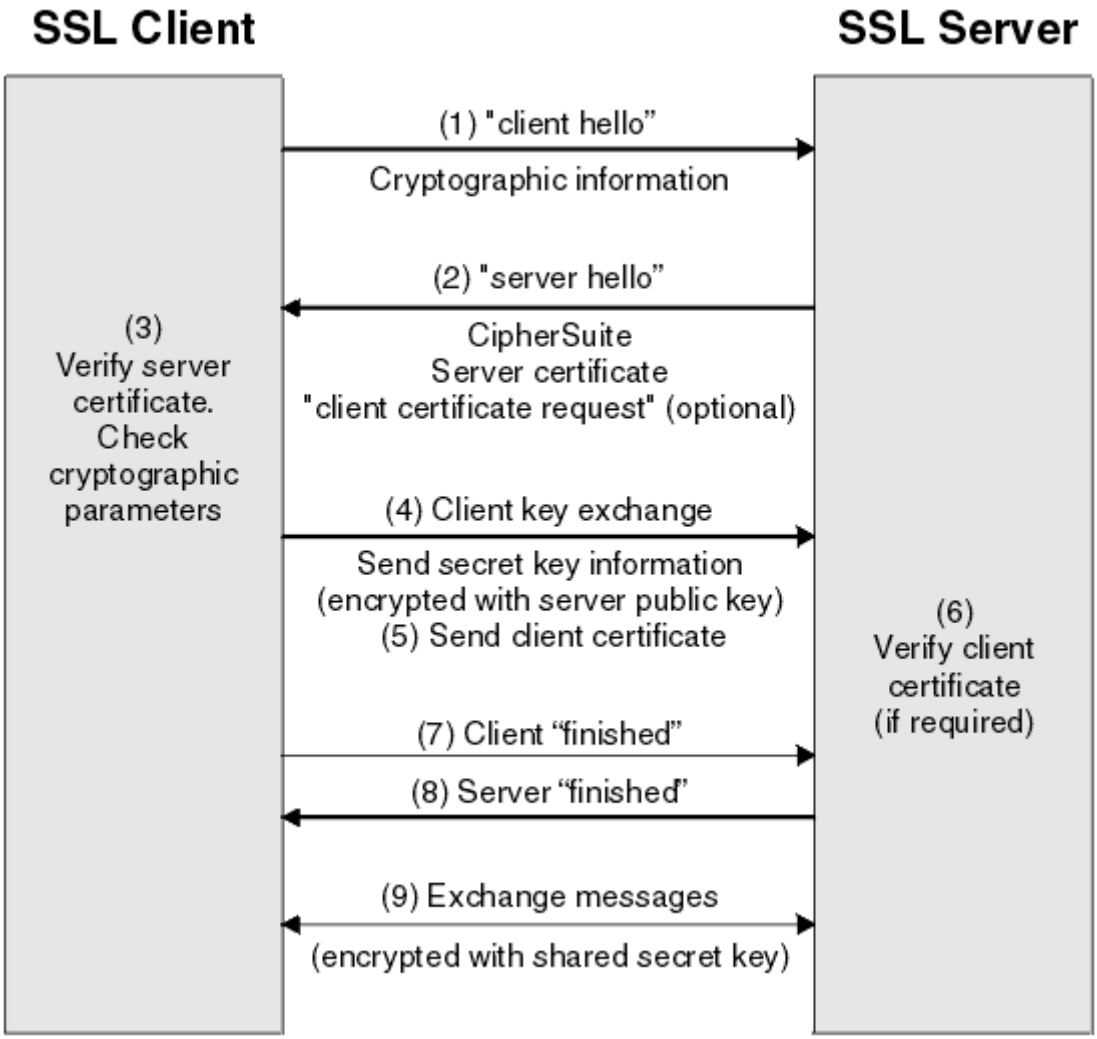
Bước 5. Nếu máy chủ SSL/TLS gửi yêu cầu xác thực client, máy khách sẽ gửi một chuỗi byte ngẫu nhiên được mã hóa bằng khóa riêng của máy khách, cùng với chứng thư số của máy khách hoặc cảnh báo không có chứng thư số. Đây chỉ là một cảnh báo, nhưng trong một số trường hợp không thể xác thực client, quá trình handshake sẽ là không thành công nếu xác thực client là bắt buộc.

Bước 6. Máy chủ SSL/TLS xác minh chứng chỉ của máy khách.

Bước 7. Máy khách gửi cho máy chủ một thông điệp "finished", được mã hóa bằng khóa bí mật để chỉ ra rằng máy khách đã hoàn thành nhiệm vụ trong quá trình handshake.

Bước 8. Máy chủ SSL/TLS gửi cho client một thông điệp "finished", được mã hóa bằng khóa bí mật để chỉ ra rằng máy chủ đã hoàn thành nhiệm vụ trong quá trình handshake.

Bước 9. Trong thời lượng của phiên SSL/TLS, máy chủ và máy khách hiện có thể trao đổi các thông điệp được mã hóa đối xứng với khóa bí mật chung.



Hình 11. Mô hình handshake trong SSL/TLS

2. Cách SSL/TLS kiểm tra tính xác thực

Để xác thực máy chủ, máy khách sử dụng khóa công khai của máy chủ để mã hóa dữ liệu sử dụng để tính toán khóa bí mật. Máy chủ chỉ có thể tạo khóa bí mật khi nó giải mã đúng dữ liệu đó bằng khóa riêng.

Để xác thực client, máy chủ sử dụng khóa công khai trong chứng chỉ của client để giải mã dữ liệu được gửi từ client trong bước 5 của quá trình bắt tay. Việc trao đổi thông điệp "finished" được mã hóa bằng khóa bí mật xác nhận rằng quá trình xác thực đã hoàn tất.

Nếu bất kỳ bước xác thực nào thất bại, quá trình handshake sẽ không thành công và kết thúc phiên.

Việc trao đổi chứng thư số trong quá trình handshake SSL/TLS là một phần của quy trình xác thực. Cần có các chứng chỉ như: CA X cấp chứng chỉ cho máy khách SSL/TLS, CA Y cấp chứng chỉ cho máy chủ SSL/TLS:

Nếu chỉ cần xác thức máy chủ, máy chủ cần:
- Chứng chỉ được cấp cho máy chủ CA Y
- Khóa riêng của máy chủ

Và máy khách cần:
- Chứng chỉ CA cho CA Y

Nếu máy chủ SSL/TLS yêu cầu xác thực máy khách, máy chủ sẽ xác minh danh tính của client bằng cách xác minh chứng thư số của client bằng khóa công khai do CA đã cấp cho client, trong trường hợp này là CA X.

Đối với cả xác thực máy chủ và máy khách, máy chủ cần:
- Chứng chỉ được cấp cho máy chủ CA Y
- Khóa riêng của máy chủ
- Chứng chỉ CA cho CA X

và máy khách cần:
- Chứng chỉ được cấp cho client CA X
- Khóa riêng của client
- Chứng chỉ CA cho CA Y

Mã hoá kết nối: Tất cả các thông tin trao đổi giữa client và server được mã hoá trên đường truyền nhằm nâng cao khả năng bảo mật. Điều này rất quan trọng đối với cả hai bên khi có các giao dịch mang tính riêng tư. Ngoài ra, tất cả các dữ liệu được gửi đi trên một kết nối SSL đã được mã hoá còn được bảo vệ nhờ cơ chế tự động phát hiện các xáo trộn, thay đổi trong dữ liệu. (đó là các thuật toán băm – hash algorithm).

3. Quá trình xác thực chứng chỉ

Trong bước 3 và bước 6 của quá trình handshake, máy khách xác minh chứng chỉ của máy chủ và ngược lại. Quá trình xác thực chứng chỉ bao gồm:

a) Kiểm tra chữ ký số.
b) Chuỗi chứng chỉ được kiểm tra; nên có chứng chỉ CA trung gian.
c) Thời hạn sử dụng và ngày kích hoạt và thời gian hiệu lực được kiểm tra.
d) Trạng thái thu hồi của chứng chỉ được kiểm tra

4. Đặt lại khóa bí mật

Trong quá trình handshake SSL/TLS, một khóa bí mật được tạo để mã hóa dữ liệu giữa máy khách và máy chủ. Khóa bí mật được tạo từ văn bản ngẫu nhiên, được gửi như một phần của handshake và sử dụng để mã hóa, chuyển đổi văn bản rõ thành bản mật được và bản mật thành bản rõ.

Khóa bí mật cũng được sử dụng trong thuật toán MAC (Mã xác thực tin nhắn), được sử dụng để xác định xem tin nhắn có bị thay đổi hay không.

Nếu khóa bí mật bị lộ, bản rõ của thông điệp có thể được giải mã từ bản mật, hoặc khiến thông điệp thay đổi mà không bị phát hiện. Ngay cả đối với một thuật toán phức tạp, bản rõ có thể bị lộ bằng cách áp dụng mọi phép biến đổi toán học có thể cho bản mật. Để giảm thiểu lượng dữ liệu có thể được giải mã hoặc thay đổi nếu khóa bí mật bị hỏng, khóa bí mật có thể được đặt lại theo định kỳ. Khi khóa bí mật đã được đặt lại, khóa bí mật cũ không thể sử dụng để giải mã dữ liệu được mã hóa bằng khóa bí mật mới.

IV. Ứng dụng của SSL/TLS

- Đóng gói các giao thức ví dụ như HTTP, FTP, SMTP, NNTP và XMPP
- Cho phép trao đổi riêng tư trên mạng
- Cho phép các ứng dụng client-server giao tiếp với nhau an toàn

Ngày nay, SSL được sử dụng rộng rãi như là nền tảng bảo mật cho:

- Các truy cập vào những ứng dụng có yêu cầu bảo mật cao như Oracle, PeopleSoft, hay Sicbel từ bất kỳ một Web browser nào.
- Các Website bán hàng qua mạng đang được thanh toán bằng thẻ tín dụng.
- Các trung học tập trực tuyến, tài chính ngân hàng, giao dịch chứng khoán và dịch vụ thanh toán hoá đơn.
- Các nhà cung cấp dịch vụ y tế khi muốn chia sẻ các thông tin nghiên cứu riêng tư của mình.
- Các công ty bảo hiểm cung cấp các dịch vụ trục tuyến cho các đại lý và các khách hàng.
- Các trung cung cấp dịch vụ thương mại điện tử trực tiếp.
- Các dịch vụ của chính phủ đòi hỏi sự bảo mật thông tin như thuế, an ninh xã hội, quân đội và các thông tin về sức khoẻ.
- Các trang du lịch mà có thể đặt phòng và vé trực tuyến.
- Các mạng nội bộ có yêu cầu bảo mật các thông tin quan trọng.

Các trung mở rộng phục vụ cho các truy cập an toàn vào các nguồn thông tin của công ty từ phía các đối tác, các đại lý cung cấp và các khách hàng chính.

V. Demo

Dùng WireShark bắt gói tin khi người dùng đăng nhập vào một trang HTTP và một trang HTTPS (HTTP có sử dụng giao thức bảo mật SSL/TLS) và chỉ ra điểm khác biệt.

SSL/TLS và các ứng dụng

1. **Bắt gói tin không có SSL/TLS**

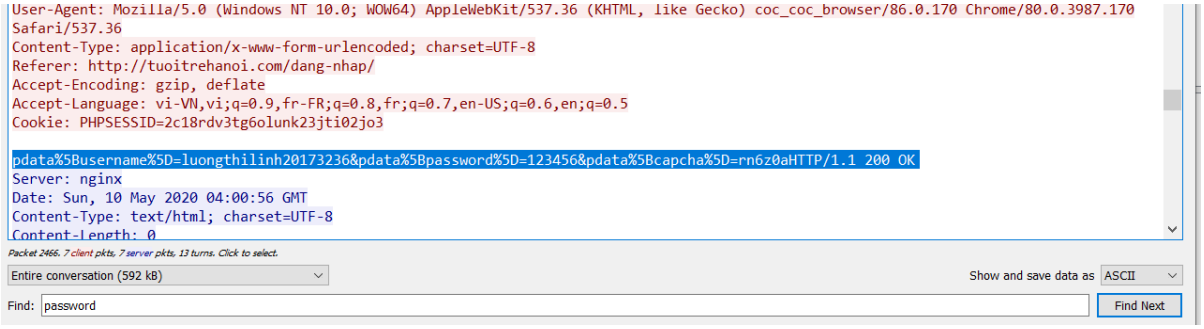
Bắt gói tin qua trang http://tuoitrehanoi.com

Bước 1: Mở WireShark và tiến hành bát gói tin.

Bước 2: Đăng nhập vào trang http://tuoitrehanoi.com

Bước 3: Dừng bắt gói tin

Bước 4: Lọc gói tin và phân tích

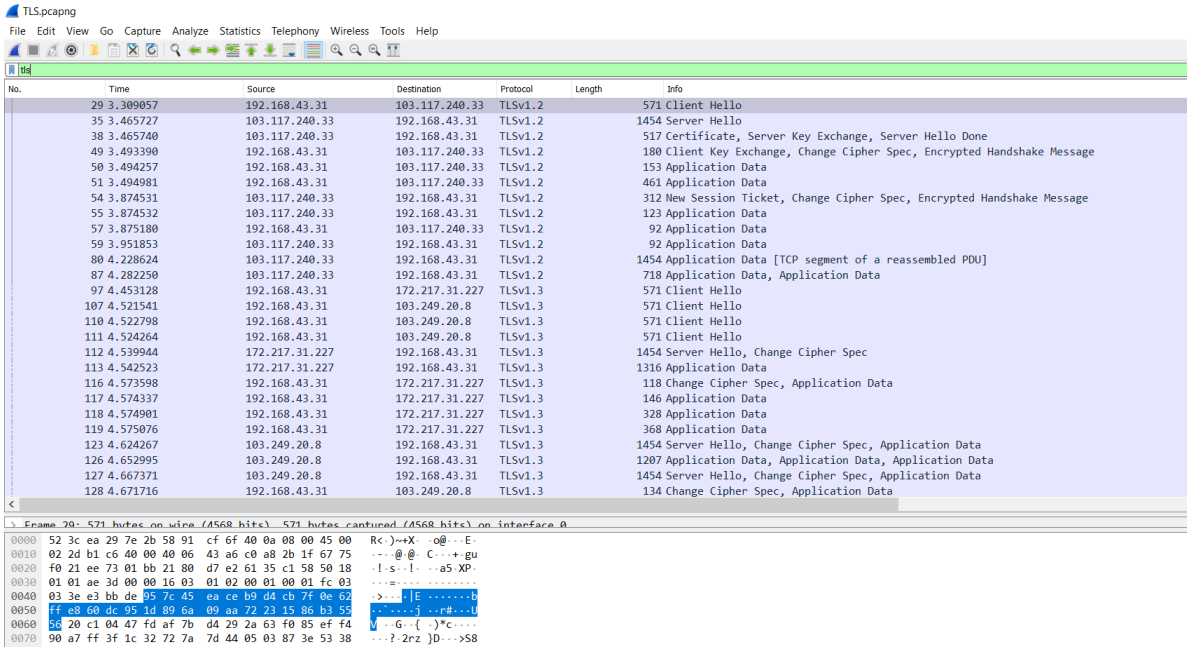


Bắt được gói tin và đọc được thông tin về tài khoản như username và password.

2. **Bắt gói tin có SSL/TLS**

Bắt gói tin qua trang https://shopee.com

Các bước bắt gói tin và phân tích giống như phần trên.



Bắt được gói tin nhưng thông tin đã được mã hóa, không thể đọc được các thông tin username, password như ở trang web HTTP mà chỉ nhận được các message xác thực trong quá trình handshake giữa client và server.

3. Lời kết

Từ phần demo trên, ta nhận thấy HTTPS an toàn hơn so với HTTP rất nhiều trong việc mã hóa dữ liệu, bảo mật thông tin cá nhân. Tuy nhiên ưu điểm của HTTP là tốc độ phản hồi của website truy cập nhanh hơn HTTPS rất nhiều và được sử dụng cho các trang tin tức cần thông tin nhanh, còn phải nhập dữ liệu như tài khỏa ngân hàng, email cá nhân thì nên sử dụng HTTPS. Ngoài ra chúng ta cũng dễ dàng nhận biết với biểu tượng khóa ở thanh địa chỉ để phân biệt website đó có sử dụng HTTPs hay không.

Tài liệu tham khảo

[1] William Stallings, Cryptography and Network Security 5th edition

[2] Mark Stamp, Information Security: Principles and Practice, 2007

[3] https://www.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q009940_.htm

[4] https://www.digistar.vn/cach-thuc-hoat-dong-cua-ssl

[5] http://doc.edu.vn/tai-lieu/bao-cao-tim-hieu-giao-thuc-ssl-tls-cach-tan-cong-va-phong-chong-7627/?fbclid=IwAR161w8PbtTm2z3EacOaiADNv5FOp1g-fVMP4SRPGgXlHGH7WhWydrDX1B8

[6] https://www.slideshare.net/conglongit90/giao-thc-bo-mt-ssl?from_action=save

[7] Stephen Thomas, SSL &TLS Essentials, Securing the Web, 2002

[8] https://developer.mozilla.org/en-US/docs/Archive/Security/Introduction_to_SSL#The_SSL_Protocol