# Các dạng BT An toàn an ninh mạng

21020055 - Trần Thùy Dung

May 27

## 1 Tiền đề

Mỗi năm, thầy Thọ lại ra một kiểu đề khác nhau, yêu cầu sự động não và hiểu bản chất vấn đề của sinh viên. Do đó những dạng dưới đây chỉ mang tính chất luyện tập.

## 2 An toàn mức giao vận

### 2.1 Handshake SSL Protocol

Dạng bài được cho sẽ yêu cầu sinh viên vẽ thông báo 4 giai đoạn của giao thức Handshake, rồi chỉ ra trong những thông báo tùy chọn và `client_key_exchange` có những tham số nào. Để chỉ ra được, cần xác định rõ giải thuật được áp dụng, và kiểu khóa công khai server/client sở hữu.

Bảng 1: Tham số các thông báo

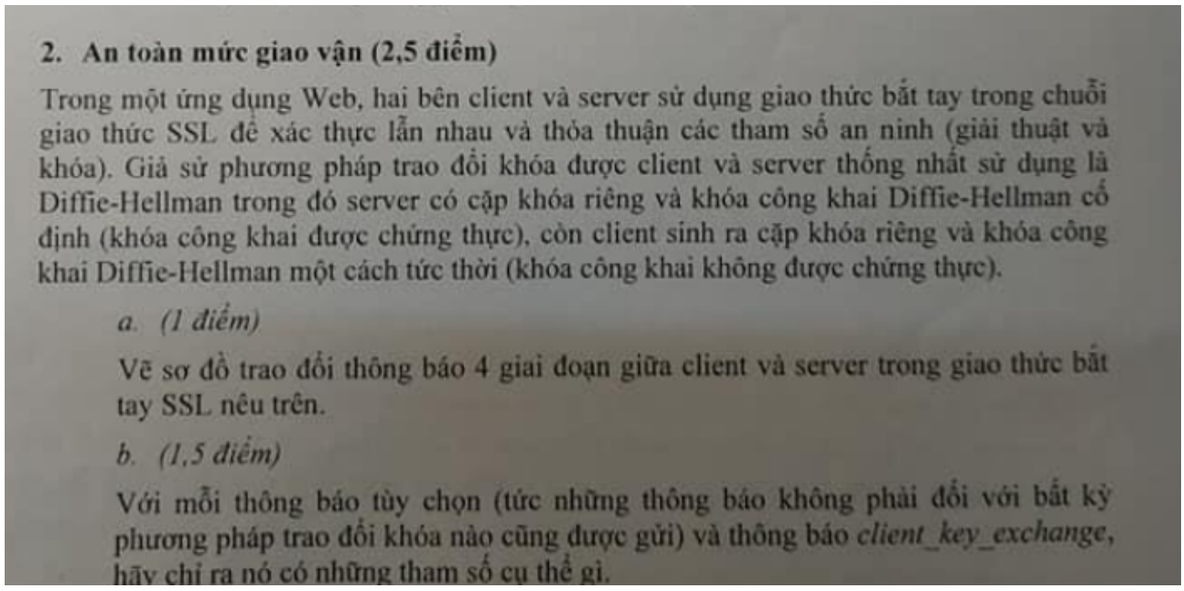
| Thông báo | Tham số | Mô tả |
|---|---|---|
| `server_key_exchange` | **DH vô danh.** 2 giá trị DH và $K_S^{DH}$ {`K_S, q,` $\alpha$ } | Không có trong DH cố định. |
| | **DH tức thời.** 3 thông số DH giống DH vô danh, cùng 1 chữ ký cho các thông số. S{`K_C, q,` $\alpha$ } | |
| | **RSA,** $K_S^{RSA}$ **chỉ ký.** Khóa công khai *tạm thời* server sinh ra cho mã hóa, được ký bởi khóa công khai RSA trong cert. | Nếu $K_S^{RSA}$ có thể mã hóa, **không** gửi thông báo này. |
| `certificate_request` | Các loại `certificate_type`:<br>• `RSA/DSS`, chữ ký số<br>• `RSA/DSS`, trong **DH cố định** cho authentication. | Không cần nói `cert.._authorities`. Không có trong DH vô danh. |
| `client_key_exchange` | **RSA.** Client gen `pre_master_secret`, mã hóa với khóa công khai RSA (dùng để mã hóa) tức thời của server. `E(pre_master_secret, K_S)` | |
| | **DH tức thời/vô danh.** Chứa thông số DH của client. (`K_C, q,` $\alpha$ ) | |
| | **DH cố định.** TB rỗng ∅. | |

Continued on next page

Bảng 1: Tham số các thông báo (Continued)

| Thông báo | Tham số | Mô tả |
|---|---|---|
| `certificate_verify` | Ký `hash(pre_msg, master_secret)` bởi khóa bí mật khớp với cert đã gửi. | SV xác minh client. Chỉ gửi khi `CA<<C>>` có khả năng ký. |

## 2.2 Bài tập



**2. An toàn mức giao vận (2,5 điểm)**

Trong một ứng dụng Web, hai bên client và server sử dụng giao thức bắt tay trong chuỗi giao thức SSL để xác thực lẫn nhau và thỏa thuận các tham số an ninh (giải thuật và khóa). Giả sử phương pháp trao đổi khóa được client và server thống nhất sử dụng là Diffie-Hellman trong đó server có cặp khóa riêng và khóa công khai Diffie-Hellman cố định (khóa công khai được chứng thực), còn client sinh ra cặp khóa riêng và khóa công khai Diffie-Hellman một cách tức thời (khóa công khai không được chứng thực).

*a. (1 điểm)*

Vẽ sơ đồ trao đổi thông báo 4 giai đoạn giữa client và server trong giao thức bắt tay SSL nêu trên.

*b. (1,5 điểm)*

Với mỗi thông báo tùy chọn (tức những thông báo không phải đối với bất kỳ phương pháp trao đổi khóa nào cũng được gửi) và thông báo *client_key_exchange*, hãy chỉ ra nó có những tham số cụ thể gì.

**3. An toàn mức giao vận (3 điểm)**

Trong một ứng dụng Web, hai bên client và server sử dụng giao thức Handshake trong chuỗi giao thức SSL để xác thực lẫn nhau và thỏa thuận các tham số an ninh (các giải thuật và khóa mật mã). Giả sử phương pháp trao đổi khóa được client và server thống nhất sử dụng sau khi trao đổi các thông báo *client_hello* và *server_hello* ở giai đoạn 1 là RSA. Client có sẵn một cặp khóa riêng và khóa công khai DSS trong đó khóa công khai DSS đã được chứng thực từ trước. Server cũng có sẵn một cặp khóa riêng và khóa công khai DSS trong đó khóa công khai DSS cũng đã được chứng thực từ trước.

*a. (1 điểm)*

Vẽ sơ đồ trao đổi thông báo 4 giai đoạn giữa client và server trong giao thức Handshake SSL nêu trên theo cách thức cho phép hai bên xác thực lẫn nhau. Chỉ rõ thông báo nào cho phép client xác thực server và ngược lại thông báo nào cho phép server xác thực client.

**Lời giải**

Vẽ đầy đủ tất cả các thông báo (không thiếu bất kỳ thông báo tùy chọn nào) 0,75 điểm

Thông báo server_key_exchange cho phép client xác thực server còn thông báo certificate_verify cho phép server xác thực client (0,25 điểm) (phải trả lời đúng cả hai ý mới được 0,25 điểm, chỉ cần sai một ý là bị 0 điểm cho câu hỏi này)

*b. (2 điểm)*

Với mỗi thông báo tùy chọn (tức những thông báo không phải đối với bất kỳ phương pháp trao đổi khóa nào cũng được gửi) và thông báo *client_key_exchange*, hãy chỉ ra nó có những tham số cụ thể gì.

**Lời giải**

Thông báo certificate ở giai đoạn 2 chứa chứng thực khóa công khai DSS của server (Có thể viết tắt là CA<<S>>$^{DSS}$ hay CA<<S>>$_{DSS}$) 0,25 điểm

Thông báo server_key_exchange chứa khóa công khai RSA có chức năng mã hóa của server (0,25 điểm) được ký với khóa riêng DSS của server (0,25 điểm) (Có thể viết tắt là S{$PU_s^{RSA}$}$^{DSS}$)

Thông báo certificate_request bao gồm certificate_type và certificate_authorities trong đó certificate_type chỉ ra kiểu giải thuật mật mã khóa công khai là DSS và chế độ sử dụng là chữ ký số (cũng có thể viết là xác thực), không cần nói rõ về certificate_authorities (0,25 điểm)

Thông báo certificate ở giai đoạn 3 chứa chứng thực khóa công khai DSS của client (Có thể viết tắt là CA<<C>>$^{DSS}$ hay CA<<C>>$_{DSS}$) 0,25 điểm

Thông báo client_key_exchange chứa khóa bí mật pre_master_secret được sinh ra bởi client (0,25 điểm) và được mã hóa với khóa công khai RSA của server (0,25 điểm) (Có thể viết tắt là E($PU_s^{RSA}$, pre_master_secret))

Thông báo certificate_verify chứa chữ ký của client trên các thông báo client và server trước đó trao đổi với nhau và master_secret sử dụng khóa riêng DSS của client (0,25 điểm)

Consider the SSL Handshake Protocol. Suppose that the RSA key exchange method is used. Both the client and the server have a fixed RSA public/private key pair. Both public keys are certified by a certificate authority. They can only be used for digital signature purposes and are not suitable for encryption. The client and the server need to authenticate each other.

*a. (1 point)*

Draw the message exchange expected for this scenario.

*b. (1.5 point)*

Describe the parameters associated with each situation dependent message and with the *client_key_exchange* message.

Trong một ứng dụng Web, hai bên client và server sử dụng giao thức bắt tay trong chuỗi giao thức SSL để xác thực lẫn nhau và thỏa thuận các tham số an ninh (giải thuật và khóa). Giả sử phương pháp trao đổi khóa được client và server thống nhất sử dụng là Diffie-Hellman trong đó server có cặp khóa riêng và khóa công khai Diffie-Hellman cố định (khóa công khai được chứng thực), còn client sinh ra cặp khóa riêng và khóa công khai Diffie-Hellman một cách tức thời (khóa công khai không được chứng thực).

*a. (1 điểm)*

Vẽ sơ đồ trao đổi thông báo 4 giai đoạn giữa client và server trong giao thức bắt tay SSL nêu trên.

*b. (1,5 điểm)*

Với mỗi thông báo tùy chọn (tức những thông báo không phải đối với bất kỳ phương pháp trao đổi khóa nào cũng được gửi) và thông báo *client_key_exchange*, hãy chỉ ra nó có những tham số cụ thể gì.

# 3 Kerberos

**Table 4.1** Summary of Kerberos Version 4 Message Exchanges

| | |
|---|---|
| **(1) C → AS** | $ID_c \| ID_{tgs} \| TS_1$ |
| **(2) AS → C** | $\mathrm{E}(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$ |
| | $Ticket_{tgs} = \mathrm{E}(K_{tgs}, [K_{c,\,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| \mathrm{Lifetime}_2])$ |

(a) Authentication Service Exchange to obtain ticket-granting ticket

| | |
|---|---|
| **(3) C → TGS** | $ID_v \| Ticket_{tgs} \| Authenticator_c$ |
| **(4) TGS → C** | $\mathrm{E}(K_{c,\,tgs}, [K_{c,\,v} \| ID_v \| TS_4 \| Ticket_v])$ |
| | $Ticket_{tgs} = \mathrm{E}(K_{tgs}, [K_{c,\,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| \mathrm{Lifetime}_2])$ |
| | $Ticket_v = \mathrm{E}(K_v, [K_{c,\,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| \mathrm{Lifetime}_4])$ |
| | $Authenticator_c = \mathrm{E}(K_{c,\,tgs}, [ID_C \| AD_C \| TS_3])$ |

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

| | |
|---|---|
| **(5) C → V** | $Ticket_v \| Authenticator_c$ |
| **(6) V → C** | $\mathrm{E}(K_{c,\,v}, [TS_5 + 1])$(for mutual authentication) |
| | $Ticket_v = \mathrm{E}(K_v, [K_{c,\,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| \mathrm{Lifetime}_4])$ |
| | $Authenticator_c = \mathrm{E}(K_{c,\,v}, [ID_C \| AD_C \| TS_5])$ |

(c) Client/Server Authentication Exchange to obtain service

**Table 4.2** Rationale for the Elements of the Kerberos Version 4 Protocol

| | |
|---|---|
| **Message (1)** | Client requests ticket-granting ticket. |
| $ID_C$ | Tells AS identity of user from this client. |
| $ID_{tgs}$ | Tells AS that user requests access to TGS. |
| $TS_1$ | Allows AS to verify that client's clock is synchronized with that of AS. |
| **Message (2)** | AS returns ticket-granting ticket. |
| $K_c$ | Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2). |
| $K_{c,tgs}$ | Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key. |
| $ID_{tgs}$ | Confirms that this ticket is for the TGS. |
| $TS_2$ | Informs client of time this ticket was issued. |
| $Lifetime_2$ | Informs client of the lifetime of this ticket. |
| $Ticket_{tgs}$ | Ticket to be used by client to access TGS. |

(a) Authentication Service Exchange

| | |
|---|---|
| **Message (3)** | Client requests service-granting ticket. |
| $ID_V$ | Tells TGS that user requests access to server V. |
| $Ticket_{tgs}$ | Assures TGS that this user has been authenticated by AS. |
| $Authenticator_c$ | Generated by client to validate ticket. |
| **Message (4)** | TGS returns service-granting ticket. |
| $K_{c,tgs}$ | Key shared only by C and TGS protects contents of message (4). |
| $K_{c,v}$ | Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key. |
| $ID_V$ | Confirms that this ticket is for server V. |
| $TS_4$ | Informs client of time this ticket was issued. |
| $Ticket_V$ | Ticket to be used by client to access server V. |
| $Ticket_{tgs}$ | Reusable so that user does not have to reenter password. |
| $K_{tgs}$ | Ticket is encrypted with key known only to AS and TGS, to prevent tampering. |
| $K_{c,tgs}$ | Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket. |
| $ID_C$ | Indicates the rightful owner of this ticket. |
| $AD_C$ | Prevents use of ticket from workstation other than one that initially requested the ticket. |
| $ID_{tgs}$ | Assures server that it has decrypted ticket properly. |
| $TS_2$ | Informs TGS of time this ticket was issued. |
| $Lifetime_2$ | Prevents replay after ticket has expired. |
| $Authenticator_c$ | Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay. |

| | |
|---|---|
| $K_{c,tgs}$ | Authenticator is encrypted with key known only to client and TGS, to prevent tampering. |
| $ID_C$ | Must match ID in ticket to authenticate ticket. |
| $AD_C$ | Must match address in ticket to authenticate ticket. |
| $TS_3$ | Informs TGS of time this authenticator was generated. |

(b) Ticket-Granting Service Exchange

| | |
|---|---|
| **Message (5)** | Client requests service. |
| $Ticket_V$ | Assures server that this user has been authenticated by AS. |
| $Authenticator_c$ | Generated by client to validate ticket. |
| **Message (6)** | Optional authentication of server to client. |
| $K_{c,v}$ | Assures C that this message is from V. |
| $TS_5 + 1$ | Assures C that this is not a replay of an old reply. |
| $Ticket_v$ | Reusable so that client does not need to request a new ticket from TGS for each access to the same server. |
| $K_v$ | Ticket is encrypted with key known only to TGS and server, to prevent tampering. |
| $K_{c,v}$ | Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket. |
| $ID_C$ | Indicates the rightful owner of this ticket. |
| $AD_C$ | Prevents use of ticket from workstation other than one that initially requested the ticket. |
| $ID_V$ | Assures server that it has decrypted ticket properly. |
| $TS_4$ | Informs server of time this ticket was issued. |
| $Lifetime_4$ | Prevents replay after ticket has expired. |
| $Authenticator_c$ | Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay. |
| $K_{c,v}$ | Authenticator is encrypted with key known only to client and server, to prevent tampering. |
| $ID_C$ | Must match ID in ticket to authenticate ticket. |
| $AD_c$ | Must match address in ticket to authenticate ticket. |
| $TS_5$ | Informs server of time this authenticator was generated. |

(c) Client/Server Authentication Exchange

## 3.1 Bài tập

**1. Phân phối khóa và xác thực người dùng (2,5 điểm)**

Xét hội thoại xác thực Kerberos 4. Như đã biết, trong trường hợp người dùng thuộc về một phân hệ A muốn truy nhập vào server dịch vụ thuộc về một phân hệ B khác với A thì các bên liên quan bao gồm client C, server xác thực AS của phân hệ A, server cấp thẻ TGS của phân hệ A, server cấp thẻ TGS của phân hệ B và server dịch vụ V của phân hệ B phải trao đổi với nhau tổng cộng 8 thông báo (kể cả thông báo V gửi cho C để C xác thực V).

*c. (1 điểm)*

Hãy thêm các thông tin $Hệ_C$, $Hệ_{tgs}$ và $Hệ_V$ chỉ phân hệ của người dùng, phân hệ của server cấp thẻ TGS và phân hệ của server dịch vụ V một cách tương ứng vào những chỗ thích hợp trong hội thoại xác thực Kerberos 4 để tổng số thông báo trao đổi trong trường hợp truy nhập liên phân hệ giảm xuống còn 6. Yêu cầu đặt ra là giữ nguyên các thông tin khác của hội thoại Kerberos 4 và cũng không được thêm bất kỳ thông tin nào khác vào hội thoại ngoài các thông tin chỉ phân hệ đã nêu.

*d. (1,5 điểm)*

Viết hội thoại trao đổi liên phân hệ cho phép người dùng thuộc một phân hệ này truy nhập vào server dịch vụ thuộc một phân hệ khác (ở xa)?

*a. (1 điểm)*

(a) Trao đổi với dịch vụ xác thực : để có thẻ cấp thẻ

(1) $C \rightarrow AS : ID_C \parallel Hệ_C \parallel ID_{tgs} \parallel TS_1$

(2) $AS \rightarrow C : E_{KC}[K_{C,tgs} \parallel Hệ_{tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Hạn_2 \parallel Thẻ_{tgs}]$

$Thẻ_{tgs} = E_{Ktgs}[K_{C,tgs} \parallel Hệ_C \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Hạn_2]$

(b) Trao đổi với dịch vụ cấp thẻ : để có thẻ dịch vụ

(3) $C \rightarrow TGS : ID_V \parallel Thẻ_{tgs} \parallel Dấu_C$

(4) $TGS \rightarrow C : E_{KC,tgs}[K_{C,V} \parallel Hệ_V \parallel ID_V \parallel TS_4 \parallel Thẻ_V]$

$Thẻ_V = E_{KV}[K_{C,V} \parallel Hệ_C \parallel ID_C \parallel AD_C \parallel ID_V \parallel TS_4 \parallel Hạn_4]$

$Dấu_C = E_{KC,tgs}[Hệ_C \parallel ID_C \parallel AD_C \parallel TS_3]$

(c) Trao đổi xác thực client/server : để có dịch vụ

(5) $C \rightarrow V : Thẻ_V \parallel Dấu_C$

(6) $V \rightarrow C : E_{KC,V}[TS_5 + 1]$

$Dấu_C = E_{KC,V}[Hệ_C \parallel ID_C \parallel AD_C \parallel TS_5]$

*b. (1,5 điểm)*

**Giải.** Vì sao lại thế?