BG ChVII Bài 3. Trao đổi khoá và thoả thuận khoá.

6.3.1. Giao thức trao đổi khoá Diffie-Hellman.

Hệ phân phối khoá Diffie-Hellman nói trong mục trước có thể dễ dàng biến đổi thành một giao thức trao đổi (hay thoả thuận) khoá trực tiếp giữa các người sử dụng mà không cần có sự can thiệp của một TA làm nhiệm vụ điều hành hoặc phân phối khoá. Một nhóm bất kỳ người sử dụng có thể thoả thuận

cùng dùng chung một số nguyên tố lớn p và một phần tử nguyên thuỷ α theo mod$p$, hai người bất kỳ trong nhóm A và B mỗi khi muốn truyền tin bảo mật cho nhau có thể cùng thực hiện giao thức say đây để trao đổi khoá:

1. A chọn ngẫu nhiên số a_A ($0 \le a_A \le p-2$), giữ bí mật a_A, tính $b_A = \alpha^{a_A} \bmod p$ và gửi b_A cho B.
2. Tương tự, B chọn ngẫu nhiên số a_B ($0 \le a_B \le p-2$), giữ bí mật a_B , tính $b_B = \alpha^{a_B} \bmod p$ và gửi b_B cho B.

3. A và B cùng tính được khoá chung

$$K_{A,B} = b_B^{a_A} \bmod p = b_A^{a_B} \bmod p \,(= \alpha^{a_A a_B} \bmod p)\,.$$

Giao thức trao đổi khoá Diffie-Hellman có các tính chất sau:

1. *Giao thức là an toàn đối với việc tấn công thụ động*, nghĩa là một người thứ ba, dù biết b_A và b_B sẽ khó mà biết được $K_{A,B}$.

Ta biết rằng bài toán "biết b_A và b_B tìm $K_{A,B}$" chính là bài toán Diffie-Hellman, và trong mục 6.2.3 ta có nói rằng bài toán đó tương đương với bài toán phá

mật mã ElGamal. Bây giờ ta chứng minh điều này. Phép mật mã ElGamal với khoá $K = (p, \alpha, a, \beta)$, trong đó $\beta = \alpha^a \bmod p$, cho ta từ một bản rõ x và một số ngẫu nhiên $k \in Z_{p-1}$ lập được mật mã $e_K(x,k) = (y_1, y_2)$, trong đó $y_1 = \alpha^k \bmod p,\ y_2 = x\beta^k \bmod p.$

Và phép giải mã được cho bởi

$$d_K(y_1, y_2) = y_2 (y_1^a)^{-1} \bmod p.$$

Giả sử ta có thuật toán A giải bài toán Diffie-Hellman. Ta sẽ dùng A để phá mã ElGamal như sau: Cho mật mã (y_1, y_2). Trước

hết, dùng A cho $y_1 = \alpha^k \bmod p$ và $\beta = \alpha^a \bmod p$, ta được

$$\mathrm{A}(y_1, \beta) = \alpha^{ka} = \beta^k \bmod p,$$

và sau đó ta thu được bản rõ x từ β^k và y_2 như sau :

$$x = y_2 (\beta^k)^{-1} \bmod p.$$

Ngược lại, giả sử có thuật toán B phá mã ElGamal, tức $\mathrm{B}(p, \alpha, \beta, y_1, y_2) = x = y_2 (y_1^a)^{-1} \bmod p$.

áp dụng B cho $\beta = b_A, y_1 = b_B, y_2 = 1$, ta được

$$\mathrm{B}(p, \alpha, b_A, b_B, 1)^{-1} = (1.(b_B^{a_A})^{-1})^{-1} = \alpha^{a_A a_B} \bmod p,$$

tức là giải được bài toán Diffie-Hellman.

2. *Giao thức là không an toàn đối với việc tấn công chủ động bằng cách đánh tráo giữa đường*, nghĩa là một người thứ ba C có thể đánh tráo các thông tin trao đổi giữa A và B, chẳng hạn, C thay α^{a_A} mà A định gửi cho B bởi $\alpha^{a'_A}$,và thay α^{a_B} mà B định gửi cho A bởi $\alpha^{a'_B}$, như vậy, sau khi thực hiện giao thức trao đổi khoá, A đã lập một khoá chung $\alpha^{a_A a'_B}$ với C mà vẫn tưởng là với B, đồng thời B đã lập một khoá chung $\alpha^{a'_A a_B}$ với C mà vẫn tưởng là với A; C có thể giải mã mọi

thông báo mà A tưởng nhầm là mình gửi đến B, cũng như mọi thông báo mà B tưởng nhầm là mình gửi đến A !

Một cách khắc phục kiểu tấn công chủ động nói trên là làm sao để A và B có thể kiểm thử để xác nhận tính đúng đắn của các khoá công khai b_A và b_B .Đưa vào giao thức trao đổi khoá Diffie-Hellman thêm vai trò điều phối của một TA để được một hệ phân phối khoá Diffie-Hellman như ở mục 6.2.3 là một cách khắc phục như vậy. Trong hệ

phân phối khoá Diffie-Hellman, sự can thiệp của TA là rất yếu, thực ra TA chỉ làm mỗi một việc là cấp chứng chỉ xác nhận khoá công khai cho từng người dùng chứ không đòi hỏi biết thêm bất cứ một bí mật nào của người dùng. Tuy nhiên, nếu chưa thoả mãn với vai trò hạn chế đó của TA, thì có thể cho TA một vai trò xác nhận yếu hơn, không liên quan gì đến khoá, chẳng hạn như xác nhận thuật toán kiểm thử chữ kỹ của người dùng, còn bản thân các thông tin về khoá (cả bí mật

và công khai) thì do các người dùng trao đổi trực tiếp với nhau. Với cách khắc phục có vai trò rất hạn chế đó của TA, ta được giao thức sau đây:

6.3.2. Giao thức trao đổi khoá DH có chứng chỉ xác nhận.

Mỗi người dùng A có một danh tính *ID*(A) và một sơ đồ chữ ký với thuật toán ký sig_A và thuật toán kiểm thử ver_A. TA cũng có một vai trò xác nhận, nhưng không phải xác nhận bất kỳ thông tin nào liên quan đến

việc tạo khoá mật mã của người dùng (dù là khoá bí mật hay là khoá công khai), mà chỉ là xác nhận một thông tin ít quan hệ khác như thuật toán kiểm thử chữ ký của người dùng. Còn bản thân các thông tin liên quan đến việc tạo khoá mật mã thì các người dùng sẽ trao đổi trực tiếp với nhau. TA cũng có một sơ đồ chữ ký của mình, gồm một thuật toán ký sig_{TA} và một thuật toán kiểm thử (công khai) ver_{TA}. Chứng chỉ mà TA cấp cho mỗi người dùng A sẽ là

C(A) = (*ID*(A), ver_A , sig_{TA}(*ID*(A), ver_A)).
Rõ ràng trong chứng chỉ đó TA không xác nhận bất kỳ điều gì liên quan đến việc tạo khoá của A cả. Việc trao đổi khoá giữa hai người dùng A và B được thực hiện theo giao thức sau đây:

1.A chọn ngẫu nhiên số $a_A (0 \le a_A \le p-2)$, tính $b_A = \alpha^{a_A} \bmod p$, và gửi b_A cho B.

2. B chọn ngẫu nhiên số $a_B (0 \le a_B \le p-2)$, tính $b_B = \alpha^{a_B} \bmod p$, tính tiếp $K = b_A^{a_B} \bmod p$,

$$y_B = sig_B(b_B, b_A),$$

và gửi (C(B),b_B , y_B) cho A.

3. A tính $K = b_B^{a_A} \bmod p$,

dùng ver_{B} để kiểm thử y_{B} , dùng ver_{TA} để kiểm thử C(B), sau đó tính

$$y_{\mathrm{A}} = sig_{\mathrm{A}}(b_{\mathrm{A}} , b_{\mathrm{B}}),$$

và gửi (C(A), y_{A}) cho B.

4.B dùng ver_{A} để kiểm thử y_{A} ,và dùng ver_{TA} để kiểm thử C(A).

Nếu tất cả các bước đó được thực hiện và các phép kiểm thử đều cho kết quả đúng đắn, thì giao thức kết thúc, và cả A và B đều có được khoá chung K . Do việc dùng các thuật toán kiểm

thử nên A biết chắc giá trị b_B là của B và B biết chắc giá trị b_A là của A, loại trừ khả năng một người C nào khác đánh tráo các giá trị đó giữa đường.

6.3.3. Giao thức trao đổi khoá Matsumoto-Takashima- Imai.

Giao thức trình bày trong mục trên cần dùng ba lần chuyển tin qua lại để thiết lập một khoá chung. Các tác giả Nhật Matsumoto, Takashima và Imai đề nghị một cải tiến để chỉ dùng một giao thức gồm hai lần

chuyển tin (một từ A đến B và một từ B đến A) để thoả thuận khoá như sau:

Ta giả thử rằng trước khi thực hiện giao thức, TA đã ký cấp chứng chỉ cho mỗi người dùng A theo cách làm ở mục 6.2.3:

$C(A) = (ID(A), b_A, sig_{TA}(ID(A), b_A))$,

và thuật toán kiểm thử chữ ký ver_{TA} của TA là công khai. Trong giao thức này, các b_A không trực tiếp tạo nên các khoá mật mã cho truyền tin, mà với mỗi phiên truyền tin bảo mật, khoá phiên (sesion key) sẽ được

tạo ra cho từng phiên theo giao thức.

Giao thức trao đổi khoá phiên MTI gồm ba bước (trong đó có hai lần chuyển tin) như sau:

1. A chọn ngẫu nhiên số $r_A (0 \le r_A \le p-2)$, tính $s_A = \alpha^{r_A} \bmod p$, và gửi (C(A), s_A) cho B.

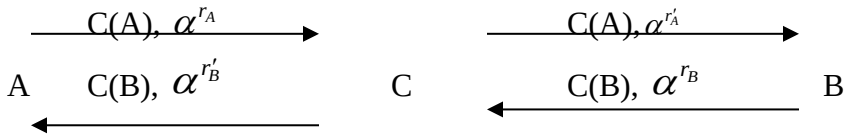
2. B chọn ngẫu nhiên số $r_B (0 \le r_B \le p-2)$, tính $s_B = \alpha^{r_B} \bmod p$, và gửi (C(B),$s_B$) cho A.

3. A tính $K = s_B^{a_A} . b_B^{r_A} \bmod p$, với giá trị b_B thu được từ C(B),

B tính $K = s_A^{a_B} . b_A^{r_B} \bmod p$, với giá trị b_A thu được từ C(A).

Hai cách tính đó đều cho cùng một giá trị $K = \alpha^{r_A a_B + r_B a_A} \bmod p$.

Giao thức này cũng có khả năng giữ bí mật khoá K như đối với giao thức Diffie-Hellman trước sự tấn công thụ động. Tuy nhiên, vì không có chứng chỉ đối với các giá trị s_A , s_B nên vẫn có nguy cơ của sự tấn công tích cực bằng việc đánh tráo giữa đường bởi một C nào đó theo kiểu sau đây:



Đáng lẽ A gửi đến B (C(A),s_A) thì C đánh tráo bằng cách nhận

(C(A),s_A) và gửi đến B $(C(A), s'_A)$, với $s'_A = \alpha^{r'_A} \bmod p$, và ngược lại, đáng lẽ B gửi đến A (C(B), s_B) thì C đánh tráo bằng cách nhận (C(B), s_B) và gửi đến A $(C(B), s'_B)$, với $s'_B = \alpha^{r'_B} \bmod p$. Khi đó, A tính được khoá

$$K_1 = \alpha^{r_A a_B + r'_B a_A} \bmod p ,$$

và B tính được khoá

$$K_2 = \alpha^{r'_A a_B + r_B a_A} \bmod p.$$

Hai giá trị K_1 và K_2 này khác nhau, nên không giúp A và B truyền tin được cho nhau, nhưng C không có khả năng tính được giá trị nào trong hai giá trị đó (vì

không biết a_A và a_B), nên khác với giao thức Diffie-Hellman ở mục 6.2.3, ở đây C chỉ có thể phá rối, chứ không thể đánh cắp thông tin được.

6.3.4. Giao thức Girault trao đổi khoá không chứng chỉ.

Giao thức Girault được đề xuất năm 1991. Trong giao thức này, người sử dụng A không cần dùng chứng chỉ C(A), mà thay bằng một *khoá công khai tự chứng thực* , được cấp trước bởi một TA. Phương pháp này sử

dụng kết hợp các đặc tính của các bài toán RSA và lôgarit rời rạc.

Giả thử n là tích của hai số nguyên tố lớn p và q, $n = p.q$, p và q có dạng $p = 2p_1+1$, $q = 2q_1+1$, trong đó p_1 và q_1 cũng là các số nguyên tố. Nhóm nhân Z_n^* đẳng cấu với tích $Z_p^* \times Z_q^*$. Cấp cao nhất của một phần tử trong Z_n^* là bội chung bé nhất của p -1 và q -1, tức là bằng $2p_1q_1$. Giả sử α là một phần tử cấp $2p_1q_1$ của Z_n^*. Nhóm cyclic sinh bởi α được ký hiệu là G, bài toán tính lôgarit rời

rạc theo cơ số α trong G được giả thiết là rất khó.

Các số n và α là công khai. Chỉ TA biết p ,q . TA chọn số mũ công khai e , với gcd(e, ϕ (n))=1,và giữ bí mật $d = e^{-1} \bmod \phi(n)$.

Mỗi người dùng A có một danh tính ID(A), chọn ngẫu nhiên một số $a_A \in$ G, giữ bí mật a_A và tính $b_A = \alpha^{a_A} \bmod n$, rồi gửi a_A ,b_A cho TA. TA thử lại điều kiện $b_A = \alpha^{a_A} \bmod n$, rồi cấp cho A một *khoá công khai tự chứng thực* $p_A = (b_A - ID(A))^d \bmod n$. Trong

khoá công khai p_A không có thông tin về a_A , nhưng TA cần biết a_A để thử điều kiện $b_A = \alpha^{a_A} \bmod n$.

Giao thức Girault trao đổi khoá giữa hai người dùng A và B được thực hiện bởi các bước sau đây:

1. A chọn ngẫu nhiên $r_A \in G$, tính $s_A = \alpha^{r_A} \bmod n$, và gửi cho B ($ID$(A),$p_A$, s_A).

2. B chọn ngẫu nhiên $r_B \in G$, tính $s_B = \alpha^{r_B} \bmod n$, và gửi cho A ($ID$(B), p_B , s_B).

3. A tính khoá $K = s_B^{a_A}(p_B^e + ID(V))^{r_A} \bmod n$,

B tính khoá $K = s_A^{a_B}(p_A^e + ID(A))^{r_B} \bmod n.$ Cả hai giá trị đó của K đều bằng nhau và bằng

$$K = \alpha^{r_A a_B + r_B a_A} \bmod n.$$

Bằng các lập luận như trong mục trước, ta dễ thấy rằng một người thứ ba C khó mà tạo ra các thông tin giả mạo để gửi đến A hoặc B, nếu tấn công bằng cách đánh tráo giữa đường thì có thể phá rối để ngăn cản A và B tạo lập khoá chung, nhưng không thể đánh cắp thông tin trao đổi giữa A và B.

Còn lại một vấn đề: Tại sao TA cần biết a_A và thử điều kiện $b_A = \alpha^{a_A} \bmod n$ trước khi cấp p_A cho A? Ta giả thử rằng TA không biết a_A và cấp $p_A = (b_A - ID(A))^d \bmod n$ cho A, và thử xem có thể xẩy ra chuyện gì?

Một người thứ ba C có thể chọn một giá trị rởm a'_A, và tính $b'_A = \alpha^{a'_A} \bmod n,$ rồi tính $b'_C = b'_A - ID(A) - ID(C),$ và đưa $(ID(C), b'_C)$ cho TA. TA sẽ cấp cho C một "khoá công khai tự chứng thực"

$$p'_C = (b'_C - ID(C))^d \bmod n.$$

Vì $b'_C - ID(C) = b'_A - ID(A),$ nên thực tế C đã được cấp

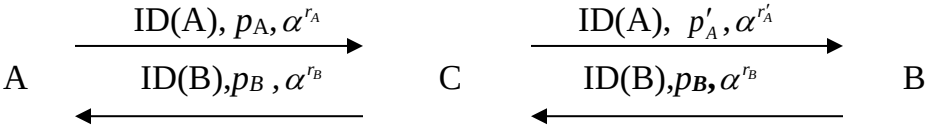
$$p'_C = p'_A = (b'_A - ID(A))^d \bmod n.$$

Bây giờ giả sử A và B thực hiện giao thức trao đổi khoá, và C xen vào ở giữa, như vậy, A gửi đến B $(ID(A), p_A, \alpha^{r_A} \bmod n),$ nhưng do bị C đánh tráo nên B lại nhận được $(ID(A), p'_A, \alpha^{r'_A} \bmod n),$ do đó B và C tính được cùng một khoá

$$K' = \alpha^{r'_A a_B + r_B a'_A} \bmod n = s_B^{a'_A} (p_B^e + ID(B))^{r'_A} \bmod n,$$

còn A tính được khoá

$$K = \alpha^{r_A a_B + r_B a_A} \bmod n.$$



B và C có cùng một khoá khác với khoá của A, nhưng B vẫn nghĩ rằng mình có chung khoá với A. Vì thế, C có thể giải mã mọi thông báo mà B gửi cho A, tức đánh cắp các thông tin từ B đến A. Việc TA biết a_A và thử điều kiện $b_A = \alpha^{a_A} \bmod n$ trước khi cấp p_A cho A là để loại trừ khả năng đánh tráo như vậy của một kẻ tấn công C.