# Về một tiêu chuẩn tham số cho bài toán logarithm rời rạc

**Nguyễn Quốc Toàn, Đỗ Đại Chí, Triệu Quang Phong**

***Tóm tắt*— Bài báo này trình bày về một số tấn công khôi phục khóa bí mật của Lim-Lee trên các giao thức trao đổi khóa kiểu Diffie-Hellman thực hiện trong một nhóm con cấp nguyên tố dựa vào bài toán logarithm rời rạc (DLP). Tấn công này có thể tiết lộ một phần hoặc toàn bộ khóa bí mật trong hầu hết các giao thức trao đổi khóa kiểu Diffie-Hellman. Tấn công liên quan chặt chẽ với việc lựa chọn các tham số cũng như việc kiểm tra tính hợp lệ khóa công khai. Từ đó, chúng tôi đề xuất một tiêu chuẩn cho tham số modulo *p* dựa trên bài toán logarithm rời rạc nhằm làm tăng cường độ an toàn cũng như tính hiệu quả của các hệ mật dựa trên bài toán logarithm rời rạc.**

***Abstract*— In this paper, we present several key recovery attacks proposed by Lim-Lee on Diffie-Hellman- type key exchange protocols which use a prime order subgroup on discrete logarithm problem. This attack may reveal part of, or the whole secret key in these protocols. In addition, this attack is closely related to the selection of parameters and the verification of validity of the public key. Then, we propose a criterion for prime modulo *p* based on discrete logarithm problem to enhance security and the efficiency of discrete log-based cryptography systems.**

***Từ khóa— tấn công khôi phục khóa; bài toán logarit rời rạc; giao thức trao đổi khóa.***

## I. GIỚI THIỆU

Nhiều giao thức mật mã đã được phát triển dựa trên bài toán logarithm rời rạc. Mục tiêu chính của người phát triển là thiết kế một giao thức khó bị phá, tương đương với tính phức tạp của bài toán logarithm rời rạc cơ bản dưới một số giả thiết hợp lý. Còn mục đích của kẻ tấn công là tìm khóa bí mật liên quan hoặc thực hiện giao thức như một người sử dụng hợp pháp mà không cần biết khóa bí mật. Mặc dù có những giao thức đã được đảm bảo (bởi chứng minh an toàn) rằng không có tấn công hiệu quả trên giao thức đó, nhưng chúng ta vẫn nên phân tích kỹ lưỡng ở khía cạnh an toàn thực hành. Vấn đề này có thể tham chiếu đến hai bài báo gần đây, một bài về chứng minh độ an toàn của Pointcheval và Stern [1] và về giả mạo chữ ký của Bleichenbacher [2].

Mục đích của bài báo này là chỉ ra tính không an toàn của nhiều lược đồ dựa trên logarithm rời rạc khác nhau sử dụng nhóm con cấp nguyên tố. Chính xác hơn, chúng tôi trình bày kiểu tấn công khôi phục khóa của Lim-Lee [4] trên những giao thức kiểu Diffie-Hellman mà có thể tìm tất cả hoặc một phần các bit khóa bí mật. Tấn công liên quan chặt chẽ với việc lựa chọn các tham số và kiểm tra các biến của giao thức. Từ đó, có thể phòng chống tấn công bằng cách bổ sung các bước kiểm tra phù hợp, hoặc sử dụng các tham số an toàn (lưu ý rằng, "an toàn" ở đây được hiểu là các tham số an toàn đối với tấn công, có nghĩa là các tham số chuẩn thường được sử dụng là không an toàn với tấn công từ góc nhìn của bài báo này).

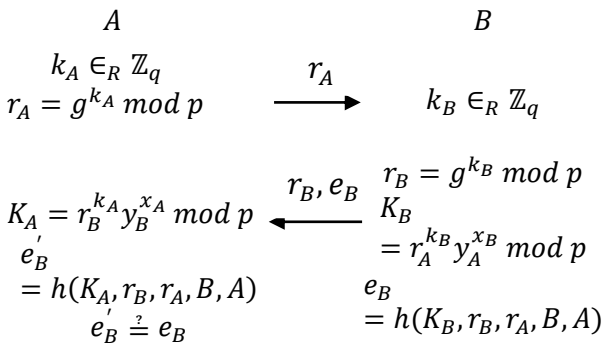
Bố cục bài báo gồm các phần sau: Sau Mục I, Mục II trình bày về giao thức MTI (Matsumoto - Takashima - Imai) sửa đổi sử dụng hàm băm *h* (gọi là Giao thức 1) và một giao thức trao đổi khóa khác [3] mà có chia sẻ cặp khóa công khai và bí mật dài hạn tương tự như Giao thức 1 (gọi là Giao thức 2). Mục III và IV trình bày kiểu tấn công khôi phục khóa của Lim-Lee [1] trên Giao thức 1 và Giao thức 2. Mục V chỉ ra cách khắc phục tấn công của Lim-Lee đối với hai giao thức này. Mục VI chỉ rõ hơn tấn công của Lim-Lee đối với giao thức HMQV (Hash Menezes-Qu-Vanstone) kiểu Diffie-Hellman liên quan đến các ước nhỏ của *p*-1. Mục VII trình bày về tham số an toàn cho bài toán DLP và vấn đề sinh số nguyên tố. Cuối cùng là Kết luận.

## II. GIAO THỨC MTI

Năm 1986, Matsumoto, Takashima và Imai đã chỉ ra cách thức để định nghĩa các giao thức thỏa thuận khóa có xác thực. Nhiều giao thức đã được thiết kế sử dụng ý tưởng tương tự như giao thức MTI. Xem xét chi tiết về các giao thức này sẽ rất hữu ích trong việc hiểu rõ hơn về các giao thức dựa trên chúng. Đây cũng là một phương tiện hữu ích để giải thích về các tấn công lên các giao thức thỏa thuận khóa.

Trong mục này, chúng ta giới hạn ở một phiên bản chứng thực của giao thức MTI với một số thay đổi nhỏ. Giao thức này được nghiên cứu trong các tài liệu [5, 6] và cũng đã được chuẩn hóa trong ISO/IEC JTC1/SC27 [7] (một trong những khuyến nghị của tiêu chuẩn này đối với một ví dụ cho cơ chế thỏa thuận khóa thứ 5, giao thức MTI A(0), là sử dụng số nguyên tố *p* an toàn (safe prime)).

Chia sẻ thông tin: hàm băm $h$.




Hình 1. Giao thức MTI sửa đổi sử dụng hàm băm $h$ (Giao thức 1).

Việc trao đổi khóa của hai thực thể $A$ và $B$ dựa trên bài toán DLP với các tham số chung là $(p, q, g)$. Trong đó, $A$ có cặp khóa công khai và bí mật dài hạn là $(x_A, y_A)$ với $y_A = g^{x_A} \bmod p$ và $B$ cũng có cặp khóa công khai và bí mật dài hạn $(y_B, x_B)$ với tính chất tương tự. Hai thực thể này chia sẻ khóa công khai dài hạn trước với nhau và giữ bí mật về khóa bí mật dài hạn. Ngoài ra, họ còn sử dụng chung hàm băm an toàn $h$. Việc trao đổi khóa được thực hiện theo các bước như sau.

*Bước 1:* $A$ lấy ngẫu nhiên $k_A \in_R \mathbb{Z}_q$, tính $r_A = g^{k_A} \bmod p$ và gửi $r_A$ tới $B$.

*Bước 2:* $B$ lấy ngẫu nhiên $k_B \in_R \mathbb{Z}_q$, tính $r_B = g^{k_B} \bmod p$, $K_B = r_A^{k_B} y_A^{x_B} \bmod p$, $e_B = h(K_B, r_B, r_A, B, A)$ và gửi $\{r_B, e_B\}$ tới $A$.

*Bước 3:* $A$ tính $K_A = y_B^{k_A} r_B^{x_A} \bmod p$, $e_B' = h(K_A, r_B, r_A, B, A)$ và kiểm tra $e_B = e_B'$ hay không. Nếu $e_B \neq e_B'$, thì $A$ dừng trao đổi và giao thức thất bại. (Tùy chọn) trường hợp còn lại, $A$ tính $e_A = h(K_A, r_A, r_B, A, B)$ và gửi $e_A$ tới $B$.

*Bước 4:* (Tùy chọn) $B$ tính $e_A' = h(K_B, r_A, r_B, A, B)$ và kiểm tra $e_A = e_A'$ hay không. Nếu $e_A \neq e_A'$, thì $B$ dừng trao đổi và giao thức với thất bại.

Trong giao thức này, các khóa tạm thời $K_A = K_B$, do

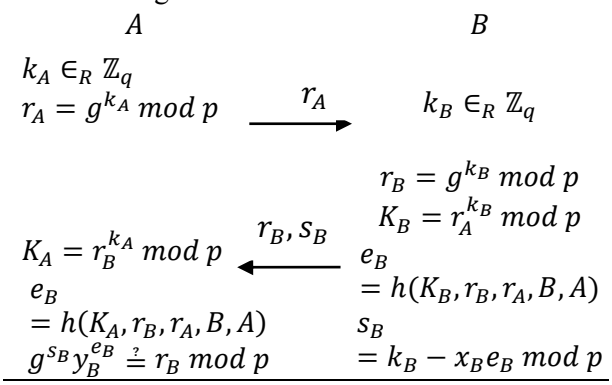
$$K_A = y_B^{k_A} r_B^{x_A} = (g^{x_B})^{k_A} \left(g^{k_B}\right)^{x_A} = \left(g^{k_A}\right)^{x_B} (g^{x_A})^{k_B} = y_A^{k_B} r_A^{k_B} = K_B.$$

Ngoài ra, chúng ta cũng xem xét một giao thức sau đây (Giao thức 4 trong [3]) giữa hai thực thể $A$ và $B$, trong đó $A$ và $B$ chia sẻ cặp khóa công khai và bí mật dài hạn tương tự như giao thức ở trên.

*Bước 1:* $A$ chọn một số ngẫu nhiên $k_A \in_R \mathbb{Z}_q$, tính $r_A = g^{k_A} \bmod p$ và gửi $r_A$ tới $B$.

*Bước 2:* $B$ chọn một số ngẫu nhiên $k_B \in_R \mathbb{Z}_q$, tính $r_B = g^{k_B} \bmod p$. Sau đó, $B$ tính $K_B = r_A^{k_B} \bmod p$, $e_B = h(K_B, r_B, r_A, B, A)$, $s_B = k_B - x_B e_B \bmod p$ và gửi $\{r_B, s_B\}$ tới $A$.

Chia sẻ thông tin: hàm băm $h$.




Hình 2. Giao thức trao đổi khóa khác (Giao thức 2).

*Bước 3:* $A$ tính $K_A = r_A^{k_B} \bmod p$, $e_B = h(K_A, r_B, r_A, B, A)$ và kiểm tra điều kiện xác minh xem $g^{s_B} y_B^{e_B} = r_B \bmod p$ hay không. Nếu kiểm tra sai, $A$ dừng trao đổi và giao thức thất bại. (Tùy chọn) trường hợp còn lại, $A$ tính giá trị băm $e_A = h(K_A, r_A, r_B, A, B)$, $s_A = k_A - x_A e_A \bmod q$ và gửi $s_A$ tới $B$.

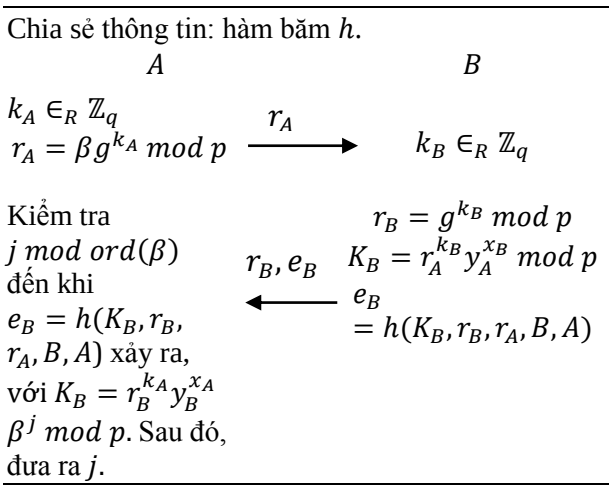
*Bước 4:* (Tùy chọn) $B$ tính $e_A = h(K_B, r_A, r_B, A, B)$ và kiểm tra điều kiện xác minh xem $g^{s_A} y_A^{e_A} = r_A \bmod p$ hay không. Nếu không đúng, thì $B$ dừng trao đổi và giao thức thất bại.

Độ an toàn của hai giao thức trên đều dựa vào tính khó giải của bài toán logarithm rời rạc, với số nguyên tố $p$ mà $p-1$ có một ước nguyên tố $q$ lớn. Việc trao đổi khóa theo hai giao thức này sẽ giúp không để lộ khóa bí mật dài hạn khi các kênh trao đổi là không an toàn. Tuy nhiên, Lim-Lee đã chỉ ra rằng nếu việc thiết kế số nguyên tố $p$ là *không tốt*, cụ thể là $p-1$ có nhiều ước nguyên tố nhỏ, thì kẻ tấn công có thể đơn giản hóa bài toán logarithm rời rạc bằng cách tấn công các giao thức này. Điều này sẽ được chỉ ra ở Mục tiếp theo.

## III. TẤN CÔNG CỦA LIM-LEE ĐỐI VỚI GIAO THỨC 1

Đầu tiên, chúng ta sẽ mô tả tấn công Lim-Lee trên Giao thức 1 đã được mô tả ở phần trước. Ở đây, kẻ tấn công $A$ sẽ cố gắng tìm ra khóa bí mật dài hạn $x_B$ $(0 \leq x_B < q)$ của người dùng trung thực $B$ bằng cách thực hiện giao thức trao đổi khóa với người này một cách không trung thực. Trong Bước 1 của giao thức, sau khi lấy $k_A \in_R \mathbb{Z}_q$, thay vì gửi giá trị $g^{k_A} \bmod p$, $A$ sẽ gửi $r_A = \beta g^{k_A} \bmod p$ tới $B$, với $\beta$ là một phần tử trong $\mathbb{Z}_p$ mà cấp của nó, ký hiệu là $ord(\beta)$, là một ước nhỏ

của $(p-1)/q$. Khi đó người dùng $B$ sẽ tính $r_B$, $e_B$ theo như giao thức và gửi $\{r_B, e_B\}$ tới $A$.




Hình 3. Tấn công Lim-Lee lên giao thức 1.

Với cách thực hiện trung thực của người $B$, thì giá trị $K_B$ thu được như sau:

$$K_B = y_A^{k_B} r_A^{x_B} = (g^{x_A})^{k_B} \left(\beta g^{k_A}\right)^{x_B}$$
$$= r_B^{x_A} y_B^{k_A} \beta^{x_B} = r_B^{x_A} y_B^{k_A} \beta^j \bmod p,$$

trong đó, $j = x_B \bmod ord(\beta)$.

Sau khi nhận được $\{r_B, e_B\}$, $A$ có thể kết thúc giao thức nếu như có yêu cầu trả lời. Vì $A$ đều biết các thành phần công khai $y_B, r_B$ của $B$, nên $A$ có thể dễ dàng tính được thành phần $r_B^{x_A} y_B^{k_A} \bmod p$ trong $K_B$, và có thể tìm được $j = x_B \bmod ord(\beta)$ trong khoảng $O(2^{|ord(\beta)|})$ bước tính, bằng cách kiểm tra phương trình xác minh $e_B = h(K_B, r_B, r_A, B, A)$ với $K_B = r_B^{x_A} y_B^{k_A} \beta^j$ cho tất cả giá trị có thể của $j \in \{0,1,\dots, ord(\beta)-1\}$ (sử dụng phương pháp vét cạn). Với cách tấn công này, $A$ sẽ thu được $j = x_B \bmod ord(\beta)$.

Khi đó, $x_B$ có dạng $x_B = j + ord(\beta).x_\beta$.

Vì $0 < x_B < q$, nên chúng ta có $0 < j + ord(\beta).x_\beta < q$, suy ra $0 \le x_\beta \le \frac{q}{ord(\beta)}$.

Từ $x_\beta = j + ord(\beta).x_\beta$ nên nếu tìm được $x_\beta$ thì sẽ tìm được $x_B$ (do đã biết $j$ và $ord(\beta)$). Mà $0 \le x_\beta < \frac{q}{ord(\beta)}$ nên việc tìm số bit bí mật của $x_B$ được rút gọn về bài toán tìm $(|q| - |ord(\beta)|)$ bit bí mật của $x_\beta$.

Trong giao thức này, kẻ tấn công $A$ có thể lặp lại việc trao đổi khóa với $B$ bằng cách sử dụng các phần tử $\beta_i \bmod p$ có cấp trơn khác nhau mà khả thi cho việc vét cạn. Điều này là thực sự nguy hiểm nếu như $p-1$ có nhiều ước nguyên tố nhỏ (với cỡ khoảng 40 bit), bởi vì, khi đó số bit bí mật của $x_B$ được rút gọn lại chỉ còn là $(|q| - \sum_{\beta_i} |ord(\beta_i)|)$ bit (tính được nhờ sử dụng Định lý Phần dư Trung Hoa), với $\beta_i$ là các phần tử cấp trơn khác nhau được sử dụng trong khi tấn công giao thức. Khi đó, chúng ta có thể tìm được số bit bí mật còn lại của $x_B$ một cách hiệu quả bằng phương pháp Pollard-Rho, Pollard Lambda hay Shank. Hơn nữa, nếu tổng số bit bí mật $\sum_{\beta_i} |ord(\beta_i)|$ bị lộ ra (trong tấn công giao thức) vượt quá $|q|$, chúng ta có thể thu được giá trị của $x_B$ mà chỉ cần sử dụng Định lý Phần dư Trung Hoa.

Lưu ý rằng, tấn công này có thể thiết lập đối với bất kỳ giao thức trao đổi khóa chứng thực nào, miễn là bước chứng thực sử dụng khóa bí mật được chia sẻ (chú ý, chứng thực như vậy chỉ khả thi khi mỗi khóa bí mật của người dùng được dẫn xuất từ khóa bí mật chia sẻ). Từ đó suy ra, hầu như tất cả các giao thức trao đổi khóa cung cấp chứng thực tường minh mà không sử dụng một kênh xác thực riêng biệt có thể bị tổn thương bởi tấn công ở trên (chúng tôi sẽ trình bày tấn công đối với giao thức HMQV ở phần sau).

## IV. TẤN CÔNG CỦA LIM-LEE ĐỐI VỚI GIAO THỨC 2

Giao thức 2 được trình bày ở trên cũng dễ bị tổn thương với kiểu tấn công này. Lưu ý rằng, giao thức này sử dụng một chữ ký số trên khóa bí mật chia sẻ $K_A = K_B = g^{k_A k_B} \bmod p$ nhằm xác thực lẫn nhau. Tuy nhiên, việc sử dụng chung khóa bí mật (ngẫu nhiên) này để xác thực và tính khóa phiên sẽ là một điểm yếu. $A$ có thể lợi dụng điều này để trích một phần thông tin về khóa bí mật $x_B$. Kịch bản của kẻ tấn công $A$ đối với người dùng trung thực $B$ trong giao thức này được mô tả như sau:

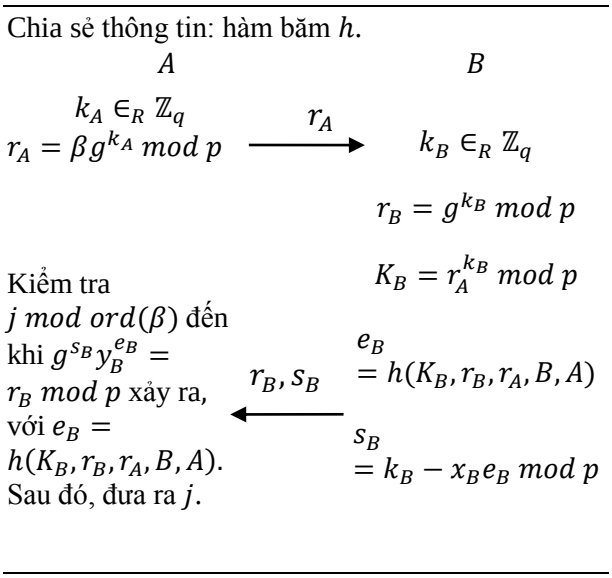
Giống như với Giao thức 1, $A$ chọn ngẫu nhiên $k_A \in_R \mathbb{Z}_q$ rồi gửi $r_A = \beta g^{k_A} \bmod p$ tới $B$ thay vì giá trị $g^{k_A} \bmod p$. Sau khi nhận được cặp $\{r_B, s_B\}$ từ $B$, $A$ thực hiện tìm kiếm vét cạn $k_B \bmod ord(\beta)$ bằng cách sử dụng phương trình xác minh $g^{s_B} y_B^{h(K_A, r_B, r_A, B, A)} = r_B \bmod p$ với $K_A = r_B^{k_A} \beta^{k_B} \bmod p$.

Khi tìm ra giá trị $j = k_B \bmod ord(\beta)$, kết hợp với việc giải phương trình

$$s_B = k_B - x_B e_B \bmod p,$$

với $e_B = h\left(r_B^{x_A} \beta^j \bmod p, r_B, r_A, B, A\right)$,

kẻ tấn công $A$ có thể thu được $|ord(\beta)|$-bit của khóa bí mật $x_B$, vì $x_B = (k_B - s_B) e_B^{-1} \bmod q$.




Hình 4. Tấn công Lim-Lee lên Giao thức 2

Điều này sẽ rút ngắn số bit bí mật của $x_B$ chỉ còn lại $(q - |ord(\beta)|)$ bit. Tuy nhiên, trong trường hợp này, sự lặp lại tấn công với phần tử $\beta'$ có cấp trơn khác (ước nguyên tố nhỏ khác của $p$ – 1) không trợ giúp cho việc tìm nhiều bit bí mật, do mỗi lần sử dụng các $k_B$ khác nhau (nên sẽ không áp dụng được Định lý Phần dư Trung Hoa) và $ord(\beta)$ không chia hết $q$ (nghĩa là $ord(\beta)$ là một ước trơn thực sự khác 1 của $p - 1$).

## V. KHẮC PHỤC TẤN CÔNG CỦA LIM-LEE

Có thể dễ dàng ngăn chặn tấn công của Lim-Lee bằng cách kiểm tra $r_i^q \equiv 1 \bmod q$ trước khi xây dựng khóa bí mật. Tuy nhiên, điều này làm tăng đáng kể số lượng phép tính và như vậy ảnh hưởng đến hiệu quả cài đặt, đặc biệt là việc sử dụng giao thức cho các thiết bị có tài nguyên hạn chế hoặc các ứng dụng trên môi trường truyền thông có tốc độ chậm, băng thông hẹp, ví dụ điển hình như bảo mật thoại qua vệ tinh. Hơn nữa, trong hầu hết các ứng dụng thực tế dùng giao thức SSL/TLS với bộ thư viện mật mã OpenSSL, việc kiểm tra này đã bị bỏ qua. Hàm *DH_check_pub_key* trong file *dh_check.c* chỉ kiểm tra sự hợp lệ của khóa công khai trong khoảng $(1, p - 1)$. Các hàm *ssl3_get_key_exchange* (trong file *s3_clnt.c*) và hàm *ssl3_get_client_key_exchange* (trong file *s3_srvr.c*) đều không kiểm tra khóa công khai nhận được từ đối tác trước khi tính khóa chung.

Một giải pháp tốt hơn và an toàn hơn là chú trọng tới việc sinh số nguyên tố $p$ với tiêu chuẩn chống lại tấn công này. Đó là chọn các số nguyên tố $p$ sao cho $(p - 1)/2q$ có các ước nguyên tố không bé hơn $q$ (việc sử dụng vét cạn theo ý tưởng tấn công Lim-Lee không còn khả thi nữa) hoặc tốt nhất là số nguyên tố an toàn dạng $p = 2q + 1$. Với số nguyên tố an toàn, tấn công của Lim-Lee chỉ thu được 1 bit chẵn-lẻ của khóa bí mật. Chú ý rằng, không một giao thức trao đổi khóa nào có thể bảo vệ bit chẵn-lẻ của khóa bí mật nếu như không kiểm tra cấp của khóa công khai nhận được.

Điều này cũng đúng đối với một giao thức trao đổi khóa 1 chiều sử dụng hiệu quả cho các ứng dụng thư điện tử như sau: A tính $r_A = g^{k_A} \bmod p$ với $k_A \in_R [1, q - 1]$, tính khóa phiên $K = h\left(y_B^{k_A} \bmod p, r_A, d\right)$ với $d$ là nhãn thời gian, mã hóa một thông điệp $m$ thu được bản mã $c = E_{K(m)}$ và gửi $(r_A, d, c)$ cho $B$. Khi đó, $B$ có thể tính khóa phiên $K = h\left(r_A^{x_B} \bmod p, r_A, d\right)$ và giải mã $c$. Trong giao thức này, $A$ có thể gửi $r_A = -g^{k_A} \bmod p$. Nếu $B$ không trả lời hoặc thông báo là một thư rác, thì $A$ biết rằng $x_B$ là lẻ. Tấn công này có thể lặp lại $t$ lần, nếu $2^t | (p - 1)$ thì sẽ để lộ $t$ bit thấp của $x_B$.

Trong hầu hết các tiêu chuẩn đã công bố, số nguyên tố $p$ có dạng $p = q\omega + 1$ với $q$ là một số nguyên tố lớn (cỡ 160 bit trở lên). Tuy nhiên, việc sinh số nguyên tố kiểu này về cơ bản sẽ bị tổn thương bởi tấn công trên. Cụ thể, nếu $p$ được sinh ra có dạng $p - 1 = qw$ mà $w$ có thể được phân tích thành nhiều ước nguyên tố nhỏ (cỡ 40 bit) thì bài toán logarithm có thể được giải trong thời gian hợp lý với hai kiểu tấn công mô tả ở phần trước. Ngay cả khi $w$ chỉ có một số, thay vì nhiều ước nguyên tố nhỏ, thì việc tấn công giao thức có thể giúp đơn giản hóa việc giải bài toán logarithm rời rạc.

Ví dụ như, với số nguyên tố $q$ cỡ 160 bit để giải bài toán logarithm rời rạc $y = g^x \bmod p$, với $g$ là phần tử sinh cấp $q$ trong $\mathbb{Z}_p^*$, cần sử dụng khoảng $O(2^{80})$ phép toán với các phương pháp hiện thời, vì vậy bài toán này rất khó. Nhưng nếu việc tấn công giao thức để lộ ra 40 bit bí mật của $x$, thì bài toán này chỉ cần đến $O(2^{60})$ phép toán để giải, đây là công việc khả thi hơn rất nhiều.

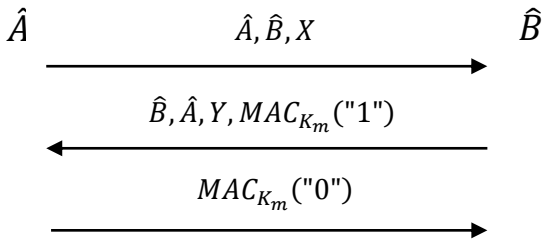
## VI. TẤN CÔNG CỦA LIM-LEE ĐỐI VỚI GIAO THỨC HMQV

Để thấy rõ hơn tấn công của Lim-Lee đối với các giao thức kiểu Diffie-Hellman liên quan đến các ước nhỏ của $p - 1$, trong phần này, chúng tôi mô tả một kiểu tấn công được đưa ra bởi A. Menezes trong [8], áp dụng đối với giao thức HMQV nếu các bên tham gia giao thức không kiểm tra tính hợp lệ của khóa công khai. Các tham số hệ thống cho giao thức trao đổi khóa HMQV được mô tả trong Bảng 1.

BẢNG 1. CÁC THAM SỐ HỆ THỐNG CHO HMQV

| $(p, q, g)$ | Các tham số DLP |
|---|---|
| $H$ | Có thể là một hàm hash hoặc một hàm dẫn xuất khóa |
| $\bar{H}$ | Hàm hash với độ dài đầu ra là $l = \left(\lfloor \log_2 q \rfloor + 1\right)/2$ |
| $\hat{A}, \hat{B}$ | Định danh của thực thể A, B |
| $a, b$ | Khóa bí mật dài hạn của A, B: $a \in_R [1, q-1], b \in_R [1, q-1]$ |
| $A, B$ | Khóa công khai dài hạn của A, B: $A = g^a \bmod p, B = g^b \bmod p$ |
| $x, y$ | Khóa bí mật tạm thờicủa A, B :$x \in_R [1, q-1], y \in_R [1, q-1]$ |
| $X, Y$ | Khóa công khai tạm thời của A, B: $X = g^x \bmod p, Y = g^y \bmod p$ |
| MAC | Thuật toán xác thực thông điệp |
| $d, e$ | $d = \bar{H}(X, \hat{B}), e = \bar{H}(Y, \hat{A})$ |
| $\sigma_{\hat{A}}$ , $\sigma_{\hat{B}}$ | Khóa chia sẻ được tính bởi A, B: $\sigma_{\hat{A}} = (YB^e)^{x+da}$, $\sigma_{\hat{B}} = (XA^d)^{y+eb}$ |
| $K_m$ | Khóa MAC $K_m = H(\sigma_{\hat{A}}, 0) = H(\sigma_{\hat{B}}, 0)$ |
| $K$ | Khóa phiên $K = H(\sigma_{\hat{A}}, 1) = H(\sigma_{\hat{B}}, 1)$ |

Các bước thực hiện giao thức được mô tả như sau: Trước tiên A gửi $(\hat{A}, \hat{B}, X)$ cho B. Khi nhận được, B sẽ kiểm tra $X \neq 0$ hay không, sau đó tính $\sigma_{\hat{B}} = \left(XA^d\right)^{y+eb}$, $K_m = H(\sigma_{\hat{B}}, 0)$ rồi tạo thẻ xác thực $MAC_{K_m}("1")$ và gửi $(\hat{B}, \hat{A}, Y, MAC_{K_m}("1"))$ cho A. Nhận được, A sẽ kiểm tra $Y \neq 0$, tính $\sigma_{\hat{A}} = (YB^e)^{x+da}$, $K_m = H(\sigma_{\hat{A}}, 0)$, kiểm tra thẻ xác thực $MAC_{K_m}("1")$ . Sau đó tạo thẻ xác thực $MAC_{K_m}("0")$ gửi cho B. B nhận và kiểm tra $MAC_{K_m}("0")$ từ A.




Hình 5. Giao thức HMQV 3-pha

*Tấn công giao thức HMQV 3-pha nếu không kiểm tra tính hợp lệ của khóa công khai dài hạn:*

Gọi $G'$ là nhóm nhân aben $\mathbb{Z}_p^*$ cấp $N = p - 1$ và $G = \langle g \rangle$ là nhóm con cấp nguyên tố $q$ của $G'$ ($q$ là ước nguyên tố lớn của $N$). Kẻ tấn công $\hat{A}$ chọn một phần tử $\gamma \in G'$ có cấp $t$ với $t$ là một ước nguyên tố nhỏ nào đó của $N$ $(t \neq q)$. $\hat{A}$ chọn $a \in_R [1, t-1]$ và nhận một chứng thực cho khóa công khai $A = \gamma^a \bmod p$ với giả thiết rằng Cơ quan chứng thực không kiểm tra sự hợp lệ về cấp của $A$. Sau đó A chọn $x \in_R [1, t-1]$, tính $X = \gamma^x \bmod p$ và gửi $(\hat{A}, \hat{B}, X)$ cho B. Khi nhận được, B sẽ chỉ kiểm tra $X \neq 0$ (theo mô tả mặc định, giao thức HMQV không kiểm tra cấp của $X$) và tính $\beta = XA^d$, $\sigma_{\hat{B}} = \beta^{y+eb}$, sau đó tính khóa MAC $K_m = H(\sigma_{\hat{B}}, 0)$, gửi $(\hat{B}, \hat{A}, Y, MAC_{K_m}("1"))$ cho A. Kẻ tấn công A sẽ tìm thông tin về khóa thông qua thẻ $MAC_{K_m}("1")$ như sau: A tính $\beta = XA^d$ sau đó tính $K'_m = H(\beta^{y+ec}, 0$ với $c = 0,1,2,\cdots, t-1$ cho đến khi $MAC_{K'_m}(1) = MAC_{K_m}("1")$. Một khi đã tìm được giá trị $c$ thỏa mãn $K'_m = K_m$, kẻ tấn công sẽ biết được $c = b\,(mod\ t)$. Lặp lại tấn công này với các số nguyên tố $t$ nhỏ khác, giả sử rằng $p - 1$ có đủ các ước nhỏ, sử dụng Định lý Phần dư Trung Hoa kẻ tấn công sẽ thu được khóa bí mật dài hạn $b$.

## VII. THAM SỐ AN TOÀN CHO BÀI TOÁN DLP VÀ VẤN ĐỀ SINH SỐ NGUYÊN TỐ

Các tấn công ở trên cho ta thấy, cần phải sử dụng các số nguyên tố $p$, $q$ trong bài toán DLP với yêu cầu $(p-1)/2q$ là số nguyên tố hoặc mỗi ước nguyên tố của nó phải lớn hơn $q$. Những số nguyên tố như vậy được sinh nhanh hơn nhiều so với việc sinh số nguyên tố an toàn $p = 2q + 1$.

Để sinh một số nguyên tố $p$ thỏa mãn $p = 2qp_1 + 1$, đầu tiên ta chọn một số nguyên tố ngẫu nhiên $p_1$ với độ dài $|p| - |q| - 1$, sau đó sinh số nguyên tố $p = 2qp_1 + 1$ với $q$ là số nguyên tố ngẫu nhiên với độ dài phụ thuộc vào tham số an toàn của hệ thống. Như vậy, ta cần phải tạo ra một số các số nguyên tố $q$ để có thể tìm được $p$. Việc này không yêu cầu nhiều thời gian do kích cỡ của

$q$ thường nhỏ hơn nhiều so với $p$ (chẳng hạn, dựa vào mật độ số nguyên tố $1/\ln(x)$, số các số nguyên tố $q$ cần tìm là khoảng 710 để sinh được một số nguyên tố $p$ có cỡ 1024 bit).

Thời gian tìm các số nguyên tố dạng $p = 2qp_1p_2 \dots p_n + 1$ với các $p_i > q$ là nhanh hơn việc tìm số nguyên tố có dạng ở trên. Đầu tiên ta xác định số $n$ từ bất đẳng thức $l = |p_i| \approx \frac{|p|-|q|-1}{n} \geq |q|$. Sau đó sinh ra một tập hợp các số nguyên tố $p_i$. Giả sử tập hợp này chứa $m$ số nguyên tố độ dài $l$ thì chúng ta có $C_m^n$ các ứng cử cho số nguyên tố $p$. Dựa trên mật độ số nguyên tố, ta có thể làm cho số này đủ lớn để đảm bảo với xác suất cao có thể tìm được một số nguyên tố $p$ từ tập hợp các số nguyên tố $p_i$ này. Chẳng hạn, với số nguyên tố $p$ cỡ 2048 bit và một số nguyên tố $q$ cỡ 256 bit, ta tính được $l \approx 256{,}1, n = 6$. Khi đó ta có thể chọn $m = 15$ để tạo ra 5005 ứng cử viên cho $p$. Khi đó, bằng việc kiểm tra nguyên tố của một số ứng cử viên trong tập này sẽ cho ta một số nguyên tố $p$ cần tìm với xác suất cao.

## VIII. KẾT LUẬN

Bài toán DLP đã được nghiên cứu và ứng dụng nhiều trong thực tế. Tuy nhiên, các tiêu chuẩn tham số cho DLP chỉ xoay quanh độ lớn của $p$ và độ lớn của một ước nguyên tố $q$ của $p - 1$. Tính an toàn của các giao thức dựa trên bài toán logarithm rời rạc sẽ không đảm bảo, nếu việc sinh số nguyên tố $p$ bị xem nhẹ, cụ thể ở đây là $p - 1$ có phần trơn lớn. Như vậy, ta cần tiêu chuẩn cho modulo $p$ như sau: Số nguyên tố $p$ thỏa mãn tính chất $(\mathrm{p} - 1)/2\mathrm{q}$ có các ước nguyên tố không nhỏ hơn $q$. Mục đích của việc xây dựng các số nguyên tố kiểu này là không để lộ đáng kể số bit bí mật từ phần trơn mà làm giảm tính khó của bài toán logarithm rời rạc. Trường hợp an toàn nhất của bài toán DLP là $p - 1 = 2q$, tuy nhiên mật độ những số nguyên tố này là rất nhỏ và nếu sử dụng những số nguyên tố như vậy thì sẽ gặp rủi ro về tính hiệu quả khi phải làm việc trong một nhóm con rất lớn (cỡ $p$).

Bài báo này đưa ra một cách nhìn đầy đủ hơn về tiêu chuẩn cho modulo $p$. Để đưa ra tiêu chuẩn phù hợp và có thể áp dụng trong thực tế, hướng nghiên cứu tiếp theo là phải xây dựng thuật toán sinh các số nguyên tố thỏa mãn tiêu chuẩn an toàn trên và đánh giá mật độ phân bố cũng như tính hiệu quả của thuật toán sinh các số nguyên tố như vậy.

## TÀI LIỆU THAM KHẢO

[1]. D. Pointcheval, J. Stern. "Security proofs for signature schemes", EUROCRYPT'96, vol. 1070, pp. 387-398, 1996.

[2]. D. Bleichenbacher, "Generating ElGamal Signatures Without Knowing the Secret Key", EUROCRYPT'96, vol. 1070, pp. 10-18, 1996.

[3]. C. Lim and P. Lee, "Several practical protocols for authentication and key exchange", Information Processing Letters 53, 1995.

[4]. C. Lim and P. Lee, "A Key Recovery Attack on Discrete Log-based Schemes Using a Prime Order Subgroup", EUROCRYPT'97, pp. 68-73, 1997.

[5]. M. Just and S. Vaudenay. "Authenticated multi-party key agreement", ASIACRYPT'96, 1996.

[6]. A. J. Menezes, M. Qu and S. A. Vanstone, "Some new key agreement protocols providing implicit authentication", In Proc. SAC'95, Carleton Univ., Ottawa, Ontario, May 1995.

[7]. ISO/IEC 11770-3, "Information technology, Security techniques, Key management, Part 3: Mechanisms using asymmetric techniques", 2015 .

[8]. A. Menezes and B. Ustaoglu, "On the importance of public-key validation in the MQV and HMQV key agreement protocols", 2005.

[Phần giới thiệu các tác giả ở trang 38]

SƠ LƯỢC VỀ TÁC GIẢ

**TS. Nguyễn Quốc Toàn**

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

Email: nqtoan@bcy.gov.vn

Tốt nghiệp Học viện Kỹ thuật Mật mã năm 2000. Bảo vệ Thạc sĩ tại Học viện Kỹ thuật Mật mã năm 2007. Bảo vệ Tiến sỹ tại Viện Khoa học và Công nghệ Quân sự, Bộ Quốc Phòng năm 2012.

Hướng nghiên cứu hiện nay: Mật mã khóa công khai (tham số an toàn cho các hệ mật, lược đồ chữ ký số).

**CN. Triệu Quang Phong**

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

E-mail: phongtrieu53@gmail.com

Tốt nghiệp ngành Toán học, Đại học Khoa học tự nhiên - Đại học Quốc gia Hà Nội năm 2014.

Hướng nghiên cứu hiện nay: Mật mã khóa công khai (chứng minh an toàn cho các giao thức mật mã).

**CN. Đỗ Đại Chí**

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

Email: chidd@bcy.gov.vn

Tốt nghiệp ngành Toán học, Đại học Khoa học tự nhiên - Đại học Quốc gia Hà Nội năm 2014.

Hướng nghiên cứu hiện nay: Mật mã khóa công khai (Bài toán DLP).