Xem xét các thuộc tính an toàn của họ giao thức STS

Triệu Quang Phong

***Tóm tắt*— Trên thực tế, các giao thức trao đổi khóa cần đạt được những tính chất an toàn như: tính xác thực khóa ẩn, tính chứng nhận khóa hiện, tính chất an toàn về phía trước, kháng tấn công KCI và kháng tấn công UKS. Đối với họ giao thức STS (Station-to-Station), bốn thuộc tính đầu đã được thảo luận trong [2], trong khi thuộc tính cuối đã được xem xét trong [1]. Trong bài báo này, chúng tôi sẽ đánh giá một cách rõ ràng những thuộc tính an toàn này đối với họ giao thức STS, bao gồm giao thức STS-ENC, STS-MAC và ISO-STS-MAC.**

***Abstract*— In fact, it is expected that, a key exchange protocol achieves some security features such as implicit key authentication, explicit key confirmation, (perfect) forward secrecy, KCI resistance, UKS resistance this. For the family of STS protocols, the first four features were discussed in [2], while the final feature was considered in [1]. In this paper, we will analyze and evaluate these security features for the family of STS protocol (including STS-ENC, STS-MAC, and ISO-STS-MAC protocols).**

***Từ khóa*— giao thức STS-ENC; giao thức STS-MAC; giao thức ISO-STS-MAC; xác thực khóa ẩn; chứng nhận khóa hiện, tấn công UKS; tấn công KCI; an toàn về phía trước.**

***Keywords— STS-ENC protocol; STS-MAC protocol, ISO-STS-MAC protocol; implicit key authentication; explicit key confirmation; UKS attack; KCI attack; (perfect) forward secrecy*.**

I. GIỚI THIỆU

Trong thực tế, các giao thức trao đổi khóa giúp hai (hay nhiều thực thể) thiết lập một khóa chia sẻ bí mật chung. Khóa này có thể được sử dụng để đạt được tính chất mật mã nào đó, chẳng hạn như tính bí mật, tính toàn vẹn dữ liệu hoặc các tính chất khác. Tuy nhiên, việc thiết kế và đánh giá độ an toàn cho các giao thức trao đổi khóa cho đến nay vẫn được xem là một vấn đề khó.

Bài báo được nhận ngày 20/6/2017. Bài báo được gửi cho phản biện thứ nhất vào ngày 6/7/2017 và nhận được ý kiến đồng ý đăng của phản biện thứ nhất vào ngày 28/7/2017. Bài báo được gửi cho phản biện thứ hai vào ngày 8/7/2017 và nhận được ý kiến đồng ý đăng của phản biện thứ hai vào ngày 31/7/2017.

Giao thức trao đổi khóa hai bên theo cơ chế phi đối xứng đầu tiên được đề xuất bởi W. Diffie và M. Hellman trong [6], có tên gọi Giao thức trao đổi khóa Diffie Hellman. Mặc dù sau đó, V. Ooschot trong [4] đã chỉ ra rằng, giao thức này không an toàn trước một tấn công khá cơ bản là tấn công kẻ đứng giữa (man-in-the-middle-attack), nhưng trao đổi khóa Diffie-Hellman lại là cơ sở cho hầu hết các giao thức trao đổi khóa hai bên sau này (như STS [2], MQV [8], SIGMA [7],…). Hơn nữa, các giao thức trong chuẩn quốc tế ISO/IEC 11770-3 [9] cũng được dựa trên cơ chế trao đổi khóa Diffie-Hellman.

Gần đây, Liên Bang Nga đã công bố dự thảo chuẩn về giao thức trao đổi khóa (hai bên), trong đó có hai giao thức Echinacea - 2 và Echinacea - 3. Hai giao thức này được thiết kế trên cơ sở của "cơ chế thỏa thuận khóa số 7" trong ISO/IEC 11770-3 [9], đây thực chất là giao thức ISO-STS-MAC (một biến thể của họ giao thức STS).

Đối với một giao thức trao đổi khóa, người ta mong muốn đạt được những tính chất an toàn như: tính xác thực khóa ẩn, tính chứng nhận khóa hiện, tính chất an toàn về phía trước, kháng tấn công KCI, kháng tấn công UKS, các tính chất này sẽ được định nghĩa trong Mục II. Theo đó, mục đích của bài báo này là phân tích và đánh giá một cách rõ ràng các tính chất an toàn trên đối với họ giao thức STS (bao gồm giao thức STS-ENC, STS-MAC và ISO-STS-MAC) nhằm tạo cơ sở cho việc tìm hiểu sâu hơn về hai giao thức Echinacea - 2 và Echinacea - 3 của Liên Bang Nga. Mặc dù các tính chất trên của STS-MAC và STS-ENC đã được bàn bạc trong [2] (ngoại trừ khả năng kháng UKS được trình bày chi tiết trong [1]), nhưng theo quan điểm của chúng tôi các lập luận này là chưa thực sự rõ ràng.

Phần còn lại của bài báo có bố cục như sau: Mục II trình bày về khái niệm của các thuộc tính an toàn cần xem xét đối với một giao thức trao đổi khóa. Mục III mô tả về ba giao thức trao đổi khóa STS-MAC, STS-ENC và ISO-STS-MAC. Mục IV phân tích đánh giá đối với ba giao thức này theo các thuộc tính an toàn được liệt kê trong Mục II. Cuối cùng, kết luận của bài báo được trình bày trong Mục V.

II. CÁC THUỘC TÍNH AN TOÀN CƠ BẢN

Trong một giao thức trao đổi khóa, gọi A và B là hai *thực thể trung thực* (*honest entity*), nghĩa là, các thực thể hợp pháp, thực hiện các bước của một giao thức một cách chính xác. Hai tính chất trước tiên được quan tâm như sau:

- *Tính chất xác thực khóa ẩn (implicit key authentication):* Một giao thức thỏa thuận khóa được gọi là cung cấp tính chất *xác thực khóa ẩn* (của B đối với A), nếu thực thể A được đảm bảo rằng, không có thực thể nào khác ngoài thực thể B có thể biết được giá trị của khóa bí mật mà anh ta chia sẻ trong phiên cụ thể. Lưu ý, tính chất xác thực khóa ẩn không nhất thiết rằng A được đảm bảo về việc B thực sự sở hữu khóa bí mật chia sẻ. Một giao thức thỏa thuận khóa cung cấp tính chất xác thực khóa ẩn cho cả hai thực thể tham gia trao đổi được gọi là một *giao thức trao đổi khóa có xác thực (hay giao thức AK).*
- *Tính chất chứng thực khóa hiện (explicit key confirmation):* Một giao thức được gọi là cung cấp tính chất *chứng nhận khóa hiện* (của B đối với A) nếu thực thể A được đảm bảo rằng, thực thể B đã thực sự tính ra được khóa bí mật chung.

Nếu cả tính chất xác thực khóa ẩn và chứng thực khóa hiện (của B đối với A) được cung cấp, thì giao thức thiết lập khóa được gọi là cung cấp xác thực khóa hiện (của B đối với A). Một giao thức thỏa thuận khóa mà cung cấp tính chất xác thực khóa hiện cho cả hai thực thể tham gia được gọi là một giao thức thỏa thuận khóa có xác thực kèm theo tính chất chứng nhận khóa (giao thức AKC).

Ngoài ra, một số tính chất an toàn mong đợi khác của các giao thức trao đổi khóa cũng được xác định bao gồm:

- *Tính an toàn về phía trước (Perfect Forward Secrecy):* Nếu các khóa bí mật dài hạn của một hoặc nhiều thực thể bị lộ, thì tính bí mật của các khóa phiên được thiết lập trước đó bởi các thực thể trung thực sẽ không bị ảnh hưởng.
- *Tính kháng tấn công mạo danh thỏa hiệp khóa (Key Compromise Impersonation).* Nếu khóa bí mật dài hạn của thực thể A bị lộ, thì kẻ tấn công không thể mạo danh một thực thể B trung thực nào đó để chia sẻ khóa bí mật với A.
- *Tính kháng tấn công chia sẻ khóa nhưng không rõ đối tác (Unknown Key-Share).* Một tấn công UKS trên giao thức AK hoặc AKC là một tấn công khiến một thực thể A kết thúc và tin tưởng rằng A đã chia sẻ một khóa K với B, trong khi B lại tin tưởng rằng mình chia sẻ khóa K với một thực thể $E \neq A$.

III. HỌ GIAO THỨC STS

Họ giao thức trao đổi khóa STS đã được đề xuất lần đầu tiên bởi V. Oorschot và J. Wiener trong [2]. Đây là một họ giao thức dựa trên cơ chế trao đổi khóa Diffie-Hellman và có xác thực khóa.

Một số ký hiệu được sử dụng khi mô tả và phân tích họ giao thức này được định nghĩa như trong Bảng 1.

BẢNG 1. MỘT SỐ KÝ HIỆU TRONG BÀI

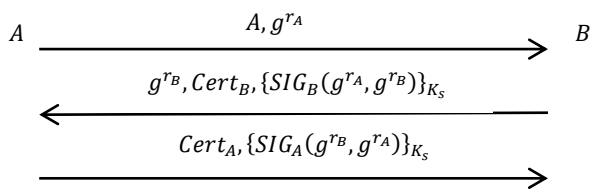
A, B	Các thực thể (bên tham gia) trung thực
E	Bên đối kháng (hay kẻ tấn công)
a	Khóa ký (bí mật dài hạn) của A trong lược đồ chữ ký
P_A	Khóa xác minh (công khai dài hạn) của A trong lược đồ chữ ký
$SIG_A(M)$	Chữ ký của A trên thông điệp M
$Cert_A$	Chứng chỉ của A chứa thông tin định danh của A, khóa công khai P_A và một số thông tin khác có thể có.
$\{M\}_K$	Phép mã hóa thông điệp M bằng cách sử dụng một lược đồ mã hóa khóa đối xứng với khóa K.
$MAC_K(M)$	Mã xác thực thông điệp của M dưới khóa K
G, g, n	Các tham số Diffie-Hellman; g là một phần tử cấp nguyên tố n trong nhóm nhân hữu hạn phần tử G
r_A	Khóa bí mật ngắn hạn của A; $1 \leq r_A \leq n-1$
$g^{r_A r_B}$	Bí mật Diffie-Hellman ngắn hạn của hai bên A và B
K_S	Khóa bí mật chia sẻ giữa hai bên tham gia; hay còn gọi là khóa bí mật chung, hoặc khóa phiên.

A. Giao thức STS cơ bản

Phiên bản đầu tiên của họ giao thức STS trong [2] là giao thức STS cơ bản. Giao thức này được minh họa trong Hình 1. Lưu ý rằng, do STS cơ bản sử dụng cơ chế mã hóa để xác thực khóa, nên giao thức này còn được gọi là STS-ENC (giao thức STS sử dụng mã hóa).

Giả sử A là bên khởi tạo và B là bên phúc đáp, giao thức này được mô tả như sau:

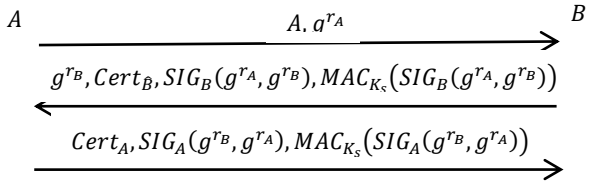
Bên khởi tạo A chọn một số nguyên bí mật ngẫu nhiên $r_A \in [1, n-1]$ và sau đó gửi cho B thông điệp A, g^{r_A}. Sau khi nhận được thông điệp này, B chọn một số nguyên bí mật ngẫu nhiên $r_B \in [1, n-1]$, tính khóa bí mật Diffie-Hellman ngắn hạn $K_S = (g^{r_A})^{r_B}$, và gửi $g^{r_B}, Cert_B, \{SIG_B(g^{r_A}, g^{r_B})\}_{K_S}$ cho A. Sau khi nhận được thông điệp phúc đáp, A tính $K_S = (g^{r_B})^{r_A}$ để giải mã $\{SIG_B(g^{r_A}, g^{r_B})\}_{K_S}$ nhằm thu được chữ ký $SIG_B(g^{r_A}, g^{r_B})$ và sau đó sử dụng $Cert_B$ để xác minh tính hợp lệ của chữ ký này với khóa công khai P_B. Nếu việc xác minh trên là hợp lệ, A gửi cho B thông điệp $Cert_A, \{SIG_A(g^{r_B}, g^{r_A})\}_{K_S}$. Sau khi nhận được thông điệp cuối, B giải mã $\{SIG_A(g^{r_B}, g^{r_A})\}_{K_S}$ bằng khóa K_S và sau đó sử dụng $Cert_A$ để xác minh tính hợp lệ của chữ ký $SIG_A(g^{r_A}, g^{r_B})$ với khóa P_A. Nếu việc xác minh của B đưa ra kết quả hợp lệ, việc trao đổi khóa giữa A và B được xem là thành công và khóa bí mật chia sẻ của họ là K_S.



Hình 1. Giao thức STS cơ bản

B. Giao thức STS-MAC

Ngoài giao thức STS-ENC, trong [2] đã đề cập đến một biến thể với tên gọi STS-MAC. Giao thức này cũng cung cấp tính chứng nhận khóa hiện bằng cách sử dụng hàm MAC (an toàn) thay vì sử dụng hàm mã hóa (an toàn) như STS-ENC. Phân tích cho việc STS-MAC đạt được tính chất xác thực khóa ẩn và chứng nhận khóa hiện được thực hiện một cách tương tự như STS-ENC. Giao thức STS-MAC được mô tả theo Hình 2.

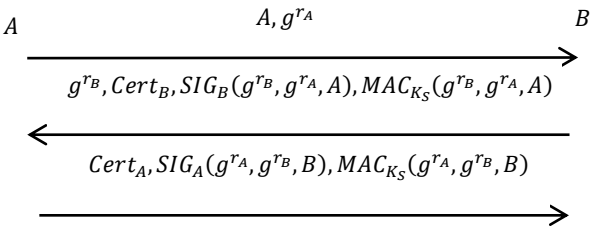


Hình 2. Giao thức STS-MAC

Theo [1], STS-MAC có thể được ưu tiên hơn so với STS-ENC trong nhiều kịch bản thực tế vì việc tồn tại các hạn chế trong cách sử dụng phép mã hóa an toàn. Hơn nữa, việc sử dụng hàm mã hóa để cung cấp chứng nhận khóa trong STS-ENC là điều đáng nghi ngờ - theo cách truyền thống, mục đích cơ bản của phép mã hóa là cung cấp tính bí mật và nếu một lược đồ mã hóa được sử dụng để chứng minh việc sở hữu một khóa, thì nó được biểu diễn bởi phép giải mã, chứ không phải bởi phép mã hóa. Một lợi thế của STS-ENC so với STS-MAC là giao thức này có thể phục vụ cho các trao đổi ẩn danh.

C. Giao thức ISO-STS-MAC

Chuẩn quốc tế ISO 11770-3 đã đưa ra hai biến thể tương ứng của STS-ENC và STS-MAC - chúng được mô tả như "cơ chế thỏa thuận khóa 7" trong [9]. Tuy nhiên, như đã đề cập trong mục trước, STS-MAC dường như được ưu tiên hơn STS-ENC. Do đó, chúng ta chỉ xem xét biến thể ISO của STS-MAC - hay được gọi là giao thức ISO-STS-MAC (xem Hình 3).



Hình 3. Giao thức ISO-STS-MAC

Khác với mô tả gốc cho giao thức STS-MAC, các định danh của người nhận được bao gồm trong các chữ ký của người gửi trong ISO-STS-MAC. Điều này đã được thực hiện để phù hợp với các cơ chế xác thực thực thể trong 9798-3 được đề cập trong [10]. Một điểm khác nhau nữa giữa ISO-STS-MAC và STS-MAC là, trong giao thức ISO-STS-MAC, thuật toán MAC được áp dụng cho thông điệp được ký, chứ không phải là cho chữ ký của thông điệp đó. Tuy nhiên, cần lưu ý rằng, các thiết lập khóa bí mật chia sẻ giữa hai bên tham gia trong ISO-STS-MAC là tương tự như trong STS-MAC.

IV. XEM XÉT CÁC THUỘC TÍNH AN TOÀN ĐỐI VỚI HỌ GIAO THỨC STS

A. Tính chất xác thực khóa ẩn

Mục này, phân tích về các tính chất xác thực của giao thức STS-ENC dưới điều kiện các nguyên thủy chữ ký số, hàm mã hóa là an toàn và giả thiết về độ khó của bài toán Diffie-Hellman tính toán CDH (Computational Diffie-Hellman).

Xem xét hai thực thể trung thực A và B, giả sử rằng, A kết thúc phiên trao đổi với khóa chia sẻ $K_S = g^{r_A r_B}$ và tin tưởng rằng B là người tham gia trao đổi với mình, khi đó không có một thực thể

nào khác ngoài B có thể tính ra khóa K_s. Thật vậy, chúng ta xem xét 2 trường hợp tấn công như sau.

Trường hợp đầu, E là kẻ tấn công bị động, khi đó E chỉ "nghe trộm" các thông tin được truyền đi giữa hai bên (có nghĩa A và B thực sự trao đổi với nhau). Như vậy, khi hàm mã hóa là an toàn, thì thông tin duy nhất mà E có thể khai thác để tính ra giá trị K_S chỉ là các giá trị công khai g^{r_A}, g^{r_B}. Tuy nhiên, nếu từ hai giá trị này mà E có thể tính ra được giá trị $g^{r_A r_B}$, chúng ta có thể xây dựng từ một kẻ tấn công để giải bài toán Diffie-Hellman tính toán (CDH), điều này là mâu thuẫn với giả thiết về độ khó của bài toán CDH.

Trường hợp thứ hai, E là kẻ tấn công chủ động, nghĩa là E có khả năng thay đổi hoặc sửa chữa thông điệp mà A nhận được. Theo độ khó của bài toán CDH, thì E có khả năng tính ra được khóa K_S chỉ khi E là người sinh ra g^{r_B}. (Vì trong trường hợp E lặp lại thông điệp của B chứa g^{r_B} trong một phiên khác để gửi cho A thì E không thể biết gì ngoài g^{r_B}). Tuy nhiên, trong trường hợp này thì E cần có khả năng tạo ra chữ ký của B trên thông điệp mới g^{r_A}, g^{r_B} để A chấp nhận tính hợp lệ của thông điệp mà mình nhận được. Do r_A được A sinh mới sau mỗi phiên, nên khi E (chứ không phải B) sinh ra r_B trong phiên đang xét, nên việc B đã từng ký lên g^{r_A}, g^{r_B} là không đáng kể. Điều này sẽ là mâu thuẫn với giả thiết về độ an toàn của lược đồ chữ ký số.

Từ kết quả của hai trường hợp trên, chúng ta thấy STS-ENC cung cấp tính chất xác thực khóa ẩn của B đối với A; lập luận tương tự ta cũng có STS-ENC cung cấp tính chất xác thực khóa ẩn của A đối với B. Do đó, giao thức này cung cấp tính chất xác thực khóa ẩn (lẫn nhau).

Một cách tương tự, STS-MAC và ISO-STS-MAC cũng cung cấp tính chất này.

Nhận xét 1. Yêu cầu hai bên tham gia A và B là trung thực khi xem xét tính xác thực khóa ẩn (lẫn nhau) là cần thiết. Thật vậy, trong trường hợp mà một trong hai bên bị mua chuộc, thì tính chất trên không thể đạt được đối với các giao thức trao đổi khóa nói chung và giao thức STS-ENC nói riêng. Cụ thể, xét trong trường hợp của giao thức STS-ENC, nếu A (đóng vai trò là bên khởi tạo) tin tưởng chia sẻ khóa K với bên tham gia B không trung thực - nghĩa là B đã bị mua chuộc bởi kẻ tấn công E để ký lên bất cứ thông điệp gì mà E yêu cầu, thì tính xác thực khóa ẩn của B đối với A không được đảm bảo. Thật vậy, khi nhận được thông điệp khởi tạo g^{r_A} của A, E sinh giá trị r_E, tính $K_S = (g^{r_A})^{r_E}$ và yêu cầu B ký lên thông điệp g^{r_A}, g^{r_E} (chữ ký $SIG_B(g^{r_A}, g^{r_E})$) sau đó mã hóa chữ ký nhận được với khóa K_S. Tiếp theo E gửi cho A thông điệp $g^{r_E}, Cert_B, \{SIG_B(g^{r_A}, g^{r_E})\}_{K_S}$. Hiển nhiên rằng, A sẽ tin tưởng thông điệp E gửi như thông điệp được gửi từ B và tính ra khóa chia sẻ $K_S = (g^{r_E})^{r_A}$. Điều này cho thấy rằng, việc xác thực khóa ẩn của B đối với A đã bị vi phạm vì E đã tính được khóa bí mật mà A tin tưởng chia sẻ với B.

Chúng ta hoàn toàn thu được kết quả như vậy đối với giao thức STS-MAC và ISO-STS-MAC bằng lập luận tương tự.

B. Tính chất chứng nhận khóa hiện

Ở trên, chúng ta đã chỉ ra rằng, STS-ENC thỏa mãn tính chất xác thực khóa ẩn dưới giả thiết độ khó của bài toán CDH và độ an toàn của lược đồ chữ ký mà giao thức này sử dụng. Ở đây, chúng ta sẽ chỉ ra rằng, nếu A hoàn tất quá trình trao đổi với khóa bí mật K_S và tin tưởng rằng mình trao đổi với B thì B đã thực sự tính được khóa K_S. Thật vậy, do chỉ có B (ngoài A) mới có khả năng tính được khóa K_S (theo tính xác thực khóa ẩn), nên theo độ an toàn của hàm mã hóa, chỉ có B mới có khả năng tính được giá trị mã hóa với khóa K_S mà A chấp nhận là hợp lệ và chỉ khi khóa K_S được tính ra. Điều này chỉ ra rằng, khi A hoàn tất quá trình trao đổi với khóa bí mật K_S và tin tưởng rằng mình trao đổi với B, thì A được đảm bảo rằng B đã thực sự tính được khóa K_S.

Như vậy, STS-ENC cung cấp tính chất chứng thực khóa hiện của B đối với A. Bằng lập luận tương tự ta cũng có STS-ENC cung cấp tính chất chứng thực khóa hiện của A đối với B. Do đó, giao thức này cung cấp tính chất chứng thực khóa hiện (lẫn nhau).

Một cách tương tự, STS-MAC và ISO-STS-MAC cũng cung cấp tính chất này.

C. Tấn công UKS lên giao thức STS

Trong mục này, chúng ta sẽ xem xét khả năng kháng tấn công UKS (Unknown Key Share) của các giao thức STS-MAC, STS-ENC, ISO-STS-MAC. Cần lưu ý rằng, nếu một giao thức AK hoặc AKC bị tổn thương trước một tấn công UKS (trong đó E là một thực thể không trung thực), thì điều này không mâu thuẫn với tính chất xác thực khóa ẩn của giao thức. Bởi theo định nghĩa, việc cung cấp tính chất xác thực khóa ẩn chỉ được xem xét trong trường hợp B được liên kết với một thực thể trung thực trong giao thức, tuy nhiên E không phải là thực thể trung thực.

Một lưu ý khác là, giao thức ISO-STS-MAC có thể kháng được mọi dạng tấn công UKS dựa trên độ an toàn của hàm MAC và dưới giả thiết DDH (Decisional Diffie-Hellman). Thật vậy, giả sử tồn tại một kẻ tấn công khiến cho A tin tưởng chia sẻ khóa K_S với bên tham gia B, trong khi B lại tin tưởng mình chia sẻ khóa K_S với $E \neq A$. Khi đó, giá trị MAC dưới khóa K_S mà B gửi đi là $MAC_{K_S}(\dots, E)$ trong khi giá trị MAC mà kẻ tấn công cần chuyển cho A là $MAC_{K_S}(\dots, A)$. Lưu ý rằng, chỉ có hai giá trị MAC với khóa K_S được tính là $MAC_{K_S}(\dots, A)$ và $MAC_{K_S}(\dots, B)$ (do các khóa ngắn hạn mà các bên tham gia được chọn ngẫu nhiên, nên khóa chia sẻ trong mỗi phiên trao đổi là độc lập). Điều này có nghĩa, kẻ tấn công phải có khả năng giả mạo giá trị MAC với khóa K_S trên thông điệp mới. Điều này làm mâu thuẫn với giả thiết về độ an toàn của hàm MAC hoặc giả thiết về độ khó của bài toán DDH.

Do vậy trong phần tiếp theo, tác giả chỉ xem xét các tấn công UKS lên hai giao thức STS-MAC và STS-ENC, bao gồm:

1. Các tấn công UKS thay đổi khóa công khai lên STS-MAC và STS-ENC

Các tấn công này đã được phân tích trong [1] và [7]. Kịch bản của tấn công này là các bên tham gia có thể đăng ký khóa công khai mà bỏ qua việc chứng minh hiểu biết về khóa bí mật tương ứng. (Lưu ý rằng, trong khi "bằng chứng về quyền sở hữu" như vậy được yêu cầu bởi một số nhà cung cấp chứng thực số (Certificate Authority - CA) cho việc lưu hành một chứng thư số, đây không phải là yêu cầu chung cho các chứng nhận khóa công khai; đặc biệt nó không được thỏa mãn trong nhiều kịch bản "phân phối ngoài luồng", các web đáng tin cậy,….

Tấn công UKS thay đổi khóa công khai trên giao thức STS-ENC: Trong mô tả tấn công này, A là bên khởi tạo và B là bên phúc đáp. Chúng ta xem xét tấn công hai trường hợp sau:

Trong tấn công UKS chống lại bên phúc đáp, kẻ tấn công E đăng ký khóa công khai P_A như khóa công khai của chính E; nghĩa là, $P_E = P_A$ và thực hiện các bước như trong Hình 4. Ký hiệu $A \hookrightarrow B$ có ý nghĩa rằng A đã chuyển một thông điệp được chỉ định gửi tới B, tuy nhiên thông điệp này bị chặn bởi kẻ tấn công E và có thể không được phân phát tới B.

Vì $P_A = P_E$, chúng ta có $SIG_A(g^{r_A}, g^{r_B}) = SIG_E(g^{r_A}, g^{r_B})$. Theo cách như vậy B chấp nhận khóa K_S và tin rằng K_S được chia sẻ với E, trong khi thực tế khóa này được chia sẻ với A. Lưu ý rằng E không biết giá trị của K_S.

1. $A \hookrightarrow B: g^{r_A}$
2. $E \rightarrow B: g^{r_A}$
3. $B \rightarrow E: g^{r_B}, Cert_B, \{SIG_B(g^{r_A}, g^{r_B})\}_{K_S}$
4. $E \rightarrow A: g^{r_B}, Cert_B, \{SIG_B(g^{r_A}, g^{r_B})\}_{K_S}$
5. $A \hookrightarrow B: Cert_A, \{SIG_A(g^{r_B}, g^{r_A})\}_{K_S}$
6. $E \rightarrow B: Cert_E, \{SIG_A(g^{r_B}, g^{r_A})\}_{K_S}$

Hình 4. Tấn công UKS thay đổi khóa công khai trên STS-ENC

Tấn công UKS thay đổi khóa công khai trên giao thức STS-MAC: Theo một cách tương tự, tấn công UKS thay đổi khóa công khai trên giao thức STS-ENC có thể áp dụng cho giao thức STS-MAC.

Ngăn chặn các tấn công UKS thay đổi khóa công khai:

- Chúng ta luôn ngăn chặn được kiểu tấn công này bằng cách yêu cầu các thực thể chứng minh việc sở hữu khóa bí mật tương ứng với khóa công khai trong quá trình chứng thực khóa.
- Đối với STS-ENC, một cách khác để kháng lại tấn công UKS thay đổi khóa công khai trên giao thức này là mã hóa các chứng chỉ cùng chữ ký của các bên tham gia dưới khóa bí mật chia sẻ.

2. Các tấn công UKS trực tuyến trên STS-MAC

Như được chỉ ra trong phần trước, các tấn công UKS thay đổi khóa công khai có thể được ngăn chặn bằng cách yêu cầu các thực thể chứng minh sự sở hữu khóa bí mật tương ứng với khóa công khai trong quá trình chứng thực khóa. Tuy nhiên, phương pháp này không giúp STS ngăn chặn được các tấn công UKS trực tuyến được giới thiệu trong [1]. Các giả thiết sau được đặt ra để các tấn công là hiệu quả.

- Lược đồ chữ ký được sử dụng trong STS có tính chất lựa chọn khóa chữ ký kép (DSKS). Giả sử rằng, P_A (khóa công khai của A) và chữ ký $SIG_A(M)$ của A trên thông điệp M được cho biết trước, thì kẻ tấn công có thể lựa chọn một cặp khóa (P_E, e) thỏa mãn tính chất $SIG_A(M)$ cũng là chữ ký của E trên thông điệp M. Trong [1] đã đưa ra các bằng chứng tính đúng đắn của giả thiết này trên các lược đồ chữ ký RSA, Rabin, ElGamal, DSA, ECDSA,... Ví dụ về tính chất DSKS trên DSA được thể hiện trong Bảng 2 ở Mục Kết luận.

- E có khả năng tạo ra khóa công khai của mình được chứng nhận trong một lần chạy của giao thức STS. Giả thiết này là hợp lý, ví dụ, trong các trường hợp mà ở đó việc chuyển các thông điệp bị làm chậm là bình thường và CA cho phép đăng ký ở chế độ trực tuyến.

Dựa trên tính chất DSKS của lược đồ chữ ký số (RSA, Rabin, ElGamal, DSA, ECDSA,...), tấn công UKS trực tuyến lên STS-MAC được mô tả như sau:

Mô tả tấn công UKS trực tuyến (lên bên phúc đáp): Tấn công này trên STS-MAC là tương tự tấn công thay đổi khóa công khai chống lại bên khởi tạo như đã đề cập ở trước (tấn công đó phù hợp với chế độ ngoại tuyến). Khác biệt ở đây là kẻ tấn công thay vì lập lại khóa công khai của bên khởi tạo A như khóa công khai của mình, thì sẽ chọn cặp khóa công khai- bí mật (P_E, e) thỏa mãn $SIG_E(g^{r_A}, g^{r_B}) \equiv SIG_A(g^{r_A}, g^{r_B})$ sau khi bên khởi tạo A gửi thông điệp cuối cùng.

Chúng ta cũng thu được một tấn công UKS trực tuyến lên bên khởi tạo theo cách tương tự.

Lưu ý rằng, đối với STS-ENC thì tấn công UKS trực tuyến dựa trên tính chất DSKS của các lược đồ chữ ký được sử dụng sẽ không khả thi. Bởi kẻ tấn công không thể thu được chữ ký nào của một trong hai bên tham gia (trung thực) để thực hiện tấn công DSKS.

Dưới đây, chúng ta sẽ xem xét một số biện pháp có thể thực hiển để tránh được kiểu tấn công UKS trực tuyến trên STS-MAC được đề xuất trong [1].

Ví dụ về tấn công DSKS trên lược đồ chữ ký số DSA:

- Các tham số miền: Các số nguyên tố p, q thỏa mãn $q|(p-1)$ và một phần tử $\alpha \in \mathbb{Z}_p^*$ cấp q. Thông thường p có độ dài 1024 bit và q có độ dài 160 bit.
- Cặp khóa: khóa bí mật dài hạn của A là a, $1 \le a \le q-1$. Một khóa công khai là $P_A = (p, q, \alpha, y)$, ở đó $y = \alpha^a$.
- Sinh chữ ký: Để ký một thông điệp M, A chọn một số nguyên ngẫu nhiên $k \in [1, q-1]$, và tính $m = H(M)$, $r = (\alpha^k \bmod p) \bmod q$, và $s = k^{-1}(m + ar) \bmod q$. Chữ ký của A trên M là (r, s).
- Xác minh chữ ký: Với một bản sao khóa công khai của A, người ta có thể xác minh chữ ký (r, s) trên thông điệp M có phải của A hay không bằng cách tính $m = H(M)$, $u_1 = s^{-1}m \bmod q$, $u_2 = s^{-1}r \bmod q$ và xác minh rằng $r = (\alpha^{u_1} y^{u_2} \bmod p) \bmod q$ có đúng hay không.
- Hoạt động của kẻ tấn công: Khóa công khai P_A và chữ ký (r, s) của A trên thông điệp M. E chọn một số nguyên ngẫu nhiên $e \in [1, q-1]$ sao cho $t := ((u_1 + eu_2) \bmod q) \neq 0$. E sau đó tính $r_1 = \alpha^{u_1} y^{u_2} \bmod p$ và $\bar{\alpha} = r_1^{t^{-1} \bmod q} \bmod p$, và tạo khóa công khai $P_E = (p, q, \bar{\alpha}, \bar{y})$, ở đó $\bar{y} = \bar{\alpha}^e \bmod p$. Lưu ý rằng $ord(\bar{\alpha}) = q$, vì vậy P_E là khóa công khai DSA hợp lệ. Hơn nữa, (r, s) là chữ ký hợp lệ của E trên thông điệp M vì:

$$(\bar{\alpha}^{u_1} \bar{y}^{u_2} \bmod p) \bmod q$$

$$= (\bar{\alpha}^{u_1 + eu_2} \bmod p) \bmod q$$

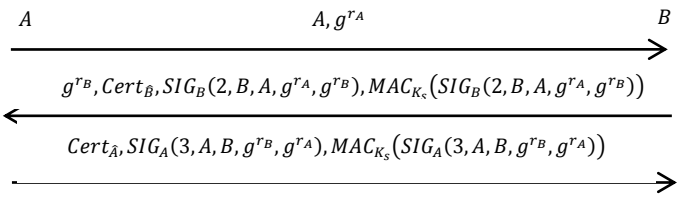
$$= (\bar{\alpha}^{t} \bmod p) \bmod q$$

$$= (r_1 \bmod p) \bmod q$$

$$= (\alpha^{u_1} y^{u_2} \bmod p) \bmod q = r.$$

Ngăn chặn tấn công UKS trực tuyến. Trong tấn công UKS trực tuyến, kẻ tấn công biết khóa bí mật e tương ứng với khóa công khai P_E đã chọn. Vì vậy, không giống như trường hợp của tấn công thay đổi khóa công khai, các tấn công trực tuyến không thể bị ngăn chặn bằng cách yêu cầu các thực thể chứng minh sự sở hữu các khóa bí mật tương ứng với khóa công khai của họ trong quá trình chứng thực.

- Nếu A gửi chứng chỉ $Cert_A$ trong thông điệp đầu tiên thay vì trong thông điệp cuối, thì tấn công UKS trực tuyến lên bên phúc đáp không thể thực hiện. Tuy nhiên, tấn công UKS trực tuyến lên bên khởi tạo vẫn thành công.
- Nếu các chứng chỉ được trao đổi trước khi chạy giao thức, thì các tấn công UKS trực tuyến sẽ thất bại.
- Đưa định danh của người gửi và người nhận cũng như chỉ số luồng vào trong thông điệp được ký (Hình 5) để ngăn chặn các tấn công UKS trực tuyến. Giao thức sửa đổi được biểu diễn như sau:



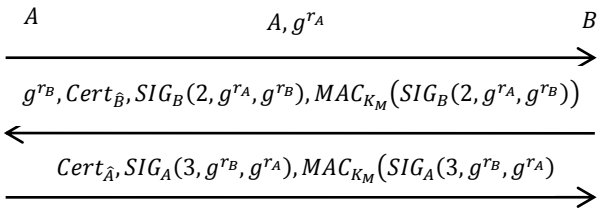
Hình 5. Biến thể 1 của STS-MAC

- Thay vì đưa định danh của các thực thể vào thông điệp ký, người ta có thể đưa chúng vào trong hàm dẫn xuất khóa, mục đích là để dẫn xuất khóa chia sẻ từ bí mật chia sẻ $g^{r_A r_B}$. Sửa đổi này được biểu diễn trong Hình 6, trong các khóa được tính theo công thức

$$K_S||K_M = H(g^{r_A r_B}, A, B)$$

hoặc

$$K_S = H(01, g^{r_A r_B}, A, B),$$
$$K_M = H(10, g^{r_A r_B}, A, B).$$



Hình 6. Biến thể 2 của STS-MAC

Theo [1], trong tất cả phương pháp được đưa ra thì biện pháp đưa định danh của các bên tham gia vào trong thông điệp ký là hợp lý nhất để kháng lại các tấn công UKS trên giao thức STS-MAC, vì nó phù hợp với các quan điểm về mặt hiệu năng và đảm bảo về mặt lý thuyết khi sử dụng các nguyên thủy mật mã trong thời điểm đó. Tuy nhiên, trong [3] chỉ ra rằng biện pháp này không thể giúp STS-MAC kháng lại được tấn công UKS trực tuyến trên giao thức STS MAC. Cụ thể, công trình này chỉ ra rằng, các giao thức STS-MAC sửa đổi theo cách đưa định danh vào trong chữ ký (Hình 4) đều bị tổn thương trước tấn công UKS trực tuyến khi lược đồ chữ ký được sử dụng là El Gamal.

Mệnh đề 1 [3]***:*** *Giả sử rằng lược đồ chữ ký trong giao thức STS-MAC sửa đổi (Hình 4) là El Gamal. Gọi* (p, α, y) *là các tham số công khai trong lược đồ chữ ký El Gamal của bên khởi tạo A (ở đó p là số nguyên tố),* $y = \alpha^a$ *với a là khóa ký và g là phần tử sinh của nhóm nhân* $\mathbb{Z}_p^*$*. Giả sử rằng chữ ký trên* $(3, \alpha^{r_A}, \alpha^{r_B}, A, B)$ *trong thông điệp cuối cùng mà A gửi cho B là*

$$(r, s) = \left(\alpha^k, k^{-1}(H(3, g^{r_A}, g^{r_B}, A, B) - ar)\right),$$

với H là hàm băm. Khi đó E có thể lựa chọn khóa công khai của chính mình trong lược đồ chữ ký El Gamal để B xác minh chữ ký (r, s) *là hợp lệ trên thông điệp* $(3, \alpha^{r_A}, \alpha^{r_B}, E, B)$.

Mệnh đề này đã được chứng minh trong [3].

Kết quả trên chỉ ra rằng, nếu lược đồ chữ ký được sử dụng trong giao thức STS-MAC là lược đồ El Gamal, thì phương pháp số 3 ở trên để ngăn chặn UKS trực tuyến cho giao thức này là không có tác dụng. Tuy nhiên, khác với lược đồ chữ ký El Gamal, các lược đồ chữ ký như: DSA, ECDSA, GOST R 34.10-2012,... đều được xây dựng trên nhóm con có phần tử sinh cấp q, thay vì trên cả nhóm cơ sở. Do đó, việc xem xét kết quả trong [3] có còn đúng không - khi các lược đồ chữ ký DSA, ECDSA, GOST R 34.10-2012,... được thay thế cho lược đồ El Gamal là thực sự cần thiết. Dưới đây, chúng tôi chỉ ra rằng nhận định trên vẫn còn đúng khi lược đồ chữ ký được sử dụng trong STS-MAC là DSA.

Mệnh đề 2: *Giả sử rằng lược đồ chữ ký trong giao thức STS-MAC sửa đổi (Hình 4) là DSA. Gọi* (p, q, α, y) *là các tham số công khai trong lược đồ chữ ký DSA của bên khởi tạo A (ở đó* p, q *là các số nguyên tố thỏa mãn* $q|(p-1)$, $y = \alpha^a$ *với* a *là khóa ký và* g *là phần tử cấp* q *của nhóm nhân* $\mathbb{Z}_p^*$*. Giả sử rằng, chữ ký trên* $(3, \alpha^{r_A}, \alpha^{r_B}, A, B)$ *trong thông điệp cuối cùng mà A gửi cho B là :*

$$(r, s) = \left(\alpha^k \bmod q, k^{-1}(H(3, g^{r_A}, g^{r_B}, A, B) + ar) \bmod q\right),$$

với H là hàm băm. Khi đó, E có thể lựa chọn khóa công khai của chính mình trong lược đồ chữ ký DSA để B xác minh chữ ký (r, s) *là hợp lệ trên thông điệp* $(3, \alpha^{r_A}, \alpha^{r_B}, E, B)$.

Chứng minh :

Đầu tiên, E tính $h' = H(3, g^{r_A}, g^{r_B}, E, B)$. Chú ý rằng, trong lược đồ chữ ký DSA chúng ta có $\gcd(r, q) = 1$ và $\gcd(s, q) = 1$. Sau đó, E chọn ngẫu nhiên số nguyên $e \in [1, \dots, q-1]$ sao cho $t = h' + er \bmod q \neq 0$ và sử dụng e khóa bí mật dài hạn của mình. Từ đó E tính $r_1 = \alpha^{H(3, g^{r_A}, g^{r_B}, A, B)s^{-1}} y^{rs^{-1}} \bmod p$, $\bar{\alpha} = r_1^{st^{-1} \bmod q} \bmod p$ và thiết lập khóa công khai mới cho mình là $(p, q, \bar{\alpha}, \bar{y})$, với $\bar{y} = \bar{\alpha}^e$.

B sau đó xác minh chữ ký (r, s) với tham số công khai $(p, q, \bar{\alpha}, \bar{y})$. Sau khi tính $h' = h(3, g^{r_A}, g^{r_B}, E, B)$, B sẽ kiểm tra xem đẳng thức $r = \left(\bar{\alpha}^{h's^{-1}} \bar{y}^{rs^{-1}} \bmod p\right) \bmod q$ hay không. Kết quả đưa ra là đúng do:

$$\begin{aligned}
&\bar{\alpha}^{(-h')} \bar{y}^r r^s \bmod p \\
&\quad = \left(\bar{\alpha}^{h's^{-1}} \bar{\alpha}^{ers^{-1}} \bmod p\right) \bmod q \\
&\quad = \left(\bar{\alpha}^{h's^{-1} + ers^{-1}} \bmod p\right) \bmod q \\
&\quad = \left(\bar{\alpha}^{ts^{-1}} \bmod p\right) \bmod q \\
&\quad = (r_1 \bmod p) \bmod q \\
&= \left(\alpha^{h(3, g^{r_A}, g^{r_B}, A, B)s^{-1}} y^{rs^{-1}} \bmod p\right) \bmod q \\
&= r.
\end{aligned}$$

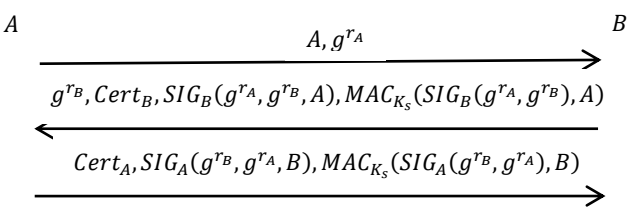
Do đó, ta có điều phải chứng minh.

Nhận xét 2. Theo cách tương tự, chúng ta thấy rằng, kết quả trên vẫn đúng nếu ta thay lược đồ chữ ký DSA trong STS-MAC bằng các lược đồ chữ ký ECDSA và họ các lược đồ chữ ký GOST. Hệ quả là, giải pháp đưa định danh của các bên tham gia vào trong chữ ký để giúp STS-MAC tránh tấn công UKS trực tuyến (như các biện pháp 3) dường như không có tác dụng.

Nhận xét 3. Như chúng ta thấy, có thể ngăn chặn các tấn công UKS trực tuyến trên giao thức STS-MAC bằng cách:

- Các bên tham gia trao đổi chứng chỉ trước khi thực hiện giao thức. Tuy nhiên, cách này có thể làm gia tăng số bước trao đổi giữa hai bên tham gia. Nhìn chung, đây không phải là phương pháp ngăn chặn các tấn công UKS, vì nó không giúp STS-MAC tránh được tấn công UKS ngoại tuyến.
- Có thể tính khóa bí mật chia sẻ giữa hai bên tham gia bằng cách đưa định danh của họ và bí mật được chia sẻ $g^{r_A r_B}$ vào hàm dẫn xuất khóa.
- Có thể sử dụng phương án của ISO-STS-MAC.

Tuy nhiên, theo quan điểm cài đặt mà nói, việc đưa định danh của các bên tham gia vào hàm dẫn xuất khóa có thể khiến việc cài đặt trở nên phức tạp hơn, trong khi phương án của ISO-STS-MAC lại có thể yêu cầu việc tính giá trị hàm MAC trên một thông điệp có độ dài bit lớn. Do đó, để cải thiện điều này chúng tôi thấy phương án sau sửa đổi cho STS-MAC dưới đây có thể phù hợp hơn cho việc cài đặt.



Hình 7. Giao thức sửa đổi đề xuất

Chúng ta có thể thấy giao thức trong Hình 6 là sự lai ghép giữa ISO-STS-MAC và STS-MAC. Việc kết hợp như vậy giúp cho giao thức này kế thừa tất cả các tính chất của ISO-STS đã được phân tích trong bài báo này (kể cả kháng tấn công UKS), trong khi chỉ yêu cầu tính các giá trị MAC trên những thông điệp có độ dài ngắn hơn đối với ISO-STS-MAC (bởi trên thực tế một chữ ký của một thông điệp thường có độ dài bit ngắn hơn đáng kể so với thông điệp đó).

D. Các thuộc tính an toàn khác

Tiếp theo, tác giả trình bày kết quả sẽ đánh giá một số thuộc tính an toàn khác của các giao thức STS-MAC, STS-ENC và ISO-STS-MAC. Lưu ý rằng, chúng ta cần đến các giả thiết về độ khó của bài toán Diffie-Hellman quyết định (DDH), độ an toàn của lược đồ chữ ký số và độ an toàn của hàm MAC để thực hiện các đánh giá này.

Kháng tấn công KCI (Key Compromise Impersonation). Giả sử rằng trong giao thức ISO-STS-MAC, kẻ tấn công biết khóa bí mật dài hạn của bên tham gia A và có khả năng mạo danh bên tham gia B trung thực nào đó để tham gia trao đổi với A. Khi đó, kẻ tấn công E đã gửi cho A chữ ký của B trên thông điệp chứa giá trị *nonce* mới mà A sinh ra trong khi E không biết khóa bí mật dài hạn của B. Như vậy, kẻ tấn công cần khả năng giả mạo chữ ký của B trên thông điệp chứa giá trị *nonce* mới của A (do kẻ tấn công mạo danh B để trao đổi với A trong phiên này, nên B chưa bao giờ ký lên một thông điệp như vậy). Tuy nhiên, theo giả thiết về độ an toàn của lược đồ chữ ký, điều này dường như là bất khả thi đối với kẻ tấn công. Do đó, ISO-STS-MAC có khả năng kháng tấn công KCI. Lưu ý rằng, kết quả tương tự cũng đúng cho STS-MAC và STS-ENC.

Tính chất an toàn về phía trước. Do khóa phiên trong giao thức ISO-STS-MAC chỉ phụ thuộc vào các giá trị ngắn hạn, nên các khóa phiên không phụ thuộc vào giá trị của các bí mật dài hạn của hai bên tham gia A và B. Vì vậy, nếu kẻ tấn công E không trực tiếp sinh ra các giá trị bí mật ngắn hạn thì việc E tính được khóa phiên là không thể (do giả thiết độ khó của bài toán DDH) ngay cả khi E có biết tất cả các giá trị bí mật dài hạn của bên tham gia đó. Tuy nhiên, trong trường hợp kẻ tấn công sinh ra các giá trị bí mật ngắn hạn như vậy (nghĩa là, A và B không tạo ra giá trị này và như vậy cũng chưa bao giờ ký lên thông điệp chứa giá trị công khai tương ứng với nó) thì E cũng không thể tạo ra chữ ký của bên tham gia A hoặc B trên giá trị công khai ngắn hạn mà E sinh ra (trong luồng thông điệp sửa đổi) vì kẻ tấn công chỉ thực sự biết được khóa bí mật dài hạn của A hoặc B (hoặc cả A và B) sau khi phiên hoàn tất. Do đó, hiểu biết về khóa bí mật dài hạn không giúp gì cho kẻ tấn công trên giao thức ISO-STS-MAC tìm hiểu về các khóa phiên trước đó. Điều này có nghĩa là ISO-STS-MAC có khả năng cung cấp tính chất an toàn về phía trước. Hơn nữa, kết quả tương tự cũng đúng cho STS-MAC và STS-ENC.

V. KẾT LUẬN

Trong bài báo này, chúng tôi đã nghiên cứu và phân tích một cách rõ ràng các thuộc tính an toàn (bao gồm tính xác thực khóa ẩn, tính chứng nhận khóa hiện, khả năng kháng tấn công UKS, khả năng kháng tấn công KCI và tính an toàn về phía trước) của ba giao thức STS-MAC, STS-ENC và ISO-STS-MAC. Theo đó, kết quả nhận được có thể được tóm tắt qua Bảng 2.

BẢNG 2. CÁC THUỘC TÍNH AN TOÀN VỚI STS

Các tính chất	**Giao thức STS-ENC**	**Giao thức STS-MAC**	**Giao thức ISO-STS-MAC**
Số bước	3	3	3
Xác thực khóa ẩn	Cho cả *A* lẫn *B*	Cho cả *A* lẫn *B*	Cho cả *A* lẫn *B*
Chứng thực khóa hiện	Cho cả *A* lẫn *B*	Cho cả *A* lẫn *B*	Cho cả *A* lẫn *B*
Kháng tấn công UKS thay đổi khóa công khai	không	không	Có
Kháng tấn công UKS trực tuyến dựa trên tính chất DSKS của lược đồ chữ ký	Có	không	Có
Tính chất an toàn về phía trước	Có ngay cả khi khóa bí mật dài hạn của *A* và *B* bị lộ	Có ngay cả khi khóa bí mật dài hạn của *A* và *B* bị lộ	Có ngay cả khi khóa bí mật dài hạn của *A* và *B* bị lộ
Kháng tấn công KCI	Có	Có	Có

Bên cạnh đó, chúng tôi đã đề xuất 1 phương án sửa đổi (Hình 7) cho STS-MAC mà có đầy đủ các tính chất của ISO-STS-MAC được phân tích trong bài báo này và việc cài đặt của nó có thể đơn giản hơn so với ISO-STS-MAC.

Như đã trình bày ở Mục I, kết quả của bài báo này là tạo cơ sở quan trọng cho việc tìm hiểu hai giao thức Echinacea - 2 và Echinacea - 3 trong *Dự thảo chuẩn về giao thức trao đổi khóa* của Liên Bang Nga. Do đó, trong những nghiên cứu tiếp theo, chúng tôi sẽ tập chung nghiên cứu về hai giao thức này.

TÀI LIỆU THAM KHẢO

[1] S. Blacke-Wilson, A. Menezes. "Unknown key-share attacks on the station-to-station (STS) protocol" In: International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, pp. 154-170, 1999.

[2] W. Diffie, P. van Oorschot, and M. Wiener. "Authentication and authenticated key exchanges. Designs", Codes and cryptography, pp.107-125, 1992.

[3] J. Baek, K. Kim. "Remarks on the unknown key share attacks". IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, pp. 2766-2769, 2000.

[4] A. Menezes, P. van Oorshot, and S. Vanstone. "Handbook of Applied Cryptography". CRC Press, New York, 1997.

[5] M. Bellare, R. Canetti and H. Krawczyk. "A modular approach to the design and analysis of authentication and key exchange protocols". Proceedings of the 30th Annual Symposium on the Theory of Computing, 1998.

[6] W. Diffie and M. E. Hellman. "New directions in cryptography". IEEE Transactions on Information Theory, vol. 22, pp. 644-654, 1976.

[7] H. Krawczyk, "SIGMA: The 'SiGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols". Crypto '03, LNCS No. 2729, pp. 400–425, 2003.

[8] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. "An efficient protocol for authenticated key agreement". Designs, Codes and Cryptography, pp. 119-134, 2003.

[9] ISO/IEC 11770-3, "Information Technology-Security Techniques-Key Management, Part 3: Mechanisms Using Asymmetric Techniques", 2008.

[10] ISO/IEC 9798-3, Information Technology-Security Techniques-Entity Authentication Mechanisms-Part 3: Entity Authentication Using a Public-Key Algorithm", 1993.

SƠ LƯỢC VỀ TÁC GIẢ

CN. Triệu Quang Phong

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

Email: phongtrieu53@gmail.com

Quá trình đào tạo: Nhận bằng cử nhân chuyên ngành Toán học tài năng, 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.