p-ISSN 3030 - 4695
e-ISSN 3030 - 4709

MỘT LƯỢC ĐỒ CHỮ KÝ MÙ NGƯỠNG VỚI BẢO MẬT CHUYỂN TIẾP

Lê Đình Hải [1], Trịnh Viết Cường[2], Lê Quốc Huy[3]

TÓM TẮT

Bài báo đề xuất chữ ký mù ngưỡng với bảo mật chuyển tiếp (Forward secure threshold blind signature - FSTBS), tích hợp bảo mật chuyển tiếp vào chữ ký mù ngưỡng. Bảo mật chuyển tiếp đảm bảo tất cả các chữ ký tạo ra trước kia vẫn an toàn dù khóa hiện tại bị lộ. Cơ chế này đạt được thông qua việc cập nhật khóa bí mật theo từng giai đoạn thời gian. Trong bài báo này, chúng tôi đề xuất xây dựng một lược đồ FSTBS dựa trên hệ Snowblind với mục tiêu bổ sung thêm tính bảo mật chuyển tiếp trong khi vẫn bảo tồn các ưu điểm về hiệu suất ký, kích thước chữ ký và tính bảo mật. Tuy nhiên, lược đồ FSTBS được đề xuất làm tăng chi phí tính toán ở các bước sinh khóa và cập nhật khóa.

Từ khóa: *Chữ ký mù, chữ ký mù ngưỡng, Snowblind, bảo mật chuyển tiếp.*

DOI: *https://doi.org/10.70117/hdujs.84.2.2026.1061*

1. ĐẶT VẤN ĐỀ

Chữ ký số là nền tảng của an ninh thông tin hiện đại, bảo đảm tính toàn vẹn, xác thực và không chối bỏ trong giao dịch điện tử. *Chữ ký mù* do Chaum đề xuất năm 1982 [1] cho phép người dùng nhận chữ ký trên một thông điệp mà người ký không biết nội dung thông điệp. Nhờ *tính mù*, người ký không thể liên kết chữ ký với thông điệp hoặc người dùng, nhờ đó đảm bảo quyền riêng tư. Nhưng một thách thức lớn đối với chữ ký số nói chung và chữ ký mù nói riêng là *nguy cơ khóa bí mật bị lộ*. Khi đó toàn bộ hệ thống ký sẽ sụp đổ. Để giảm rủi ro lộ khóa, *chữ ký ngưỡng* phân tán khóa bí mật cho nhiều chủ thể: khóa được chia thành n phần và cần ít nhất t phần ($t \leq n$) để hoàn thành việc ký thông điệp. Hệ chữ ký ngưỡng giúp hệ thống chịu lỗi và chống gian lận tốt hơn, vì kẻ tấn công phải xâm phạm đồng thời t trong n thành viên mới giả mạo được chữ ký - điều này được gọi là *tính chịu ngưỡng* [2]. *Chữ ký mù ngưỡng* kết hợp hai tính chất trên (tính mù và tính chịu ngưỡng): nhiều bên cùng tham gia ký mà không biết nội dung và cần một số lượng người ký nhất định để hoàn thành việc ký. Nhưng một thách thức lớn đối với các hệ chữ ký vừa đề cập ở trên là duy trì tính bảo mật ngay cả khi khóa bí mật đã bị lộ. Chẳng hạn, trong hệ chữ ký mù ngưỡng ở trên, khóa bí mật *sk* được phân tán bí mật đến từng người ký để giảm rủi ro lộ khóa, nhưng khóa *sk* này vẫn được giữ nguyên trong suốt vòng đời ký của hệ thống. Do đó, nếu kẻ tấn công kiểm soát đủ số lượng người ký tại một thời điểm bất kỳ, chúng có thể khôi phục khóa bí mật và giả mạo chữ ký cho mọi thông điệp, kể cả các thông điệp trong quá khứ. Một giải pháp cho vấn đề trên là *tính bảo mật chuyển tiếp (cho chữ ký số)*, yêu cầu rằng: thời gian

[1]*Lớp K10 Khoa học Máy tính, Khoa CNTT&TT, Trường Đại học Hồng Đức; Email: ledinhhai789it@gmail.com*
[2] *Phòng QLKHCN&HTQT, Trường Đại học Hồng Đức*
[3] *Trường Đại học Tân Tạo, tỉnh Tây Ninh*

p-ISSN 3030 - 4628
e-ISSN 3030 - 4636

được chia thành các khoảng rời rạc, khóa bí mật được cập nhật định kỳ sao cho việc lộ khóa ở hiện tại không cho phép giả mạo chữ ký ở quá khứ [3]. Nói cách khác, với *bảo mật chuyển tiếp* chữ ký tạo ra trước thời điểm lộ khóa vẫn được đảm bảo bảo mật. Cơ chế này đặc biệt quan trọng trong các ứng dụng như bầu cử điện tử, tiền điện tử hay thông điệp mật [4].

Đóng góp của chúng tôi trong bài báo này là đề xuất *hệ chữ ký mù ngưỡng mới hỗ trợ bảo mật chuyển tiếp*. Cụ thể, từ cú pháp của chữ ký mù ngưỡng, chúng tôi bổ sung thêm một thuật toán giúp cập nhật khoá bí mật theo từng khoảng thời gian. Bằng cách đó, ngay cả khi khóa hiện tại bị lộ, kẻ tấn công cũng không thể tạo ra chữ ký hợp lệ cho các thông điệp đã ký ở các khoảng thời gian trước, do các khóa cũ đã bị thay thế và xóa bỏ. Để minh hoạ cho tính khả thi của hệ chữ ký mù ngưỡng hỗ trợ bảo mật chuyển tiếp, chúng tôi đã điều chỉnh lược đồ chữ ký mù ngưỡng Snowblind [5] một cách phù hợp để bổ sung thêm thuật toán cập nhật khóa và đạt được một lược đồ mới gọi là fs-Snowblind. Đáng chú ý là Snowblind [5] là một chữ ký mù ngưỡng xây dựng trên nhóm không ghép cặp. Quá trình ký diễn ra trong ba vòng trao đổi giữa bên yêu cầu ký và những người ký, tạo ra chữ ký cuối cùng gồm một phần tử nhóm và hai phần tử vô hướng. Kích thước của chữ ký Snowblind chỉ tăng thêm một phần tử so với chữ ký Schnorr thông thường, tức gồm ba thành phần thay vì hai, làm cho Snowblind trở thành một giải pháp có tính hiệu quả và thực tiễn. Cần lưu ý rằng sơ đồ Snowblind được chứng minh an toàn về tính không thể giả mạo và tính mù trong mô hình tiên tri ngẫu nhiên. Chúng tôi lập luận rằng fs-Snowblind cũng đảm bảo các tính chất bảo mật của Snowblind đi kèm với bảo mật chuyển tiếp.

2. CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Phương pháp nghiên cứu

Phương pháp nghiên cứu được triển khai theo hệ thống các công trình học thuật, sách chuyên khảo và tài liệu chuyên ngành về mật mã, chữ ký số, chữ ký mù và chữ ký mù ngưỡng. Trên cơ sở tổng hợp và phân tích các khoảng trống trong nghiên cứu hiện có, chúng tôi thiết kế một lược đồ mới nhằm khắc phục những hạn chế còn tồn tại. Cách tiếp cận tận dụng nền tảng lý thuyết của các hệ mật phổ biến để hình thành giao thức và lược đồ chữ ký mù phù hợp. Cuối cùng, hiệu năng được lượng hóa bằng cách so sánh chi phí tính toán của các thuật toán đề xuất với các lược đồ đã công bố trước đây, sử dụng cùng giả định và tham số đánh giá.

2.2. Chữ ký mù ngưỡng

Chữ ký mù ngưỡng (Threshold blind signature - TBS) là một giao thức ký tương tác giữa người dùng và nhiều nhà phát hành, trong đó mỗi nhà phát hành giữ một phần của khóa bí mật và cùng phối hợp để tạo chữ ký trên một thông điệp đã được làm mù. Cơ chế này đảm bảo rằng không một cá nhân nào có thể tự ký hoặc tiết lộ khóa, đồng thời người ký không biết nội dung thật của thông điệp. Nhờ đó, TBS vừa phân tán quyền kiểm soát, vừa bảo vệ quyền riêng tư của người dùng.

2.3. Lược đồ chữ ký mù ngưỡng Snowblind

Snowblind [5] là một lược đồ chữ ký mù ngưỡng trên nhóm không cặp ghép. Giao thức ký đạt được hiệu quả cao, chỉ cần ba vòng tương tác và tạo ra chữ ký rất gọn gồm một phần tử nhóm và hai phần tử vô hướng trong nhóm. Một điểm mạnh khác của công trình là

p-ISSN 3030 - 4695
e-ISSN 3030 - 4709

các lược đồ chữ ký mù nền tảng vốn đã có giá trị độc lập. So với công trình gần nhất của Tessaro và Zhu [9], các lược đồ này rút ngắn kích thước chữ ký (ba phần tử thay vì bốn) và đơn giản hóa chứng minh an toàn.

2.4. Chữ ký có bảo mật chuyển tiếp

Chữ ký có bảo mật chuyển tiếp [3,7] là cơ chế ký số trong đó cho phép khóa bí mật thay đổi theo thời gian nhằm hạn chế hậu quả của việc lộ khóa. Thay vì sử dụng một khóa bí mật cố định suốt đời, toàn bộ thời gian hoạt động của hệ thống được chia thành các thời kỳ rời rạc. Mỗi khoảng thời gian e có một khóa bí mật riêng $sk^{(e)}$ dùng để ký các thông điệp trong khoảng thời gian đó. Khi chuyển sang khoảng thời gian kế tiếp $(e + 1)$, khóa bí mật được cập nhật từ $sk^{(e)}$ thành $sk^{(e+1)}$ theo một hàm một chiều F nào đó, và quan trọng là khoá $sk^{(e)}$ được xóa khỏi hệ thống. Hàm một chiều F đảm bảo rằng từ khóa mới $sk^{(e+1)}$ không thể tính ngược ra khóa cũ $sk^{(e)}$. Nhờ đó, nếu kẻ tấn công đột nhập và lấy cắp được khóa bí mật hiện tại $sk^{(e+1)}$, chúng không thể làm giả chữ ký của các khoảng thời gian $< e + 1$ trước đó, bởi những khóa ứng với các khoảng thời gian trước đó đã bị xóa và về mặt tính toán không thể khôi phục từ $sk^{(e+1)}$.

2.5. Cơ chế chia sẻ bí mật Shamir

Cơ chế chia sẻ bí mật Shamir (Shamir's secret sharing - SSS) là một trong những cơ chế chia sẻ bí mật nổi tiếng nhất, được Shamir [8] đề xuất năm 1979. Thuật toán này dựa trên ý tưởng rằng một bí mật có thể được chia thành nhiều phần và chỉ khi tập hợp đủ một số lượng phần tối thiểu t trong số n phần đã chia thì mới có thể khôi phục lại bí mật gốc. Cốt lõi của SSS dựa trên nội suy đa thức Lagrange: Người phân phối chọn một đa thức bậc $t - 1$: $f(x) = s + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}(mod\ p)$ trong đó s là bí mật cần chia sẻ; $a_{t-1}, \dots, a_{t-1}$ là các hệ số được chọn ngẫu nhiên; p là một số nguyên tố lớn. Mỗi người tham gia i sẽ nhận được một cặp $(i, f(i))$. Khi cần khôi phục bí mật, chỉ cần tối thiểu t người hợp tác để dựng lại đa thức $f(x) = \sum_{i=0}^{n} y_i \prod_{j=0\ j\neq i}^{n} \frac{x-x_j}{x_i-x_j}$ thông qua nội suy Lagrange.

2.6. Cơ chế chia sẻ bí mật xác thực được của Feldman

Cơ chế chia sẻ bí mật xác thực được của Feldman (Feldman verifiable secret sharing - Feldman VSS) [10] là phiên bản mở rộng của sơ đồ chia sẻ bí mật Shamir, đảm bảo mọi người tham gia trung thực đều nhận được phần chia sẻ hợp lệ ngay cả khi có bên bị lỗi hoặc gian lận. Cơ chế này sử dụng cam kết công khai đa thức ở dạng số mũ, bằng cách truyền phát các giá trị $B_k = g^{ak}$. Mỗi người nhận có thể tự kiểm tra tính hợp lệ của phần chia sẻ s_i thông qua đẳng thức $g^{s_i} = \prod_{k=0}^{t-1} B_k^{i^k}$.

2.7. Chữ ký mù ngưỡng và bảo mật chuyển tiếp

2.7.1. Cú pháp tổng quát

Một hệ chữ ký mù ngưỡng bảo mật chuyển tiếp (Forward secure threshold blind signature - FSTBS) gồm một thuật toán sinh ra bộ tham số công khai par do một bên đáng tin cậy sinh ra và cung cấp cho toàn bộ hệ thống. Quá trình sinh khóa gồm: (1) tạo khóa công khai, bí mật cho *n* người ký; (2) tạo khóa công khai chung cho toàn bộ hệ thống. Trong quá

p-ISSN 3030 - 4628
e-ISSN 3030 - 4636

trình ký, một nhóm S người tham gia ký, sử dụng khóa bí mật riêng sk_i mà không biết nội dung thông điệp. Người dùng kết hợp các phản hồi từ tập S để tạo chữ ký σ, đảm bảo $S \subseteq [n]$ và $t \leq |S| \leq n$. Chữ ký hợp lệ khi $FSTBS.Ver(pk, m, \sigma) = 1$.

Định nghĩa: Hệ FSTBS bao gồm các thuật toán: $FSTBS = (FSTBS.Setup, FSTBS.Gen, FSTBS.Update, FSTBS.ISign, FSTBS.USign, FSTBS.Ver)$:

- $FSTBS.Setup(1^\kappa) \rightarrow par$: Thuật toán này nhận đầu vào là một tham số bảo mật k, sinh ra bộ tham số công khai par.

- $FSTBS.Gen(n, t, par) \rightarrow (pk^{(0)}, \{(pk_i{}^{(0)}, sk_i{}^{(0)})\}^n{}_{i=1}, aux)$: Thuật toán này nhận đầu vào là số lượng người ký n, ngưỡng t, và bộ tham số công khai par. Đầu ra là khóa công khai chung $pk^{(0)}$ ứng với khóa bí mật chung $sk^{(0)}$ ($sk^{(0)}$ không được xuất ra một cách tường minh), các cặp $(pk_i{}^{(0)}, sk_i{}^{(0)})$ gồm khóa công khai cá nhân $pk_i{}^{(0)}$ và phần chia sẻ khoá bí mật $\{sk_i{}^{(0)}\}$ cho người ký $i \in \{1, \cdots, n\}$ và thông tin bổ sung aux. Thời điểm xem xét là thời điểm bắt đầu của hệ thống tức là khoảng thời gian 0.

- $FSTBS.Update(par, \{sk_i^{(e-1)}\}_{i=1}^n, e) \rightarrow (pk^{(e)}, \{(pk_i{}^{(e)}, sk_i{}^{(e)})\}^n{}_{i=1})$: Đầu vào của thuật toán là tham số hệ thống *par,* phần chia sẻ khoá bí mật hiện tại $\{sk_i^{(e-1)}\}_{i=1}^n$cho khoảng thời gian $e-1$, bộ đếm khoảng thời gian *e*. Đầu ra của thuật toán là khóa công khai mới $pk^{(e)}$ ứng với khóa bí mật chung mới $sk^{(e)}$ ($sk^{(e)}$ không được xuất ra một cách tường minh, mà được tổng hợp ngầm ẩn dựa vào ngưỡng *t*), các cặp $(pk_i{}^{(e)}, sk_i{}^{(e)})$ gồm khóa công khai cá nhân $pk_i{}^{(e)}$ và phần chia sẻ khoá bí mật $\{sk_i{}^{(e)}\}$ cho người ký $i \in \{1, \cdots, n\}$ tại khoảng thời gian e.

- $FSTBS.ISign(i, sk_i^{(e)}, aux)$ và $FSTBS.USign(pk^{(e)}, aux, m, S^{(e)})$ điều phối quá trình ký gồm các vòng tương tác giữa người ký $i \in \{1, \cdots, n\}$ với người dùng tại khoảng thời gian e. Mỗi người ký $i \in \{1, \cdots, n\}$, sẽ gọi thuật toán $FSTBS.ISign$ với đầu vào là chỉ số i, phần chia sẻ khoá bí mật $sk_i^{(e)}$ của người ký đó tại khoảng thời gian e và thông tin bổ sung aux. Người dùng sẽ gọi thuật toán $FSTBS.USign$ với đầu vào là khoá công khai $pk^{(e)}$, thông tin bổ sung aux, thông điệp m và nhóm người ký $S^{(e)}$. Đầu ra của quá trình tương tác này là chữ ký $Sig^{(e)}$ cho thông điệp m. Thông điệp *m* sẽ được *mù hoá* trong khi ký.

- $FSTBS.Ver(pk^{(e)}, m, Sig^{(e)}) \rightarrow 1/0$: Thuật toán nhận thông tin đầu vào là khóa công khai $pk^{(e)}$, chữ ký $Sig^{(e)}$, thông điệp m. Đầu ra của thuật toán cho biết chữ ký là hợp lệ (xuất ra 1) hay không hợp lệ (xuất ra 0).

2.7.2. Yêu cầu bảo mật

- *Tính mù chuyển tiếp:* Tính mù chuyển tiếp đảm bảo rằng: ngay cả khi toàn bộ khóa bí mật tại khoảng thời gian e bị lộ, đối thủ cũng không thể liên kết được chữ ký với thông điệp đã được ký tại các khoảng thời gian trước đó.

- *Tính ngưỡng chuyển tiếp:* Trong hệ FSTBS, tính ngưỡng cần được duy trì trong suốt quá trình cập nhật khóa. Nghĩa là: tại mọi khoảng thời gian e, phải có từ t trong n phần khóa $\{sk_i^{(e)}\}_{i=1}^n$ thì mới có thể kết hợp thành khóa bí mật $sk^{(e)}$ và việc cập nhật khóa qua các khoảng thời gian e không phá vỡ tính chất này.

p-ISSN 3030 - 4695
e-ISSN 3030 - 4709

- *Tính không thể giả mạo thêm một chuyển tiếp:* Tính không thể giả mạo thêm một yêu cầu rằng: ngay cả khi kẻ giả mạo có thể thực hiện nhiều truy vấn ký ở một khoảng thời gian e' trước khoảng thời gian e, sau đó kẻ giả mạo biết khóa bí mật ở khoảng thời gian e, thì kẻ giả mạo vẫn không thể tạo ra nhiều chữ ký hơn số lượng truy vấn ký đã thực hiện ở khoảng thời gian e'.

3. KẾT QUẢ NGHIÊN CỨU VÀ THẢO LUẬN

Trong phần này, chúng tôi đề xuất một lược đồ FSTBS cụ thể được xây dựng dựa trên lược đồ Snowblind [5]. Lược đồ FSTBS này được gọi là fs-Snowblind.

3.1. Lược đồ fs-Snowblind đề xuất

Lược đồ fs-Snowblind này bao gồm các thuật toán ($FSTBS.Setup$, $FSTBS.Gen$, $FSTBS.Update$, $FSTBS.ISign$, $FSTBS.USign$, $FSTBS.Ver$). Chú ý rằng, $FSTBS.ISign$, $FSTBS.USign$, $FSTBS.Ver$ được thực hiện tại khoảng thời gian e cụ thể. Thuật toán $FSTBS.Gen$ của fs-Snowblind có sự thay đổi so với thuật toán $Snowblind.KeyGen$ của Snowblind ở những điểm sau:

- Trong $Snowblind.KeyGen$, khoá bí mật chung sk được sinh ra trước. Sau đó, cơ chế chia sẻ khoá bí mật Shamir mới được sử dụng để phân tán khoá sk này thành các phần sk_i cho mỗi người ký i. Chúng ta không quan tâm tới khoảng thời gian.

- Trong khi đó, thuật toán $FSTBS.Gen$ theo cách thức ngược lại: Cho phép mỗi người ký i tự chọn cho mình phần khoá bí mật sk_i mà nếu đủ i phần thì có thể tổng hợp được thành khoá bí mật chung sk. Ở đây cơ chế Feldman VSS được sử dụng. Mặc định thuật toán $FSTBS.Gen$ chỉ được chạy tại khoảng thời gian 0. Thuật toán $FSTBS.Update$ được bổ sung để có thể triển khai được tính bảo mật chuyển tiếp. Thuật toán này giúp cập nhật các khóa riêng $\{sk_i^{(e-1)}\}_{i=1}^n$ và $sk^{(e-1)}$ thành $\{sk_i^{(e)}\}_{i=1}^n$ và $sk^{(e)}$. Do tính phân tán và thiết lập ngưỡng của $\{sk_i^{(e-1)}\}_{i=1}^n$, cơ chế tái phân phối bí mật được sử dụng để đảm bảo mọi thành viên trong n thành viên ký ban đầu nhận được bản cập nhật các khoá riêng này.

Các thuật toán của fs-Snowblind được mô tả cụ thể như sau:

a) Thuật toán khởi tạo hệ thống FSTBS.Setup(1^κ)

Đầu vào: 1^κ với k là tham số an toàn.

Các bước thực hiện:

Bước 1. Chọn một nhóm G có cấp p và phần tử sinh g với p là số nguyên tố κ-bit.

Bước 2. Chọn $h \leftarrow \$\ G$

Bước 3. Chọn hai hàm băm $H_{cm}, H_{sig} : \{0,1^*\} \rightarrow Z_p^*$ **,** xét một lược đồ chữ ký DS = $\{DS.Setup, DS.Sign, DS.Verify\}$ được dùng để xác thực trong quá trình ký tương tác (Hình 1). Chạy $par_{sig} \leftarrow DS.Setup(1^\kappa)$.

Đầu ra: Toàn bộ tham số hệ thống công khai par

b) Thuật toán sinh khóa FSTBS.Gen(n, t, par)

Đầu vào: n là số người ký, t là ngưỡng ký, và par là tham số hệ thống gồm (G, p, g).

p-ISSN 3030 - 4628
e-ISSN 3030 - 4636

Các bước thực hiện:

Bước 1. Mỗi người ký i chọn đa thức ngẫu nhiên $f_i^{(0)}(x)$ bậc t – 1 có các hệ số $a_{i,k}^{(0)}$ được chọn ngẫu nhiên từ $Z_p{}^*$, sao cho $s_i^{(0)} := f_i^{(0)}(0) = a_{i,0}^{(0)}$.

Bước 2. Người ký i truyền phát các cam kết $C_i^{(0)} = \{C_{i,k}^{(0)}\}_{k=0}^{t-1} = \{g^{a_{i,k}^{(0)}}\}_{k=0}^{t-1}$ cho các hệ số $a^{(0)}{}_{i,k}$ của $f^{(0)}{}_i(x)$, và gửi các phần chia sẻ $f_i^{(0)}(j)$ mã hóa cho mọi người ký $j \neq i$.

Bước 3. Người ký j xác thực các phần chia sẻ bằng Feldman VSS: Kiểm tra $g^{f^{(0)}{}_i(j)} = \prod_{k=0}^{t-1}(C_{i,k}^{(0)})^{j^k} \; mod \; p$.

Đầu ra: Thiết lập khóa công khai chung là $pk^{(0)} := \prod_{i=1}^{n} C_{i,0}^{(0)} = \prod_{i=1}^{n} g^{s_i^{(0)}} = g^{\sum_{i=1}^{n} s_i^{(0)}}$. Do đó, khóa bí mật chung có thể xem là $sk^{(0)} := \sum_{i=1}^{n} s_i^{(0)}$ mod p (tuy nhiên, chú ý rằng không ai trực tiếp tính được tổng $\sum_{i=1}^{n} s_i^{(0)}$ mod p này). Mỗi người ký i tính phần chia sẻ của mình $sk_i^{(0)} = \sum_{j=1}^{n} f_j^{(0)}(i)$ mod p.

c) Thuật toán cập nhật khóa FSTBS.Update$\left(par, \{sk_i^{(e-1)}\}_{i=1}^{n}, e\right), S^{(e)})$

Đầu vào: Tham số hệ thống par và tập phần chia sẻ cũ $\{sk_i^{(e-1)}\}_{i=1}^{n}$, khoảng thời gian e, và nhóm người ký $S^{(e)}$ trong khoảng thời gian e.

Các bước thực hiện:

Bước 1. Mỗi người ký trong $S^{(e)}$ chọn đa thức mới và truyền phát cam kết, tương tự như thuật toán sinh khóa $FSTBS.Gen$ ở trên.

- Mỗi người ký $i \in S^{(e)}$ chọn đa thức ngẫu nhiên $f_i{}^{(e)}(x)$ bậc t – 1 có các hệ số $a_{i,k}^{(e)}$ (k = 0,…, t–1) được chọn ngẫu nhiên từ $Z_p{}^*$ sao cho $s_i{}^{(e)} = f_i{}^{(e)}(0) = a_{i,0}^{(e)}$.

- Mỗi người ký $i \in S^{(e)}$, gửi cam kết $C_i{}^{(e)} = \{C_{i,k}{}^{(e)}\}_{k=0}^{t-1} = \{g^{a_{i,k}^{(e)}}\}_{k=0}^{t-1}$ cho các hệ số $a_{i,k}^{(e)}$ của $f_i{}^{(e)}(x)$, và gửi các phần $f_j{}^{(e)}(0)$ cho người ký $j \neq i$.

- Người ký $j \in S^{(e)}$ xác thực phần chia sẻ bằng lược đồ Feldman VSS: Kiểm tra $g^{f_i{}^{(e)}(j)} = \prod_{k=0}^{t-1}(C_{i,0}{}^{(e)})^{j^k} \; mod \; p$.

- Thiết lập khóa công khai chung tại khoảng thời gian e là $pk^{(e)} := \prod_{i \in S} C_{i,0}{}^{(e)} = g^{\sum_{i \in S} s_i^{(e)}}$. Do đó, khóa bí mật chung có thể xem là $sk^{(e)} = \sum_{i \in S} s_i^{(e)}$ mod p (tuy nhiên, chú ý rằng không ai trực tiếp tính được tổng $\sum_{i \in S} s_i^{(e)}$ mod p này). Hơn nữa, ta cần phải phân phối lại khoá $sk^{(e)}$ để đảm bảo n người ký (gồm cả những người ngoài nhóm $S^{(e)}$) có phần chia sẻ của $sk^{(e)}$ để đảm bảo tính ngưỡng.

Mỗi người ký $i \in S^{(e)}$ tính phần được chia sẻ của mình $sk_i^{(e)} = \sum_{j \in S} f_j^{(e)}(i) mod \; p$.

Bước 2. Tái phân phối: Mục tiêu là tạo một đa thức f(x) bậc (t – 1) sao cho $f(0) = sk^{(e)}$.

- Mỗi người ký $i \in S^{(e)}$ tạo một đa thức phụ $h_i{}^{(e)}(x)$ với $h_i{}^{(e)}(0) = sk_i^{(e)}$.

- Tổng hợp các đa thức $h_i{}^{(e)}(x)$ theo cách phân tán, sử dụng hệ số Lagrange λ_i: $f(x) = \sum_{i \in T} \lambda_i \cdot h_i{}^{(e)}(x)$, với $T \subset S$ là một tập con t người ký bất kỳ trong nhóm S, và λ_i là hệ số Lagrange tương ứng với tập T tại $x = 0$, được định nghĩa là $\lambda_i = \prod_{j \in T, j \neq i} \frac{-j}{i-j}$ mod p. Tại x

p-ISSN 3030 - 4695
e-ISSN 3030 - 4709

= 0, giá trị của f(x) là $f(0) = \sum_{i\in T} \lambda_i \cdot h_i{}^{(e)}(0) = \sum_{i\in T} \lambda_i \cdot sk_i^{(e)} = sk^{(e)}$ (vì $sk_i^{(e)}$ là các phần chia sẻ của $sk^{(e)}$). Điều này đảm bảo rằng đa thức f(x) có hệ số tự do đúng bằng $sk^{(e)}$, mà không yêu cầu bất kỳ người ký nào biết $sk^{(e)}$ trực tiếp.

- Mỗi người ký $i \in S$ tính phần chia sẻ con $\delta_{i,j} = f(j)$ mod p cho mọi $j \in \{1,2,\dots,n\}\backslash\{i\}$, gửi $\delta_{i,j}$ kèm bằng chứng Feldman $g^{\delta_{i,j}}$.

Bước 3. Tái tạo cục bộ và xóa: Mỗi người ký $j \in \{1,2,\dots,n\}$ nhận $\delta_{i,j}$ và bằng chứng Feldman $g^{\delta_{i,j}}$, xác thực bằng chứng, tính phần chia sẻ mới $sk_j^{(e),new} := \sum_{i\in S} \delta_{i,j}$ mod p. Xóa an toàn tất cả dữ liệu liên quan đến $sk^{(e-1)}$ (phần chia sẻ cũ, đa thức cũ).

Bước 4. Xác thực: Sau khi các người ký nhận và tổng hợp các phần chia sẻ mới $sk_j^{(e),new}$ ở Bước 3, họ cần xác thực rằng các phần chia sẻ này thực sự tương ứng với khóa công khai chung mới $pk^{(e)} := \prod_{i\in S} C_{i,0}{}^{(e)} = g^{\sum_{i\in S} s_i^{(e)}} = g^{sk^{(e)}}$ được đề cập ở Bước 1. Cụ thể:

- Mỗi người ký $j \in \{1,2,\dots,n\}$ truyền phát cam kết mới $g^{sk_j^{(e),new}}$.

- Tất cả người ký hoặc một người điều phối kiểm tra $pk^{(e),new} = \prod_{j\in U}(g^{sk_j^{(e),new}})^{\lambda_j}$ với U là tập con t người ký bất kỳ, λ_j là hệ số Lagrange. (Vì $\sum_{j\in U} \lambda_j \cdot sk_j^{(e),new} = sk^{(e)}$). Nếu $pk^{(e),new} = pk^{(e)}$ thì phần chia sẻ mới $sk_j^{(e),new}$ được xác nhận là đúng, tức là chúng thực sự là các phần chia sẻ của $sk^{(e)}$ thông qua đa thức f(x) được tạo ở Bước 4.

Bước 5. Hệ thống (hoặc một người điều phối) kiểm tra xem có ít nhất t người ký xác nhận tính đúng đắn của $pk^{(e)}$ và $sk_j^{(e),new}$ mới hay không. Nếu có, hệ thống coi quá trình cập nhật là hợp lệ. Khi đó, $sk_i^{(e)} \leftarrow sk_i^{(e),new}$ với mọi $i \in \{1,2,\dots,n\}$. Nếu không, thì huỷ bỏ việc cập nhật: các người ký giữ nguyên phần chia sẻ cũ $sk_i^{(e-1)}$ và khóa công khai $pk^{(e-1)}$.

Đầu ra: Khóa công khai mới $pk^{(e)}$, tập khóa bí mật mới $\{sk_j^{(e),new}\}_{j=1}^n$.

d) Thuật toán xác minh FSTBS.Ver$(pk^{(e)}, m, Sig^{(e)})$

Đầu vào: Khóa công khai $pk^{(e)}$, khóa bí mật $Sig^{(e)}$ và thông điệp m

Các bước thực hiện:

Bước 1. Tách $(\bar{R}^{(e)}, \bar{z}^{(e)}, \bar{y}^{(e)}) \leftarrow Sig^{(e)}$.

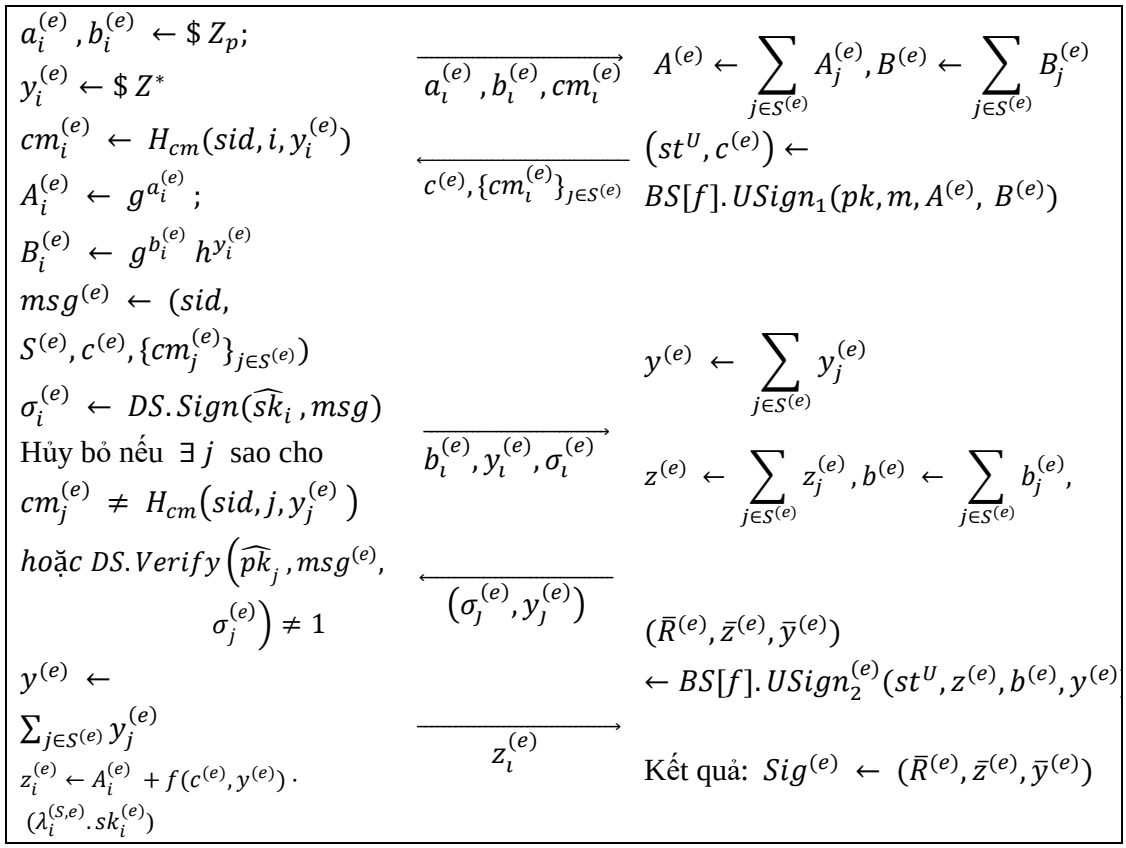
Bước 2. Tính $\bar{c}^{(e)} \leftarrow H_{sig}(pk^{(e)}, m, \bar{R}^{(e)})$. Nếu $\bar{y}^{(e)} = 0$ hoặc $\bar{R}^{(e)} \cdot pk^{f(\bar{c}^{(e)}, \bar{y}^{(e)})} \neq g^{\bar{z}^{(e)}} h^{\bar{y}^{(e)}}$, trả về 0. Ngược lại, trả về 1.

Đầu ra: Xác thực thành công khi FSTBS.Ver$\left(pk^{(e)}, m, Sig^{(e)}\right) = 1$ ngược lại thất bại khi FSTBS.Ver$\left(pk^{(e)}, m, Sig^{(e)}\right) = 0$.

e) Quá trình ký tương tác giữa $FSTBS.ISign(i, sk_i^{(e)}, aux)$ (cho người ký i) và $FSTBS.USign(pk^{(e)}, aux, m, S^{(e)})$ diễn ra như trong Hình 1 dưới đây:

FSTBS.ISign $(i, sk_i^{(e)}, aux)$:		**FSTBS.USign**$(pk^{(e)}, aux, m, S^{(e)})$**:**
Phân tích $\{\widehat{pk_\iota}\}_{i\in[n]} \leftarrow aux$	$\overleftarrow{S^{(e)}}$	Phân tích $\{\widehat{pk_\iota}\}_{i\in[n]} \leftarrow aux$
Hủy bỏ nếu $i \notin S^{(e)}$		

p-ISSN 3030 - 4628
e-ISSN 3030 - 4636



Hình 1. Quá trình ký tương tác của fs-Snowblind (Đề xuất)

(Ở đây $BS[f].USign_1$, $BS[f].USign_2$ như trong Hình 4 của bài báo [5], với $f = f(c, y)$ là một hàm phi tuyến đi từ $Z_p \times Z_p \rightarrow Z_p$. Trong [5], $f(c, y)$ có thể là một trong hai khả năng $f_1(c, y) := c + y^5$ hoặc $f_2(c, y) := cy$. Ta viết $BS[f]$ để thể hiện một lược đồ chữ ký mù được tham số hoá bởi hàm f)

3.2. Thảo luận về tính bảo mật của fs-Snowblind

Tính đúng đắn chuyển tiếp: Với mỗi khoảng thời gian e, biểu thức kiểm tra giữ nguyên dạng (kiểu Schnorr) của Snowblind, chỉ bổ sung chỉ số e, do đó tính đúng đắn được thừa hưởng hoàn toàn từ Snowblind.

Tính mù chuyển tiếp: Với mỗi khoảng thời gian e, quá trình ký của fs-Snowblind trùng với Snowblind (chỉ có thêm chỉ số e độc lập với nội dung), do đó tính mù được giữ nguyên như của Snowblind. Hơn nữa, thuật toán $FSTBS.Update$ đảm bảo tính một chiều, tức là biết $sk^{(e)}$, $sk_i^{(e)}$ không thể suy ngược được $sk^{(e\prime)}$ và $sk_i^{(e\prime)}$ (với $e' < e$), do đó tính mù của các thời điểm trước khoảng thời gian e không bị xâm phạm.

Tính chịu ngưỡng chuyển tiếp: Ở mỗi khoảng thời gian e, quy trình ký trong fs-Snowblind tương tự Snowblind, tức là chỉ khi có ít nhất t người ký mới có thể tái tạo khóa $sk^{(e)}$ hợp lệ. Một lần nữa, thuật toán $FSTBS.Update$ đảm bảo tính một chiều, tức là biết $sk^{(e)}$, $sk_i^{(e)}$ không thể suy ngược được $sk^{(e\prime)}$ và $sk_i^{(e\prime)}$ (với $e' < e$), do đó $FSTBS.Update$ không giảm tính chịu ngưỡng của fs-Snowblind.

p-ISSN 3030 - 4695
e-ISSN 3030 - 4709

Tính không giả mạo thêm một chuyển tiếp: Trong một khoảng thời gian cố định, quy trình ký trong fs-Snowblind tương tự Snowblind nên tính bảo mật giả mạo thêm một không đổi so với Snowblind. Thêm vào đó, thuật toán $FSTBS.Update$ cũng không ảnh hưởng đến tính bảo mật này.

3.3. Thảo luận về hiệu suất của fs-Snowblind

Bảng 1 cho thấy fs-Snowblind giữ nguyên cấu trúc giao thức 3 vòng và chi phí ký gần như không đổi so với Snowblind gốc. Điểm khác biệt chủ yếu nằm ở cơ chế sinh khoá và cập nhật khóa: trong fs-Snowblind, các khóa bí mật được thay đổi định kỳ theo khoảng thời gian, trong khi khóa ở Snowblind gốc là cố định. Chi phí tăng thêm chủ yếu đến từ việc sinh khoá và cập nhật khóa trong mỗi khoảng thời gian.

Bảng 1. So sánh giữa Snowblind [5] và fs-Snowblind

Đặc tính	Snowblind [5]	fs-Snowblind (đề xuất)
Khoá bí mật và khoá công khai	Cố định suốt vòng đời; pk, sk duy nhất	Cập nhật theo từng khoảng thời gian e; $pk^{(e)}, sk_i^{(e)}$ thay đổi
Quy trình ký	3 vòng; dùng sk, sk_i cố định	3 vòng; dùng $sk^{(e)}, sk_i^{(e)}$ của khoảng thời gian e hiện tại
An toàn khi khóa bị lộ	Nếu tồn tại $T \subseteq \{1, \dots, n\}$ với $\lvert T \rvert \geq t$ sao cho $\{sk_i\}_{i \in T}$ bị lộ thì kẻ tấn công có thể giả mạo chữ ký cho mọi thông điệp.	Nếu tồn tại $T \subseteq \{1, \dots, n\}$ với $\lvert T \rvert \geq t$ sao cho $sk_i^{(e)}, sk^{(e)}$ bị lộ chỉ ảnh hưởng từ khoảng thời gian e về sau.

4. KẾT LUẬN

Trong bài báo này, chúng tôi đề xuất khái niệm *chữ ký mù ngưỡng hỗ trợ bảo mật chuyển tiếp*. Chúng tôi cũng xây dựng được một lược đồ cụ thể cho khái niệm này dựa trên lược đồ chữ ký mù ngưỡng Snowblind [5]. Lược đồ do chúng tôi xây dựng vẫn giữ được những ưu điểm của Snowblind như chữ ký ngắn và các tính bảo mật khác, đồng thời có thêm tính bảo mật chuyển tiếp thông qua việc điều chỉnh cơ chế sinh khóa và bổ sung cơ chế cập nhật khóa theo từng khoảng thời gian. Tuy nhiên, hai cơ chế này cũng làm gia tăng chi phí thực thi. Trong tương lai, chúng tôi dự định tìm cách cải tiến cơ chế sinh khoá và cơ chế cập nhật khoá để tối ưu hiệu suất toàn bộ hệ thống fs-Snowblind. Một hướng nghiên cứu khác nữa là thích ứng bảo mật chuyển tiếp cho các biến thể nâng cao khác của chữ ký mù.

TÀI LIỆU THAM KHẢO

[1] Chaum, D. (1983), *Blind Signatures for Untraceable Payments*, In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds) Advances in Cryptology. Springer, Boston, MA.

[2] Galil, Z., Haber, S., Yung, M. (1988), *Cryptographic Computation: Secure Fault-Tolerant Protocols and the Public-Key Model (Extended Abstract),* In: Pomerance, C. (eds) Advances in Cryptology - CRYPTO ’87. CRYPTO 1987. Lecture Notes in Computer Science, vol 293. Springer, Berlin, Heidelberg.

p-ISSN 3030 - 4628
e-ISSN 3030 - 4636

[3] Bellare, M., Miner, S.K. (1999), *A Forward-Secure Digital Signature Scheme,* In: Wiener, M. (eds) Advances in Cryptology - CRYPTO' 99. CRYPTO 1999. Lecture Notes in Computer Science, vol 1666. Springer, Berlin, Heidelberg.

[4] S. Micali, M. Rabin and S. Vadhan (1999), *Verifiable random functions,* 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039), New York, NY, USA, pp.120-130.

[5] Crites, E., Komlo, C., Maller, M., Tessaro, S., & Zhu, C. (2023). *Snowblind: A threshold blind signature in pairing-free groups,* In T. Malkin & C. Peikert (Eds.), Advances in cryptology - CRYPTO (pp. 669–699). Springer.

[6] D. Chaum (1984), *Blind signatures for untraceable payments,* In Crypto '83, Lecture Notes in Computer Science, page 153.

[7] Kurek, R. (2020), *Efficient Forward-Secure Threshold Signatures,* In: Aoki, K., Kanaoka, A. (eds) Advances in Information and Computer Security. IWSEC 2020. Lecture Notes in Computer Science(), vol 12231. Springer, Cham.

[8] Shamir, A. (1979), *How to Share a Secret,* Communications of the ACM, 22, 612-613.

[9] S. Tessaro and C. Zhu (2022), *Short Pairing-Free Blind Signatures with Exponential Security*, In: EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022II. Ed. by O. Dunkelman and S. Dziembowski

[10] Paul Feldman (1987), *A practical scheme for non-interactive verifiable secret sharing*, In 28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987, pp.427-437. IEEE Computer Society.

A THRESHOLD BLIND SIGNATURE SCHEME WITH FORWARD SECURITY

Le Dinh Hai, Trinh Viet Cuong, Le Quoc Huy

ABSTRACT

The paper proposes a forward secure threshold blind signature scheme (FSTBS), integrating forward security into threshold blind signatures. Forward security ensures that all signatures generated in the past remain secure even if the current secret key is compromised. This is achieved by updating the secret key across discrete time periods. In this paper, we propose constructing a forward secure threshold blind signature (FSTBS) scheme based on the Snowblind framework, with the goal of adding forward security while preserving the advantages in signing efficiency, signature size, and security. However, the proposed FSTBS scheme increases the complexity of the key generation and key update steps.

Keywords: *Blind signature, threshold blind signature, snowblind, forward-Secure.*

** Ngày nộp bài:19/11/2025; Ngày gửi phản biện:21/11/2025; Ngày duyệt đăng: 28/02/2026*