Cài đặt thuật toán sinh tham số RSA 2048 bit trên thiết bị PKI Token

Vũ Tá Cường, Nguyễn Thành Trung, Lê Đình Hùng

***Tóm tắt*—Trong hệ mật mã khóa công khai RSA, tính an toàn của hệ mật phụ thuộc chủ yếu vào việc đảm bảo tính an toàn của khóa bí mật, nên nó thường được lưu trong thiết bị lưu khóa bảo mật PKI Token. Bên cạnh đó, để tăng cường tính an toàn cho khóa bí mật, khóa sẽ được sinh ra ngay trong thiết bị và không thể sao chép ra ngoài, tránh tình trạng lộ lọt khóa. Trong bài báo này, nhóm tác giả giới thiệu giải pháp sinh tham số RSA 2048 bit trên thiết bị PKI Token.**

***Abstract*—In the RSA public-key cryptography system, the security of the cryptosystem depends primarily on ensuring the security of the secret key. Therefore, the secret key is usually stored in security token. In addition, to enhance security of the secret key, the key pair are generated inside the device and cannot be copied out, avoiding the disclosure of the secret key. In this paper, we introduce the 2048-bit RSA parameter generation solution on the PKI Token device.**

Từ khóa— PKI Token; RSA 2048; sinh khóa.

Keyword— PKI Token; RSA 2048; generating key pair.

I. GIỚI THIỆU

Cũng như hầu hết các hệ mật mã khác, mô hình, cấu trúc thuật toán của hệ mật mã khóa công khai RSA là công khai. Vì vậy, việc đảm bảo an toàn trong hệ mật phụ thuộc rất nhiều vào việc đảm bảo tính an toàn của khóa bí mật. Để đảm bảo an toàn cho khóa bí mật thì nó thường được lưu trong thiết bị lưu khóa bảo mật PKI (Public Key Infrastructure) Token. Thiết bị PKI Token là thành phần quan trọng trong các giải pháp bảo mật vì khả năng lưu trữ khóa, tham số mật và tính toán mật mã trong môi trường an toàn. Ngoài ra, để đảm bảo tính an toàn tuyệt đối cho khóa bí mật, người ta lựa chọn phương pháp sinh khóa ngay trong thiết bị PKI Token. Khóa bí mật sau khi sinh ra được lưu vào thiết bị và không thể sao chép ra ngoài, tránh tình trạng lộ lọt khóa. Các thiết bị PKI Token thương mại trên thế giới [1]–[3] cũng đã hỗ trợ chức năng sinh khóa ngay trong thiết bị. Tuy nhiên, khó có thể kiểm soát được thuật toán sinh khóa, cũng như các tiêu chuẩn an toàn cho các tham số RSA. Đối với một số môi trường ứng dụng đặc biệt, tiêu chí quan trọng khi đưa ra các sản phẩm bảo mật là làm chủ toàn bộ thiết kế phần cứng, phần mềm để có thể tùy chọn cài đặt các thuật toán mật mã riêng vào thiết bị. Vì vậy, việc nghiên cứu xây dựng module sinh tham số RSA trong thiết bị PKI Token là rất cần thiết, giúp làm chủ và kiểm soát được toàn bộ quy trình sinh tham số RSA, giám sát loại trừ được các nghi ngờ mã độc, cửa hậu (backdoor) ngay cả ở mức thấp (firmware) tại phần cứng thiết bị PKI Token.

Bài báo được nhận ngày 22/5/2020. Bài báo được nhận xét bởi phản biện thứ nhất ngày 23/6/2020 và được chấp nhận đăng ngày 23/6/2020. Bài báo được nhận xét bởi phản biện thứ hai ngày 04/8/2020 và được chấp nhận đăng ngày 15/9/2020.

II. ĐỀ XUẤT GIẢI PHÁP SINH THAM SỐ RSA TRÊN THIẾT BỊ PKI TOKEN

A. Các phương pháp sinh số nguyên tố lớn

a) Phương pháp sinh số nguyên tố xác suất

Khi tìm hiểu về các phương pháp sinh số nguyên tố xác suất cần chú ý hai vấn đề chính: các phép kiểm tra nguyên tố xác suất và các phương pháp lựa chọn số nguyên dương đầu vào. Có một số phép kiểm tra nguyên tố xác suất thường được sử dụng là kiểm tra Fermat, Solovay-Strassen, Miller-Rabin và Lucas [6], [7]. Trong đó, phép kiểm tra Miller-Rabin có xác suất sai thấp nhất [6], [7]. Các phương pháp sinh số nguyên tố xác suất thường được xây dựng dựa trên phép kiểm tra này. Có 2 phương pháp lựa chọn số nguyên dương đầu vào là phương pháp lựa chọn ngẫu nhiên và phương pháp tìm kiếm tăng dần [7], [8].

b) Phương pháp sinh số nguyên tố tất định

Các phương pháp sinh số nguyên tố xác suất chỉ cho phép tạo ra các số nguyên tố với một xác suất nào đó. Phương pháp sinh số nguyên tố tất định cho phép tạo ra các số nguyên tố có thể chứng minh được về mặt toán học. Một số thuật toán sinh số nguyên tố tất định phổ biến được trình bày trong [8], [9] là: phương pháp của Maurer, phương pháp của Shawe-Taylor, phương pháp căn bậc hai và phương pháp căn bậc ba.

Đánh giá sơ bộ các phương pháp

Trong hai phương pháp sinh số nguyên tố trên, về tính an toàn, phương pháp sinh số nguyên tố tất định vượt trội hơn hẳn phương pháp sinh số nguyên tố xác suất, do xác suất sai bằng không. Tuy nhiên, xét về tốc độ thực thi và độ phức tạp tính toán thì phương pháp sinh số nguyên tố xác suất có ưu thế hơn. Phương pháp sinh số nguyên tố xác suất được cho là dễ cài đặt và có tốc độ nhanh hơn so với phương pháp sinh số nguyên tố tất định [9], [11].

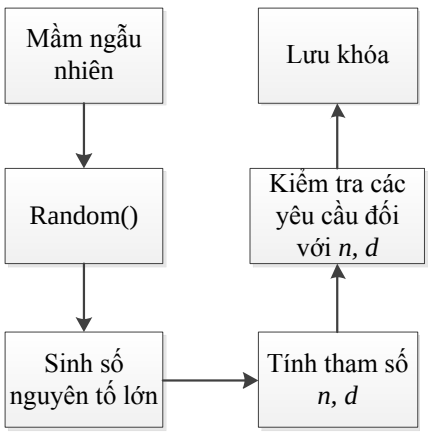
Trong hai phương pháp sinh số nguyên dương đầu vào, phép kiểm tra các ứng cử viên ngẫu nhiên được đánh giá cao hơn về phân bố đầu ra của thuật toán, tuy nhiên trong thực hành kém hiệu quả hơn thuật toán tìm kiếm tăng dần. Các nghiên cứu [11], [12] đã chỉ ra rằng phương pháp tìm kiếm tăng dần là tiết kiệm và có hiệu quả hơn đáng kể so với phương pháp chọn "ngẫu nhiên đều".

B. Đề xuất mô hình sinh tham số RSA trên thiết bị PKI Token

Việc sinh tham số RSA đòi hỏi khối lượng tính toán lớn, do phải tạo ra các số nguyên tố bí mật p và q đủ lớn để đảm bảo các tiêu chuẩn an toàn. Ngoài ra, việc thực hiện tính toán trên các số nguyên tố lớn cũng đòi hỏi tài nguyên tính toán lớn [7], [9], [11]. Để thực hiện sinh tham số RSA trên thiết bị PKI Token, các thuật toán sinh số nguyên tố cần được lựa chọn, chỉnh sửa để đảm bảo tính khả thi.

Như đã phân tích ở trên, trong các phương pháp sinh số nguyên tố đã nêu, phương pháp sinh số nguyên tố xác suất là phương pháp có độ phức tạp thấp, không yêu cầu nhiều tài nguyên, phù hợp với các thiết bị có tài nguyên hạn chế. Vì vậy, nhóm tác giả lựa chọn phương pháp sinh số nguyên tố xác suất để xây dựng module sinh tham số RSA trên thiết bị PKI Token. Ngoài ra, nhóm tác giả cũng đưa ra lựa chọn phương pháp chọn ứng cử viên đầu vào và phép kiểm tra tính nguyên tố sẽ sử dụng. Về phương pháp lựa chọn ứng cử viên đầu vào, nhóm tác giả sử dụng phương pháp tìm kiếm tăng dần, vì cách tiếp cận này được cho là hiệu quả hơn [11], [12]. Về phép kiểm tra tính nguyên tố, nhóm tác giả sẽ sử dụng kiểm tra Miller-Rabin, vì đây là phương pháp có tốc độ nhanh, dễ cài đặt và có độ chính xác cao [6], [7]. Module sinh tham số RSA trên thiết bị PKI Token hoạt động theo mô hình như trong Hình 1.

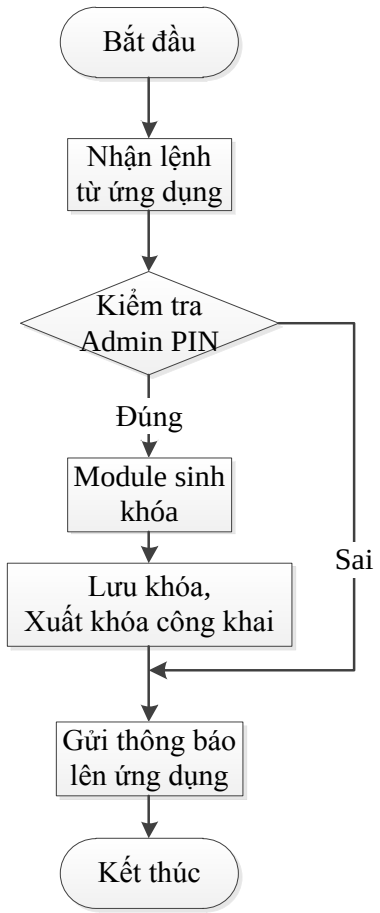
Mô hình sinh tham số RSA trên thiết bị PKI Token được mô tả cụ thể như sau: Mầm ngẫu nhiên được lấy dựa trên lỗi lượng tử hóa của bộ ADC (Analog to Digital Converter) trong con chip STM32F103. Mầm ngẫu nhiên này được sử dụng để tạo ra số ngẫu nhiên, đưa vào hàm sinh số nguyên tố lớn. Các số nguyên tố lớn được sinh ra p và q sẽ được sử dụng để tính ra các tham số (n và d). Các tham số này sẽ được đánh giá thông qua các hàm kiểm tra các tiêu chuẩn tham số RSA trước khi lưu vào thiết bị dưới dạng khóa. Để giảm bớt khối lượng tính toán cho thiết bị, số mũ công khai được gắn cố định $e = 75537$.



Hình 1. Mô hình sinh tham số RSA trên thiết bị PKI Token

III. NGHIÊN CỨU XÂY DỰNG MODULE FIRMWARE SINH THAM SỐ RSA TRÊN THIẾT BỊ PKI TOKEN

Thiết bị PKI Token là thiết bị chuyên dụng do nhóm tác giả tự nghiên cứu và chế tạo, làm chủ toàn bộ từ thiết kế phần cứng, xây dựng firmware cho thiết bị. Thiết bị này được sử dụng để lưu cặp khóa RSA nhằm đảm bảo tính an toàn cho khóa bí mật trong suốt quá trình sử dụng. Việc sinh tham số RSA trực tiếp trên thiết bị sẽ đảm bảo tính an toàn tuyệt đối cho khóa mật sinh ra, vì khóa mật sinh ra sẽ không thể sao chép và đọc ra ngoài.



Hình 2. Lưu đồ sinh khóa trên thiết bị PKI Token

Quá trình sinh khóa RSA trên thiết bị diễn ra như lưu đồ trong Hình 2. Để xây dựng module sinh tham số RSA trên thiết bị PKI Token theo lưu đồ trên, cần thực hiện một số nội dung sau: xây dựng module sinh khóa, module lưu khóa, module xuất khóa công khai và tích hợp module sinh tham số RSA vào firmware thiết bị PKI Token.

A. Nghiên cứu xây dựng module firmware sinh khóa

Việc giao tiếp giữa phần mềm trên máy tính với thiết bị được thực hiện thông qua các lệnh APDU (Application Protocol Data Unit), trong đó sẽ cho phép phần mềm máy tính gửi xuống thiết bị lệnh cần thực hiện và dữ liệu gửi xuống thiết bị. Danh sách các lệnh cần thực hiện sẽ được định nghĩa và thống nhất chung giữa phần mềm trên máy tính và thiết bị. Do đó, đầu tiên cần định nghĩa cấu trúc lệnh sinh khóa như trong các Bảng 1 và Bảng 2.

BẢNG 1. CẤU TRÚC APDU LỆNH

CLA	00
INS	0x47
P1	0x80 – Tạo cặp khóa RSA 0x81 – Xuất khóa công khai
P2	0x00
Lc	02
Trường dữ liệu	vị trí khóa
Le	00

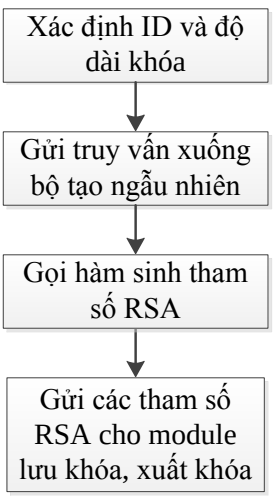
BẢNG 2. CẤU TRÚC APDU PHẢN HỒI

Trường dữ liệu	Khóa công khai
SW1-SW2	Trạng thái trả về

Ở đây, vị trí khóa cần sinh được quy định thống nhất giữa phần mềm máy tính và thiết bị, cụ thể là:

- Khóa số 01: 0xB600;
- Khóa số 02: 0xB800;
- Khóa số 03: 0xA400;
- Khóa số 04: 0xBA00;
- Khóa số 05: 0xBC00.

Khi thiết bị nhận được lệnh ADPU với câu lệnh được định nghĩa như trên, dữ liệu gửi xuống từ phần mềm máy tính sẽ được chuyển cho hàm sinh cặp khóa RSA. Tại đây, đầu tiên hàm sinh cặp khóa sẽ kiểm tra xem mã Admin PIN đã được kiểm tra trước đó chưa. Nếu trạng thái trả về cho thấy mã Admin PIN chưa được kiểm tra trước đó, hàm sẽ trả ra thông báo lỗi "không được phép tạo khóa" lên cho phần mềm máy tính thông qua APDU. Trong trường hợp việc kiểm tra mã Admin PIN trước đó là thành công, hàm tạo khóa sẽ được gọi để sinh khóa cho thiết bị. Hàm tạo khóa này được xây dựng theo quy trình như trong Hình 3.

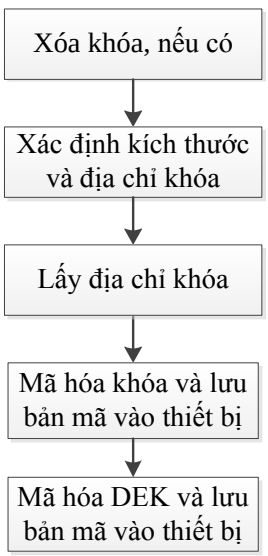


Hình 3. Quy trình hoạt động của module sinh khóa

Hoạt động cụ thể của hàm như sau: đầu tiên, hàm sẽ xác định vị trí khóa và độ dài khóa cần tạo, các tham số này được hàm gửi kèm trong lệnh sinh khóa từ trên phần mềm quản trị. Sau đó hàm gửi truy vấn xuống bộ tạo ngẫu nhiên, bộ tạo ngẫu nhiên sinh ra chuỗi ngẫu nhiên và gửi lại cho module sinh khóa. Module sinh khóa sử dụng chuỗi ngẫu nhiên đó là ứng cử viên đầu vào cho hàm sinh số nguyên tố. Sau khi nhận được chuỗi ngẫu nhiên từ bộ tạo ngẫu nhiên, hàm sẽ gọi hàm sinh tham số RSA. Các tham số RSA sau khi được tạo ra sẽ được gửi sang module lưu khóa. Cuối cùng module xuất khóa công khai sẽ được gọi để xuất khóa công khai từ cặp khóa RSA vừa sinh lên ứng dụng trên máy tính.

B. Nghiên cứu xây dựng module firmware lưu khóa

Tại module lưu khóa, sau khi nhận được các tham số RSA từ module sinh khóa, module lưu khóa sẽ tiến hành thực hiện các quy trình cần thiết để lưu khóa vừa sinh ra vào thiết bị. Khóa RSA sau khi được tạo ra sẽ được lưu vào thiết bị theo quy trình như trong Hình 4 (DEK – Data Encryption Key).



Hình 4. Quy trình hoạt động của module lưu khóa

Hoạt động của module lưu khóa được mô tả cụ thể như sau:

Bước 1: Xóa khóa tại vị trí cần lưu, nếu có. Đầu tiên, sau khi nhận được các tham số về vị trí khóa, dữ liệu khóa từ module sinh khóa, module lưu khóa sẽ tiến hành xóa khóa tại vị trí cần lưu. Bước này được thực hiện để đảm bảo giải phóng bộ nhớ phục vụ cho việc lưu khóa vào thiết bị. Ở đây cần lưu ý là, ở trên ứng dụng quản trị thiết bị PKI Token, trước khi gửi lệnh sinh khóa xuống thiết bị, ứng dụng cũng cần kiểm tra xem vị trí cần tạo đã có khóa hay chưa. Tuy nhiên thao tác này phụ thuộc hoàn toàn vào người lập trình ứng dụng. Vì thế ở phía dưới thiết bị, vẫn cần có thủ tục xóa khóa tại vị trí cần tạo, nhằm tránh trường hợp vị trí khóa cần lưu đã có khóa, khi đó việc ghi đè dữ liệu mới lên sẽ gây ra lỗi.

Bước 2: Xác định kích thước và địa chỉ khóa. Trong bước này, module sẽ xác định kích thước khóa cần lưu và địa chỉ sẽ lưu trong bộ nhớ flash của thiết bị. Kích thước khóa mặc định trong trường hợp này là 2048 bit, còn địa chỉ của khóa đã được quy định từ trước ứng với từng khóa cụ thể.

Bước 3: Mã hóa khóa và lưu bản mã vào thiết bị. Khóa bí mật trước khi lưu vào thiết bị sẽ được mã hóa bằng thuật toán MKC1A sử dụng khóa DEK và vector khởi tạo (IV) lấy từ bộ tạo ngẫu nhiên. Khóa bí mật sau khi được mã hóa sẽ được lưu vào bộ nhớ flash của thiết bị cùng với khóa công khai.

Bước 4: Mã hóa DEK và lưu bản mã vào thiết bị. Khóa DEK (dùng để mã hóa và giải mã khóa bí mật) sẽ được mã hóa sử dụng mã PIN của User và lưu vào bộ nhớ flash của thiết bị.

C. Nghiên cứu xây dựng module firmware xuất khóa công khai

Sau khi sinh khóa, khóa bí mật được mã hóa và lưu trong thiết bị, còn khóa công khai sẽ được thiết bị xuất ra ngoài. Khóa công khai được xuất ra ngoài theo định dạng chuẩn được quy định trong tiêu chuẩn ISO 7816-4, cụ thể như sau:

```
Tiêu đề:
7f49 xx (xx – là tổng độ dài của phản hồi)
Dữ liệu khóa công khai RSA:
   81 xx modulus (xx là độ dài số modulus)
   82 xx exponent (xx là độ dài của số mũ e)
```

Giá trị độ dài `xx` cần tuân theo quy định trong tiêu chuẩn ISO 7816-4 về số byte để lưu giá trị như sau: nếu số byte cần để lưu giá trị là 02 thì phía trước cần thêm byte chỉ số 0x82, nếu số byte cần lưu giá trị là 01 thì phía trước không cần thêm byte chỉ số.

Cụ thể, hoạt động của module firmware xuất khóa công khai được mô tả như sau:

Bước 1: Lấy dữ liệu khóa công khai. Dữ liệu khóa công khai gồm số modulus và số mũ công khai được đọc ra từ bộ nhớ flash của thiết bị.

Bước 2: Tạo phần tiêu đề cho dữ liệu phản hồi. Phần tiêu đề của dữ liệu phản hồi xuất khóa công khai được bắt đầu bằng `7f49` theo đúng chuẩn ISO 7816, tiếp theo đến tổng độ dài của phản hồi. Tổng độ dài của dữ liệu phản hồi gồm: độ dài của số modulus, độ dài của số mũ công khai và 6 bytes chỉ số (02 bytes chỉ số `7f49`, các byte còn lại sẽ được giải thích sau). Vì độ dài số modulus là 256 = 0x100 nên cần 02 bytes để lưu giá trị tổng độ dài của dữ liệu phản hồi, do đó ta có phần tiêu đề là `7f 49 82 xx xx`.

Bước 3: Tạo phần dữ liệu số modulus. Phần dữ liệu số modulus được bắt đầu bằng 81, tiếp theo đến độ dài của số modulus và số modulus. Vì cần 02 bytes để lưu giá trị độ dài của số modulus nên cần thêm chỉ số 82 sau chỉ số 81, cụ thể phần dữ liệu số modulus sẽ như sau: `81 82 01 00 modulus`.

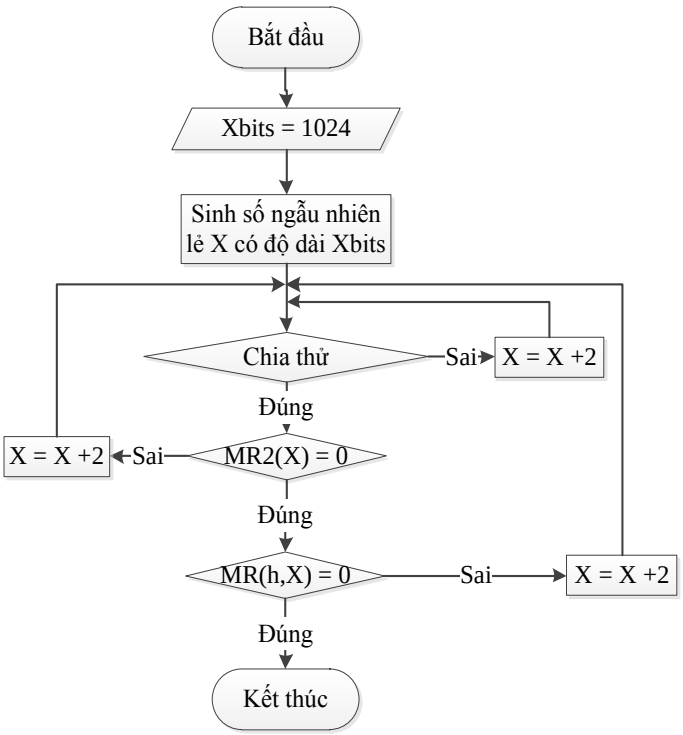
Bước 4: Tạo phần dữ liệu số mũ công khai. Phần dữ liệu số mũ công khai được bắt đầu bằng 82, tiếp theo đến độ dài của số mũ exponent, và số exponent. Vì chỉ cần 01 byte để lưu giá trị độ dài số mũ công khai, nên phần dữ liệu số mũ công khai sẽ như sau: `82 xx exponent`.

D. Nghiên cứu tích hợp module sinh tham số RSA vào firmware thiết bị PKI Token

Để tích hợp module sinh tham số RSA vào firmware thiết bị PKI Token, ngoài việc xây dựng thêm module sinh khóa, lưu khóa và xuất khóa công khai như trình bày ở trên, cần phải tùy biến, lựa chọn các hàm cần thiết trong thư viện polarSSL để tích hợp vào thiết bị. Trước tiên, trong hàm sinh số nguyên tố, nhóm tác giả tích hợp bộ tạo ngẫu nhiên trong thiết bị để sinh ra ứng cử viên ngẫu nhiên đầu vào. Do bộ nhớ của thiết bị PKI Token bị hạn chế, nên chỉ có thể đưa những thư viện cần thiết trong bộ thư viện polarSSL xuống thiết bị. Các thư viện cần thiết phục vụ việc sinh tham số RSA trên thiết bị gồm có thư viện tính toán số lớn và thư viện mật mã khóa công khai.

Để tăng tốc độ sinh tham số RSA trên thiết bị PKI Token, nhóm tác giả đã tiến hành cải tiến hàm sinh số nguyên tố trong bộ thư viện polarSSL. Theo [6], do xác suất một số nguyên ngẫu nhiên có ước số nguyên tố nhỏ là tương đối lớn và việc kiểm tra loại trừ các số nguyên ngẫu nhiên có ước số nguyên tố nhỏ ít tốn kém tài nguyên tính toán hơn kiểm tra Miller-Rabin, nên trước khi áp dụng kiểm tra Miller-Rabin, ứng cử viên ngẫu nhiên đầu vào nên được kiểm tra các ước nguyên tố nhỏ trước. Ngoài ra, theo [11], kiểm tra Miller-Rabin với cơ sở 2 có tốc độ thực thi nhanh hơn nhiều so với kiểm tra Miller-Rabin với cơ sở ngẫu nhiên khác. Vì vậy khi thêm kiểm tra Miller-Rabin cơ sở 2 trước kiểm tra Miller-Rabin với các cơ sở ngẫu nhiên khác nhau sẽ loại bỏ được phần lớn các hợp số trước khi thực hiện kiểm tra Miller-Rabin với các cơ sở ngẫu nhiên khác. Phương pháp giúp cải thiện tốc độ của phương pháp sinh số nguyên tố xác suất dựa trên kiểm tra Miller-Rabin mà không làm ảnh hưởng đến sai số của phương pháp. Bên cạnh đó, để tăng tính hiệu quả của hàm sinh số nguyên tố, nhóm tác giả lựa chọn giải pháp tìm kiếm tăng dần khi lựa chọn ứng cử viên đầu vào.

Lưu đồ thuật toán sinh số nguyên tố sau khi cải tiến được thể hiện trong Hình 5.



Hình 5. Lưu đồ thuật toán sinh số nguyên tố sau khi cải tiến

IV. NGHIÊN CỨU XÂY DỰNG CHƯƠNG TRÌNH TOKENADMIN HỖ TRỢ SINH THAM SỐ RSA

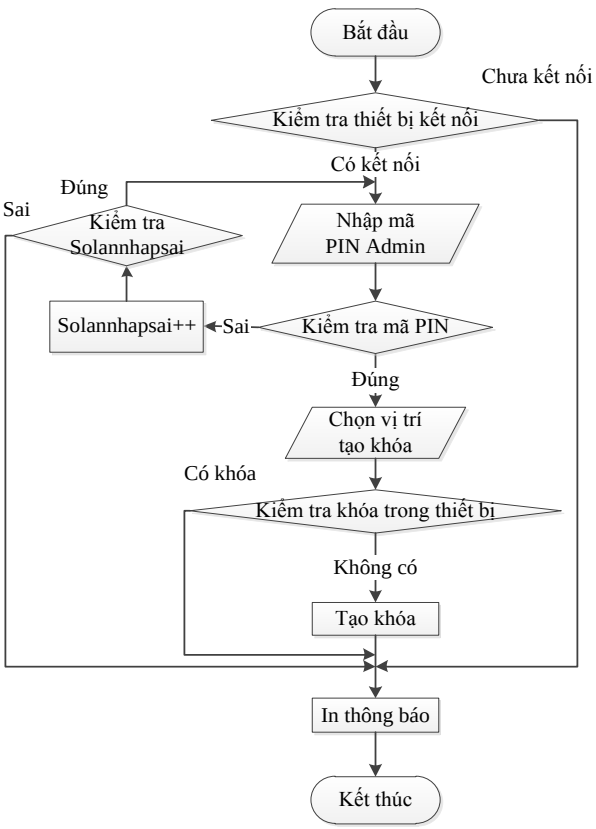
Phần mềm TokenAdmin là phần mềm chạy trên môi trường Windows dành cho người quản trị viên để quản lý, thực hiện các tác vụ cần thiết với thiết bị PKI Token. Để phần mềm hỗ trợ sinh tham số RSA trên thiết bị PKI Token, cần xây dựng thêm chức năng tạo khóa và xuất khóa công khai. Chức năng này trong phần mềm TokenAdmin được xây dựng dựa trên lưu đồ hoạt động như trong Hình 6 và Hình 7.

A. Nghiên cứu xây dựng và tích hợp module tạo khóa cho phần mềm quản trị TokenAdmin

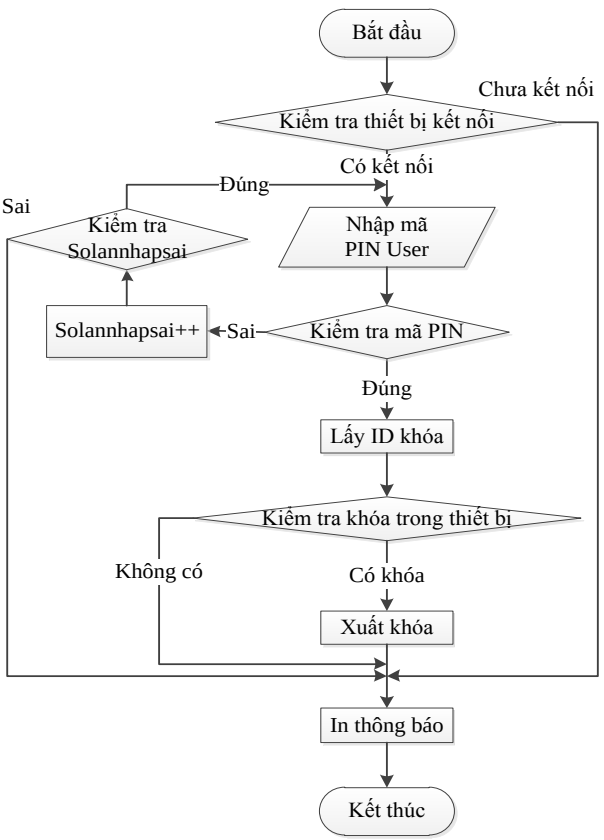
Chức năng tạo khóa có nhiệm vụ gửi lệnh tạo khóa xuống dưới thiết bị và nhận lại kết quả quá trình tạo khóa dưới thiết bị. Hoạt động cụ thể của module tạo khóa trong phần mềm quản trị PKI Token hoạt động như sau. Đầu tiên, module sẽ kiểm tra xem có thiết bị kết nối hay chưa, nếu chưa có thiết bị kết nối với máy tính, module sẽ hiển thị thông báo ra màn hình. Nếu có thiết bị kết nối, module sẽ yêu cầu người quản trị nhập vào mã PIN của Admin. Mã PIN của Admin nhập vào sẽ được kiểm tra thông qua các hàm kiểm tra mã PIN. Nếu mã PIN Admin nhập vào không đúng, module sẽ cho phép người quản trị nhập lại mã PIN Admin cho đến khi hết số lần nhập sai được phép. Nếu mã PIN Admin nhập vào hợp lệ, module sẽ yêu cầu người quản trị nhập vào vị trí khóa cần tạo. Tiếp đó, module sẽ kiểm tra vị trí khóa cần tạo đã có khóa chưa. Nếu chưa có khóa thì module sẽ gửi lệnh sinh khóa xuống thiết bị, ngược lại sẽ đưa ra thông báo. Quá trình sinh khóa trên thiết bị kết thúc, module nhận được thông báo về kết quả sinh khóa dưới thiết bị và đưa kết quả này ra thông báo.

B. Nghiên cứu xây dựng và tích hợp module xuất khóa công khai cho phần mềm quản trị TokenAdmin

Chức năng xuất khóa công khai giúp phần mềm gửi lệnh xuất khóa xuống thiết bị để lấy lên khóa công khai, lưu khóa công khai vào file phục vụ các mục đích cần thiết. Hoạt động cụ thể của module tạo khóa trong phần mềm quản trị PKI Token hoạt động như sau. Đầu tiên, module sẽ kiểm tra xem có thiết bị kết nối hay chưa, nếu chưa có thiết bị kết nối với máy tính, module sẽ hiển thị thông báo ra màn hình. Nếu có thiết bị kết nối, module sẽ yêu cầu người quản trị nhập vào mã PIN của User. Mã PIN của User nhập vào sẽ được kiểm tra thông qua các hàm kiểm tra mã PIN. Nếu mã PIN User nhập vào không đúng, module sẽ cho phép người dùng nhập lại mã PIN User cho đến khi hết số lần nhập sai được phép. Nếu mã PIN User nhập vào hợp lệ, module sẽ kiểm tra vị trí khóa cần xuất ra có khóa chưa. Nếu có khóa thì module sẽ gửi lệnh xuất khóa xuống thiết bị, ngược lại sẽ đưa ra thông báo. Thiết bị sẽ gửi khóa công khai lên máy tính, module sẽ yêu cầu người dùng nhập tên để lưu khóa công khai được xuất. Cuối cùng, module sẽ đưa ra thông báo về kết quả quá trình xuất khóa công khai.



Hình 6. Lưu đồ hoạt động module tạo khóa



Hình 7. Lưu đồ hoạt động module xuất khóa công khai

Sau khi xây dựng, tích hợp module tạo tham số RSA trên thiết bị PKI Token và bổ sung chức năng tạo khóa và xuất khóa công khai cho phần mềm quản trị PKI Token trên máy tính Windows, nhóm tác giả tiến hành kiểm thử hoạt động của module tạo tham số RSA trên thiết bị PKI Token.

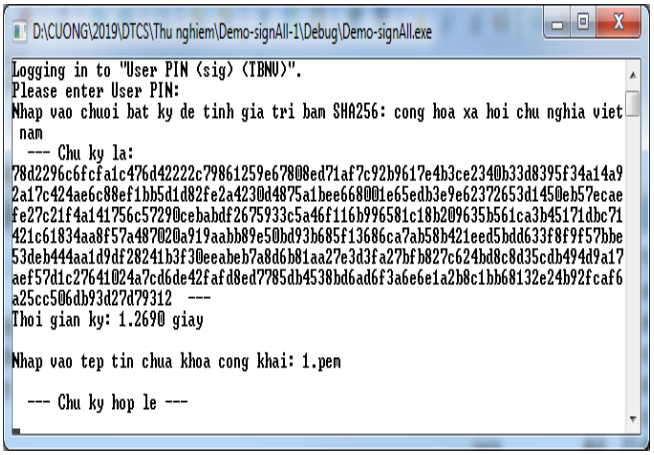
Nhóm tác giả đã thực hiện thử nghiệm hoạt động của module tạo tham số RSA 2048 bit trên thiết bị PKI Token, so sánh tốc độ sinh tham số RSA 2048 bit trên thiết bị PKI Token giữa module sử dụng hàm sinh số nguyên tố của thư viện polarSSL và module sử dụng hàm sinh số nguyên tố do nhóm tác giả cải tiến. Các thử nghiệm được thực hiện trên máy tính HP PAVILION 7000 SERIES 7000 SERIES sử dụng chip Intel® Core™ i3-2100 CPU tốc độ 3.10GHz, RAM 5.00GB. Kết quả thử nghiệm được thể hiện trong Bảng 3.

BẢNG 3. KẾT QUẢ SO SÁNH TỐC ĐỘ GIỮA MODULE SỬ DỤNG HÀM SINH SỐ NGUYÊN TỐ TRONG BỘ THƯ VIỆN POLARSSL VÀ MODULE SỬ DỤNG HÀM SINH SỐ NGUYÊN TỐ CẢI TIẾN

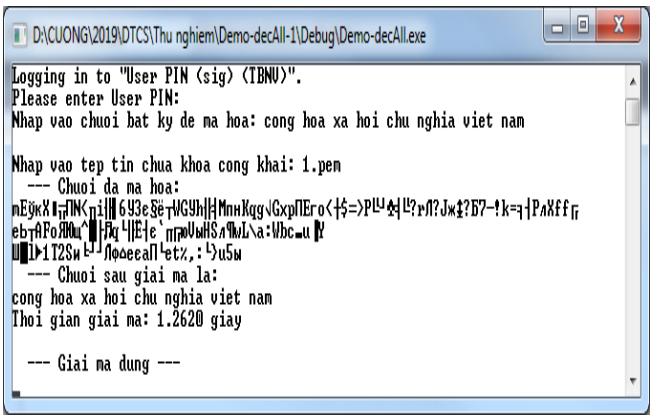
Lần thử	Module sử dụng hàm sinh số nguyên tố trong bộ thư viện polarSSL	Module sử dụng hàm sinh số nguyên tố do nhóm tác giả cải tiến
1	11 phút 45 giây	4 phút 45 giây
2	21 phút 30 giây	8 phút 6 giây
3	3 phút 44 giây	9 phút 27 giây
4	14 phút 6 giây	7 phút 24 giây
5	4 phút 11 giây	5 phút 32 giây
6	9 phút 10 giây	3 phút 33 giây
7	7 phút 30 giây	2 phút 24 giây
8	10 phút 5 giây	7 phút 51 giây
9	4 phút 46 giây	1 phút 7 giây
10	6 phút 38 giây	3 phút 37 giây
Trung bình	*9 phút 20 giây*	*5 phút 22 giây*

Như vậy, thời gian trung bình cần sinh tham số RSA 2048 bit trên thiết bị PKI Token sử dụng hàm sinh số nguyên tố trong thư viện polarSSL khoảng 9 phút 20 giây, còn khi sử dụng hàm sinh số nguyên tố do nhóm tác giả cải tiến khoảng 5 phút 22 giây. Do đó, sau khi cải tiến thuật toán, tốc độ sinh tham số RSA trên thiết bị PKI Token tăng tốc đáng kể.

Ngoài ra, nhóm tác giả cũng tiến hành thử nghiệm ký và kiểm tra chữ ký; mã hóa và giải mã sử dụng khóa sinh ra trong thiết bị. Kết quả thử nghiệm được thể hiện trong Hình 8 và Hình 9.



Hình 8. Kết quả thử nghiệm ký và kiểm tra chữ ký sử dụng khóa sinh ra trong thiết bị



Hình 9. Kết quả thử nghiệm mã hóa và giải mã sử dụng khóa sinh ra trong thiết bị

V. KẾT LUẬN

Bài báo đã trình bày việc nghiên cứu xây dựng và tích hợp module sinh tham số RSA an toàn lên thiết bị PKI Token. Trong quá trình xây dựng module sinh tham số RSA, nhóm tác giả đã tiến hành cải tiến hàm sinh số nguyên tố trong bộ thư viện polarSSL với mục đích tăng tốc độ sinh tham số RSA trên thiết bị PKI Token.

Các kết quả thử nghiệm cho thấy, module sinh tham số RSA trên thiết bị PKI Token hoạt động ổn định, đáp ứng các yêu cầu đặt ra. Ngoài ra, việc nghiên cứu và cải tiến hàm sinh số nguyên tố đã giúp tăng tốc đáng kể tốc độ sinh tham số RSA 2048 bit trên thiết bị PKI Token. Kết quả này là cơ sở để nhóm tác giả tiếp tục hoàn thiện module sinh tham số RSA trên thiết bị PKI Token.

TÀI LIỆU THAM KHẢO

[1] SafeNet eToken 5110. [02-04-2020]. url: https://www.digisign.ro/uploads/SafeNet-eToken-5110.pdf

[2] Cryptomate64. [02-04-2020]. url: http://www.cryptomate.com/.

[3] RuToken. [02-04-2020]. url: https://www.rutoken.ru/products/all/rutoken-ecp-pki/#spec

[4] National Institute of Standards and Technology. Digital Signature Standard (DSS). [02-04-2020]. url: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf.

[5] PolarSSL 1.2.10 released. [02-04-2020]. url: https://tls.mbed.org/tech-updates/releases/polarssl-1.2.10-released.

[6] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Crytography", CRC Press, 1996.

[7] Matúš Nemec, "The properties of RSA key generation process in software libraries", Brno, 2016.

[8] Hoàng Văn Thức, Luận án Tiến sĩ "Hệ tiêu chuẩn tham số an toàn cho hệ mật RSA và ứng dụng", 2011.

[9] Christophe Clavier, Benoit Feix, Loïc Thierry and Pascal Paillier, "Generating Provable Primes Efficiently on Embedded Devices", PKC 2012, LNCS 7293, pp. 372-389, 2012.

[10] Trần Duy Lai, Hoàng Văn Thức, Trần Sỹ Nam, "Thuật toán sinh số nguyên tố tất định hiệu quả trên thiết bị nhúng", Nghiên cứu Khoa học và Công nghệ trong lĩnh vực An toàn thông tin, ISSN 2615-9570, No 01. Vol 01. 2015.

[11] Jørgen Brandt, Ivan Damgård and Peter Landrock, "Speeding up Prime Number Generation", Advances in Cryptology – ASIACRYPT '91, pp. 440-449.

[12] Jørgen Brandt, Ivan Damgärd, "On generation of Probable primes by incremental search", Crypto'92, pp. 358-371.

SƠ LƯỢC VỀ TÁC GIẢ

TS. Vũ Tá Cường

Đơn vị công tác: Viện KH-CN mật mã, Ban Cơ yếu Chính phủ, Hà Nội

Email: vutacuong109@gmail.com

Quá trình đào tạo: Tốt nghiệp cử nhân năm 2011, thạc sĩ năm 2013 và tiến sĩ năm 2016 chuyên ngành "Vô tuyến điện tử", Đại học Hàng không vũ trụ Kharkov, Ucraina.

Hướng nghiên cứu hiện nay: PKI Token, kỹ thuật mật mã.

ThS. Nguyễn Thành Trung

Đơn vị công tác: Viện KH-CN mật mã, Ban Cơ yếu Chính phủ, Hà Nội

Email: trungbcy@gmail.com

Quá trình đào tạo: Tốt nghiệp cử nhân năm 1995, thạc sĩ năm 2005 chuyên ngành "Kỹ Thuật mật mã", Học viện Kỹ thuật Mật mã.

Hướng nghiên cứu hiện nay: PKI Token, kỹ thuật mật mã.

ThS. Lê Đình Hùng

Đơn vị công tác: Viện KH-CN mật mã, Ban Cơ yếu Chính phủ, Hà Nội

Email: ldhung85@gmail.com

Quá trình đào tạo: Tốt nghiệp cử nhân năm 2008 chuyên ngành "Điện tử viễn thông", Học viện Kỹ thuật quân sự, thạc sĩ năm 2017 chuyên ngành "Kỹ Thuật điện tử", Đại học Cộng nghệ - ĐHQG HN.

Hướng nghiên cứu hiện nay: PKI Token, kỹ thuật mật mã.