VỀ MỘT BACKDOOR TRONG SINH KHÓA RSA

Bạch Nhật Hồng[1], Lê Quang Huy[2*]

Tóm tắt: *Bài báo trình bày một đề xuất về thuật toán sinh khóa RSA chứa backdoor trên cơ sở cải tiến thuật toán PAP trong [1]. Thuật toán đề xuất sử dụng một hệ mật đối xứng để mã mật thông tin backdoor. Thuật toán đề xuất tốt hơn thuật toán PAP về tính bảo mật, lực lượng khóa và độ phức tạp tính toán.*

Từ khóa: Mật mã, Sinh khóa, RSA, Backdoor.

1. ĐẶT VẤN ĐỀ

Với hạ tầng mật mã khóa công khai (PKI), một vấn đề được quan tâm là việc khôi phục cặp khóa (dùng để mã) của người dùng khi có yêu cầu từ người dùng (mất khóa) hoặc từ cơ quan nhà nước (liên quan đến an ninh). Để giải quyết vấn đề trên người ta sử dụng hai cách. Cách thứ nhất là sử dụng giải pháp cơ sở dữ liệu (CSDL) khóa và các biện pháp đảm bảo an toàn cho CSDL khóa như giải pháp mềm (mã mật CSDL khóa) và giải pháp vật lý (bảo vệ máy tính…). Giải pháp sử dụng CSDL khóa có ưu điểm là các tính chất của khóa không bị giới hạn (so với thuật toán chuẩn) nhưng có nhược điểm là chi phí lớn và vận hành phức tạp. Cách thứ hai là sử dụng thuật toán sinh khóa chứa backdoor. Ưu điểm của giải pháp sử dụng backdoor là chi phí thấp, vì khi sử dụng, hệ thống không cần lưu giữ khóa riêng của người dùng (không duy trì CSDL khóa) mà chỉ cần lưu giữ và bảo vệ khóa của người thiết kế. Nhược điểm của giải pháp sử dụng thuật toán sinh khóa chứa backdoor là các tính chất khóa bị thu hẹp so với thuật toán sinh khóa chuẩn.

Với giải pháp thứ hai, nhiều thuật toán sinh khóa chứa backdoor trên một số hệ mật đã được công bố như: RSA, Elgamal, Eliptic Curve. Các thuật toán sinh khóa RSA chứa backdoor trong [1], [2] được đánh giá là kém an toàn vì độ dài khóa của người thiết kế bằng một nửa so với độ dài khóa của người dùng. Các thuật toán sinh khóa RSA chứa backdoor trong [3], [4] có độ phức tạp gần với bậc hai, có lực lượng ít hơn so với thuật toán trong [1] và cần sử dụng bộ nhớ không mất dữ liệu để lưu thông tin. Các thuật toán sinh khóa Elgamal trong [1] cũng bị đánh giá là kém an toàn vì độ dài khóa của người thiết kế (Elgamal) không tương đương với độ dài khóa của người dùng (RSA).

Do vậy, mục tiêu nghiên cứu ngoài việc tăng thêm hiểu biết để có giải pháp phòng vệ tốt hơn trong việc sử dụng các sản phẩm mật mã khi không làm chủ được mà còn có khả năng ứng dụng, bài báo tập trung nghiên cứu các thuật toán sinh khóa RSA chứa backdoor và đề xuất một thuật toán sinh khóa RSA chứa backdoor mới trên cơ sở cải tiến thuật toán PAP trong [1]. Thực hiện mục tiêu trên, bài báo được tổ chức thành 5 phần: mục 1-Đặt vấn đề, nêu lên sự cần thiết nghiên cứu và một số kết quả nghiên cứu của các tác giả đi trước; Mục 2-Các định nghĩa và cơ sở phục vụ cho việc phân tích thuật toán sinh khóa chứa backdoor; Mục 3-Trình bày và đánh giá thuật toán PAP; Mục 4-Đề xuất thuật toán sinh khóa chứa backdoor mới; Mục 5-Kết luận tóm tắt các kết quả nghiên cứu và hướng phát triển.

2. CÁC ĐỊNH NGHĨA VÀ CƠ SỞ

2.1. Thuật toán sinh khóa chứa backdoor

2.1.1. Một số ký hiệu

Gọi người thiết kế thuật toán sinh khóa chứa backdoor là người thiết kế và người dùng cặp khóa do thuật toán sinh khóa chứa backdoor tạo ra là người dùng.

1. Ký hiệu các thuật toán: Ký hiệu G_0 là thuật toán tạo khóa trung thực, tạo các khóa hợp lệ thuộc không gian khóa *KS*. Ký hiệu G_1 là thuật toán tạo khóa độc hại tạo các khóa chứa backdoor thuộc không gian khóa *KSM*. Ký hiệu k là tham số an toàn của thuật toán sinh khóa. Ta có:

$$\left(k_{priv}^{G_0}\,;\, k_{pub}^{G_0}\right) = G_0(1^k) \leftrightarrow \left(k_{priv}^{G_0},\,;\, k_{pub}^{G_0}\right) \in_U KS$$
$$\left(k_{priv}^{G_1}\,;\, k_{pub}^{G_1}\right) = G_1(1^k) \leftrightarrow \left(k_{priv}^{G_1},\,;\, k_{pub}^{G_1}\right) \in_U KSM$$

trong đó:

- $\in_U$ *KS*, $\in_U$ *KSM* nghĩa là các khóa được chọn ngẫu nhiên trong *KS, KSM*.
- $k_{priv}^{G_0}$, $k_{pub}^{G_0}$ lần lượt là ký hiệu khóa riêng và khóa công khai của G_0.
- $k_{priv}^{G_1}$, $k_{pub}^{G_1}$ lần lượt là ký hiệu khóa riêng và khóa công khai của G_1.

Ký hiệu R_1 là thuật toán khôi phục cặp khóa sao cho từ khóa công khai có thể khôi phục lại khóa riêng tương ứng. Ta có: $k_{priv}^{G_1} = R_1(k_{pub}^{G_1})$.

2. Ký hiệu các hàm bên trong *KSM*: Cấu trúc của *KSM* được diễn đạt bởi hai hàm E_{KSM} và D_{KSM} được mô tả như sau:

a) $k_{pub*}^{G_1} = E_{KSM}(k_{priv}^{G_1})$, ký hiệu $k_{pub*}^{G_1}$ là một phần khóa công khai của G_1, hàm E_{KSM} đưa thông tin khóa riêng vào khóa công khai.

b) $k_{priv}^{G_1} = D_{KSM}(k_{pub}^{G_1})$, hàm D_{KSM} khôi phục khóa riêng từ khóa công khai.

Các hàm E_{KSM} và D_{KSM} là sự kết hợp của ba hàm sau và các hàm nghịch đảo tương ứng:

Hàm thứ nhất: Gọi là hàm trích thông tin, ký hiệu là *I,* với mục đích trích từ khóa riêng $k_{priv}^{G_1}$ các thông tin $I(k_{priv}^{G_1})$ để nhúng trong khóa công khai $k_{pub}^{G_1}$ sao cho thông tin được nhúng đủ để khôi phục $k_{priv}^{G_1}$. Hàm *I* cần thỏa mãn:

a) $\left|I(k_{priv}^{G_1})\right| < \left|k_{priv}^{G_1}\right|$; Với $\left|k_{priv}^{G_1}\right|$ là độ dài tính theo bit của $k_{priv}^{G_1}$

b) $\left|I(k_{priv}^{G_1})\right| \ll \left|k_{pub}^{G_1}\right|$; Với $\left|k_{pub}^{G_1}\right|$ là độ dài tính theo bit của $k_{pub}^{G_1}$

Hàm thứ hai: Gọi là hàm che giấu thông tin, ký hiệu *E*. Hàm *E* dựa trên hệ mật đối xứng hoặc hệ mật bất đối xứng sao cho phân phối đầu ra của *E* không thể phân biệt được với phân phối đều về mặt tính toán. Thông tin giấu: $E\left(I(k_{priv}^{G_1})\right)$.

Hàm thứ ba: Được gọi là hàm nhúng, ký hiệu *f*. Nó xác định vị trí nhúng thông tin backdoor (đã được mã mật) và gắn giá trị ngẫu nhiên cùng thông tin backdoor (đã mã mật) vào trong các phần (bộ phận) của khóa công khai.

$$k_{pub*}^{G_1} = f\left(E\left(I(k_{priv}^{G_1})\right)\right).$$

Vậy ta có $E_{KSM} = f \circ E \circ I$, và $D_{KSM} = I^{-1} \circ E^{-1} \circ f^{-1}$

Ký hiệu k_B là một bộ phận của khóa công khai $k_{pub}^{G_1}$ chứa thông tin backdoor đã được mã mật hóa (nhúng).VD với sinh khóa RSA chứa backdoor, k_B là n hoặc e.

2.1.2. Định nghĩa thuật toán sinh khóa chứa backdoors

Ký hiệu thuật toán sinh khóa của một hệ mật mã khóa công khai là G_0. Một người thiết kế tạo ra một thuật toán sinh cặp khóa, G_1, và một thuật toán khôi phục cặp khóa R_1. Gọi I là hàm nén thông tin, gọi E là hàm mã mật hóa và D là hàm giải mã mật của E, gọi f_R là hàm nhúng thông tin của G_1. Cặp khóa $\left(k_{pub}^{G_1},\ k_{priv}^{G_1}\right)$ là các khóa chứa backdoor an toàn nếu các thuộc tính sau được thỏa mãn:

1. Bảo mật: Từ khóa công khai $k_{pub}^{G_1}$, người dùng không thể tính toán (khôi phục) được khóa riêng tương ứng $k_{priv}^{G_1}$.

2. Hoàn chỉnh: Các hàm I và f khả nghịch, để người thiết kế có thể khôi phục được khóa riêng từ khóa công khai tương ứng, $k_{priv}^{G_1} = R_1\left(f \circ E \circ I\left(k_{priv}^{G_1}\right)\right)$

3. Không thể phân biệt được: Với các thuật toán sinh khóa G_1 và G_0. Thì:

a) Các kết quả đầu ra của G_0, G_1 là không thể phân biệt được về mặt thống kê, hoặc về mặt tính toán.

b) Các đo đạc bên ngoài của G_0, G_1 là không thể phân biệt được một cách rõ ràng cái này với cái kia.

2.2. Một số kết quả của hệ mật RSA

2.2.1. Định lý về số các số nguyên tố

Ký hiệu $\pi(n)$ là số lượng các số nguyên tố nhỏ hơn hoặc bằng n.

Thì khi n lớn, ta có: $\pi(n) \sim \frac{n}{\ln n}$

Vậy xác suất một số nguyên ngẫu nhiên k-bit là số nguyên tố là:

$$\frac{\pi(2^k)}{2^k} \approx \frac{1}{\ln 2^k} \approx \frac{2}{2k \ln 2} \approx \frac{3}{2k} \approx 2^{-\ln k}$$

2.2.2. Số lượng khóa có thể sinh được

Số lượng các số nguyên tố k-bit:

$$\#\{p\} = \pi(2^k) - \pi(2^{k-1}) = \frac{(k-2)2^{k-1}}{k(k-1)\ln 2} \approx \frac{2^{k-1}}{k \ln 2} \approx 1{,}44.\, 2^{k-1-log_2 k}$$

Với mỗi tham số e xác định một tham số d duy nhất, tham số e có thể được chọn ngẫu nhiên trong $Z^*_{\varphi(n)}$. Số các số nguyên như vậy là số các số nguyên nguyên tố cùng nhau với $\varphi(n)$, và bằng $\varphi(\varphi(n))$. Từ [5, Fact 2, p102], ta có:

$$\frac{n}{6 \ln\ln n} < \varphi(n) < n, \qquad \frac{n}{36\,(\ln\ln n)\ln(n/(6\ln\ln n))} < \varphi(\varphi(n)) < n$$

Trong đó, ln ln n là viết tắt của ln(ln(n)).

Chia n cho ln n không ảnh hưởng nhiều đến kết quả nên ta xấp xỉ $\frac{n}{\ln n} \approx n$. $\varphi(n)$ hoặc $\varphi(\varphi(n))$ có thể được xấp xỉ bằng n. Với mỗi giá trị của e xác định duy nhất một nghịch đảo d, nên ta có lực lượng khóa của thuật toán sinh khóa RSA là:

$\#\{(p,q,d,e)\} = \#\{(p, q)\} \bullet \#\{e\} \approx (1{,}44.\, 2^{k\text{-}1\text{-}\log_2 k})^2 \bullet \varphi(\varphi(n)) \approx n.\, 2.\, 2^{2k\text{-}2\text{-}2\log_2 k}$

3. THUẬT TOÁN PAP

3.1. Giới thiệu thuật toán PAP

Thuật toán sinh khóa RSA chứa backdoor trong [1] được gọi là PAP, G_1 = PAP). Thuật toán PAP sử dụng chính hệ mật RSA để mã mật thông tin backdoor, E = RSA. Tham số modulus của người thiết kế (N) có độ dài bằng một nửa so với độ dài tham số modulus của người dùng (n). Thông tin backdoor (số nguyên tố p, $I(k_{priv}^{G_1}) = p$ được giấu ở nửa các bit cao của n, $k_B = n$, $f = \rho.2^k + r$ // $(\rho : r)$ (ρ *nối với* r). Các tham số thuật toán gồm:

+ Ký hiệu khóa công khai của người thiết kế là (N, E_1) và khóa riêng là D_1.

+ Hàm giả ngẫu nhiên F_K: $\{0, 1\}^k \rightarrow \{0, 1\}^k$, khả nghịch và chỉ xét các giá trị thỏa mãn $F_K(x) < N$.

+ Hàm G_K: $\{0, 1\}^k \rightarrow \{0, 1\}^k$ là hàm giả ngẫu nhiên khả nghịch và để đảm bảo $|G_K(x)| = k$.

+ Ký hiệu $a : b$ là xâu a nối với xâu b, xâu nối có giá trị $a.2^k + b$

Thuật toán PAP [sinh khóa RSA]

Input (k, K, E_1, N, B_1, B_2)
Output (p, q, d, e)
1. **repeat**
2. **repeat**
3. Generate random prime p//$|p| = k$
4. for $i = 0$ to B_1
5. $\rho_1 = F_{K+i}(p)$
6. if $\rho_1 < N$ then break
7. **until** $\rho_1 < N$
8. $\rho_2 \equiv \rho_1^{E_1} \bmod N$
9. **for** $j = 0$ **to** B_2
10. $\rho = G_{K+j}(\rho_2)$ //$|\rho| = k$
11. Generate a random r //$|r| = k$
12. $n_1 = \rho.2^k + r$ //$(\rho : r)$
13. $q = [n_1 / p]$
14. if q is prime then break
15. **until** q is prime
16. $n = p.q$
17. Generate a random e
// until gcd(e, $\varphi(n)$) = 1)
18. $d \equiv e^{-1} \bmod \varphi(n)$
19. return (p, q, d, e)

Thuật toán PAP [khôi phục khóa RSA]

Input (n, e, D_1, N, K, B_1, B_2)
Output (d)
1. **for** $m = 0$ **to** 1
2. $\rho.2^k + r = n$
3. $\rho = \rho + m$
4. for $j = 0$ to B_2
5. $\rho_2 = G^{-1}_{K+j}(\rho)$
6. $\rho_3 \equiv \rho_2^{D_1} \bmod N$
7. for $i = 0$ to B_1
8. $p = F^{-1}_{K+i}(\rho_3)$
9. **if** ($p \mid n$) **then break**
10. **if** ($p \mid n$) **then break**
11. **if** ($p \mid n$) **then break**
12. $q = n / p$
13. $d \equiv e^{-1} \bmod \varphi(n)$
14. return (d)

3.2. Đánh giá thuật toán PAP

Tính bảo mật: Tham số modulus của người thiết kế có chiều dài bằng một nửa tham số modulus của người dùng khiến cho backdoor có điểm yếu và do đó, tính bảo mật có thể không được đảm bảo.

Lực lượng:

Vì p được sinh ngẫu nhiên nên theo định lý về số lượng các số nguyên tố, ta có: $\#\{p\} = 1{,}44.\ 2^{k-1-log_2 k}$.

Xét cách tạo q (từ bước 7 đến bước 12): Số lượng xâu ρ: r chia hết cho p là

$$\#\{\frac{\rho:r}{p} \in Z\} = \#\{\rho: r\}.\frac{1}{p} = 2^k.\frac{1}{p} = 2^k.p^{-1}$$

Theo định lý về số các số nguyên tố: một số nguyên k-bit là số nguyên tố với xác xuất khoảng $2^{-\ln k}$.

$\#\{q\}$ = Số lượng xâu ρ: r chia hết cho p là số nguyên tố là

$$\#\{q\} = \#\left\{\frac{\rho:r}{p} \in Z\right\}.\Pr\left[\frac{\rho:r}{p}\ là\ số\ nguyên\ tố\right] = 2^k.p^{-1}.2^{-\ln k} = 2^{k-\ln k}.p^{-1}$$

Vậy, số lượng phần tử n ($\#\{n\}$) là:

$$N_{G_1,n} = \#\{n\} = \#\{p\}.\#\{q\} = 1{,}44.\ 2^{k-1-log_2 k}.2^{k-\ln k} = 1{,}44.\ 2^{2k-1-\ln k-log_2 k}.p^{-1}$$

Xét tỷ lệ giữa lực lượng của *KSM* và *KS*, vì hạng tử $\#\{e\}$ giống nhau giữa hai thuật toán sinh khóa nên có thể bỏ qua. Do đó, tỷ lệ giữa hai lực lượng:

$$R_{G_1} = \frac{N_{G_1,n}}{N_{G_0,n}} = \frac{1{,}44\ 2^{2k-1-\ln k-log_2 k}.p^{-1}}{1{,}44^2.2^{2k-2-2log_2 k}} = 2.\ln 2\ .2^{log_2 k-\ln k}.p^{-1}$$

Độ phức tạp: Vì $|p| = |N|$ và $F_{K+i}(p)$ có thể xem là một hoán vị, do đó xác xuất $F_{K+i}(p) < N$ lớn hơn ½. Do vậy, vòng lặp từ 2 đến 7 là không đáng kể. Tương tự, vòng lặp cố định từ bước 9 đến bước 14 với tối đa $B_2 = 16$. Do đó, thuật toán có bước lặp tạo q từ bước 1 đến bước 15, vì $q = [(\rho.2^k + r) / p]$ là một số nguyên giả ngẫu nhiên, nên độ phức tạp của việc tạo q trong G_1 cũng giống như trong G_0; $t_q(G_1) = B_2$. $t_q(G_0)$ tương đương với $t_q(G_1) = t_q(G_0)$. Cách tạo p cũng giống như trong G_0, tuy nhiên p được tạo bên trong vòng lặp tạo q nên và $t_p(G_1) = t_p(G_0)$. $t_q(G_0)$. Độ phức tạp trong tạo n là:

$t_n(G_1) = t_p\ .\ t_q\ +\ t_q + T(F_K) + T(\text{RSA-}k) + T(G_K)$

Việc tạo e giống như G_0 nên ta có $t_e(G_1) = t_e(G_0)$

Vậy độ phức tạp của thuật toán là:

$T(G_1) = t_e + t_p\ .\ t_q\ +\ t_q + T(F_K) + T(G_K) + T(\text{RSA-}k)$.

4. ĐỀ XUẤT THUẬT TOÁN SINH KHÓA RSA CHỨA BACKDOOR MỚI

4.1. Giới thiệu thuật toán

Phần này, trình bày một thuật toán sinh khóa RSA chứa backdoor trên cơ sở cải tiến thuật toán PAP trong [1]. Thuật toán đề xuất (G_1'=thuật toán đề xuất) sử dụng một mã khối đối xứng (E = mã khối đối xứng) để mã mật thông tin backdoor (số nguyên tố p, $I(k_{priv}^{G_1}) = p$. Thông tin backdoor được giấu trong các bit thấp của n, $k_B = n$; $f = s.2^k + r$ //(s : r) (s *nối với* r). Thuật toán đề xuất sử dụng kết quả của định lý sau:

Định lý 4.1. Ký hiệu p là một số nguyên tố, r là một số nguyên lẻ với $|p| = |r| = k$. Gọi $u = 2^k \bmod p$; $r_0 = r \bmod p$; $s_0 = (r_0 + u^2).u^{-1} \bmod p$; $s = 2^k - s_0$; $n = r + s.2^k$. Thì ta có $p \mid n$.

Chứng minh:

$u = 2^k \bmod p \Leftrightarrow 2^k = m.p + u$ với $m \in Z$

$r_0 = r \bmod p \Leftrightarrow r = l.p + r_0$ với $l \in Z$

Từ $n = s.2^k + r = 2^k.(2^k - s_0) + l.p + r_0 = (m.p + u).(m.p + u - s_0) + l.p + r_0$

$= m^2p^2 + (2mu - ms_0).p + u^2 - us_0 + l.p + r_0$

$= m^2p^2 + (2mu - ms_0 + l).p + u^2 + r_0 - u(r_0 + u^2).u^{-1} \bmod p$

$= (m^2p + 2mu - ms_0 + l).p \bmod p$

Vậy $p \mid n$

Các tham số thuật toán đề xuất:

+ Hàm mã khối đối xứng F_K: $\{0, 1\}^l \rightarrow \{0, 1\}^l$ để che giấu thông tin backdoor (số nguyên tố *p),* ví dụ F = AES 128 (l =128).

Thuật toán đề xuất [sinh khóa RSA]

Input (k, K, B_1)
Output (p, q, n, d)
1. **repeat**
2. Generate a random prime p
3. $u = 2^k \bmod p$
4. **for** i = 0 **to** B_1
5. $r = F_{K+i}(p)$;
6. **if** $r \bmod 2 = 0$ **then** $r = r + 1$
7. $r_0 = r \bmod p$
8. $s_0 = (r_0 + u^2).u^{-1} \bmod p$;
9. **if** ($s_0 \leq 2^{k-1}$) **then**
10. $s = 2^k - s_0$
11. $n = s.2^k + r$; //n is multiple of p
12. $q = n / p$
13. **if** q is prime **then break**
14. **until** q is prime
15. Generate a random e//gcd(e,$\varphi(n)$)=1
16. $d = e^{-1} \bmod \varphi(n)$
17. **return** (p, q, n, d).

Thuật toán đề xuất [khôi phục khóa RSA]

Input (n, e, K, B_1)
Output (p, q, d)
1. $k = |n| / 2$
2. $r = n \bmod 2^k$
3. **for** m = 0 **to** 1
4. $r = r - m$
5. **for** i = 0 **to** B_1
6. $p = F^{-1}_{K+i}(r)$
7. **if** ($p \mid n$) **then break**
8. **if** ($p \mid n$) **then break**
9. $q = n / p$
9. $d = e^{-1} \bmod \varphi(n)$
10. **return** (p, q, d)

4.2. Đánh giá thuật toán đề xuất

Tính bảo mật: Thuật toán đề xuất an toàn vì sử dụng mã khối đối xứng có độ dài khóa phù hợp (ví dụ AES 128), để mã mật thông tin backdoor.

Lực lượng: Xét lực lượng của p. Vì p được sinh ngẫu nhiên nên theo định lý về số lượng các số nguyên tố (mục 2.3.), ta có $\#\{p\} = 1{,}44.\ 2^{k-1-\log_2 k}$.

Xét lực lượng của q. Ở bước 5, ta có $r = F_{K+i}(p)$ vì hàm F_K là một mã khối đối xứng nên r có thể xem như một hoán vị giả ngẫu nhiên của p. Với mỗi giá trị của p thì có thể có nhiều nhất B_1 giá trị s. Do đó, số lượng phần tử nhiều nhất là $q = B_1$. Theo định lý về số các số nguyên tố (mục 2.3), một số nguyên k-bit là số nguyên tố với xác suất khoảng $2^{-\ln k}$. Vì vậy: $\#\{q \text{ là số nguyên tố}\} = B_1.\ 2^{-\ln k}$.

$\#\{n\} = \#\{p\}.\#\{q\} = 1{,}44.2^{k-1-log_2 k}.B_1.2^{-\ln k} = 1{,}44.B_1.2^{k-1-\ln k - log_2 k}$.

Xét tỷ lệ giữa lực lượng của *KSM* và *KS*, vì *e* được sinh tự do nên hạng tử $\#\{e\}$ giống nhau trong G_0 và G'_1 nên có thể bỏ qua, ta có:

$$R_{G_1} = \frac{N_{G_1,n}}{N_{G_0,n}} = \frac{1{,}44\,.B_1.2^{k-1-\ln k-log_2 k}}{1{,}44^2.2^{2k-2-2\,log_2 k}} = 2.B_1.\ln 2\,.2^{-k+log_2 k-\ln k}$$

Độ phức tạp: Cách tạo *p* giống nhau đối với G_0 và G'_1 nên $t_p(G_0) = t_p(G'_1)$.

Vì *p* được đặt trong vòng lặp tạo *q* và trong vòng lặp tạo *q* có thêm việc mã mật hóa thông tin backdoor, do đó, $t_q\,(G'_1) = t_q(G_0)\,.\,t_p\,(G_0) + T\,(F_K)$.

Vậy độ phức tạp trong tạo *n* là: $t_n\,(G'_1) = t_p + t_q.\,t_p + T\,(F_K)$.

Cách tạo *e* của G'_1 cũng giống với G_0, do đó, độ phức tạp trong việc tạo *e* là giống nhau nên ta có $t_e\,(G'_1) = t_e\,(G_0)$.

Vậy độ phức tạp của thuật toán này là: $T(G'_1) = t_p + t_q.\,t_p + \ t_e\ + T\,(F_K)$.

4.3. Tóm tắt các ưu điểm của thuật toán đề xuất

Thuật toán đề xuất ưu điểm hơn so với thuật toán PAP ở những điểm sau:

- Tính bảo mật: thuật toán đề xuất sử dụng mã khối đối xứng (có độ an toàn tương đương với khóa của người dùng) để mã mật thông tin backdoor nên tính bảo mật được đảm bảo. Thuật toán PAP có độ dài khóa của người thiết kế bằng một nửa độ dài khóa của người dùng nên tính bảo mật có điểm yếu.

- Lực lượng: Tỷ lệ lực lượng của thuật toán đề xuất ($R_{G_1} = 2.B_1.\ln 2\,.2^{-k+log_2 k-\ln k}$) lớn hơn (tốt hơn) một chút so với tỷ lệ lực lượng thuật toán PAP ($R_{G_1} = 2.\ln 2\,.2^{log_2 k-\ln k}.p^{-1}$).

- Độ phức tạp của thuật toán đề xuất, $T(G'_1) = t_p + t_q.\,t_p + \ t_e\ + T(F_K)$ ít phức tạp hơn độ phức tạp của thuật toán PAP, $T(G_1) = t_p + t_q.\,t_p + \ t_e + T(F_K) + T(G_K) + T(\text{RSA-}k)$.

5. KẾT LUẬN

Trên cơ sở thuật toán PAP, một thuật toán mới được đề xuất với các ưu điểm về tính bảo mật, lực lượng khóa và độ phức tạp tính toán hơn so với thuật toán PAP. Thuật toán này có thể ứng dụng tốt trong phần sinh khóa của thiết bị PKI Token hoặc HSM. Ngoài ra, thuật toán có thể được xem xét thêm theo hướng sử dụng hệ mật khác để mã mật hóa thông tin backdoor, hoặc rút bớt thông tin backdoor hoặc đề xuất phương pháp tìm kiếm số nguyên tố *q* hiệu quả hơn hoặc xem xét sự phù hợp của các tham số do thuật toán sinh ra với tiêu chuẩn tham số ứng dụng cho một hạ tầng PKI cụ thể.

TÀI LIỆU THAM KHẢO

[1]. Young A and Yung M, *"The Dark Side of 'Black-Box' Cryptography or: Should We Trust Capstone?"*, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.616&rep=rep1&type=pdf, 1996.

[2]. Young A and Yung M, *"Kleptography: Using Cryptography Against Cryptography"*, https://cryptome.org/2013/09/klepto-crypto.pdf, 1997.

[3]. Young A and Yung M, "*Malicious Cryptography: Kleptographic Aspects"*, https://pdfs.semanticscholar.org/6c9c/9bb21f1b52480df05ce7a9266436ff594535.pdf, 2005.

[4]. Young A and Yung M, "*A Space Efficient Backdoor in RSA and Its Applications"*, http://link.springer.com/chapter/10.1007%2F11693383_9#page-1, 2005.

[5]. A. Menezes, P. van Oorschot, and S. Vanstone, "*Handbook of Applied Cryptography"*, CRC Press, 2001.

ABSTRACT

ASYMETRIC BACKDOOR IN RSA KEY GENERATION

In this paper, a propose of a backdoored RSA key generation algorithm based on improvement of PAP algorithm in [1] is presented. The proposed algorithm uses RSA cryptosystem to encrypt backdoor information. The proposed algorithm is better than the PAP in security, cardinality, complexity.

Keywords: Cryptography, Key generation, RSA, Backdoor.

Nhận bài ngày 29 tháng 12 năm 2016
Hoàn thiện ngày 02 tháng 3 năm 2017
Chấp nhận đăng ngày 05 tháng 4 năm 2017

Địa chỉ: [1] Trường Đại học Sư phạm Kỹ thuật Hưng Yên;
[2] Cục Chứng thực số và Bảo mật thông tin - Ban Cơ yếu Chính phủ.
[*] Email: lequanghuyabc@gmail.com