TẤN CÔNG HỆ MÃ KHÓA RSA

Giảng viên hướng dẫn: PGS.TS Nguyễn Đình Hân

Sinh viên thực hiện:

Nguyễn Thị Quý	20185396
Phạm Thị Thu Hương	20185367
Hoàng Phương Cúc	20185332
Nguyễn Thị Diệu Linh	20180815
Đặng Thị Hồng Nhung	20185391

Nội dung

- Tổng quan hệ mã khóa công khai và RSA
- Mã hóa và giải mã RSA
- Một số phương án tấn công RSA
- Cài đặt tấn công
- Một số tấn công nhân tử hóa N với N lớn
- Ứng dụng của RSA

BẢNG PHÂN CÔNG CÔNG VIỆC

Thành viên	MSSV	Công việc
Phạm Thị Thu Hương	20185367	Cài đặt tấn công. Một số tấn công nhân tử hóa số N với N lớn
Nguyễn Thị Diệu Linh	20150815	Mã hóa, giải mã RSA, viết chương trình. Ứng dụng RSA
Nguyễn Thị Quý	20185396	Tấn công số mũ công khai, số mũ riêng, thành phần công khai bé
Hoàng Phương Cúc	20185332	Một số giả thiết ngầm, phân tích các số nguyên lớn. Phương thức tấn công cơ bản: Modul chung, Mù (blinding)
Đặng Thị Hồng Nhung	20185391	Tổng quan hệ mã hoá khoá công khai. Tổng quan hệ RSA. Slide

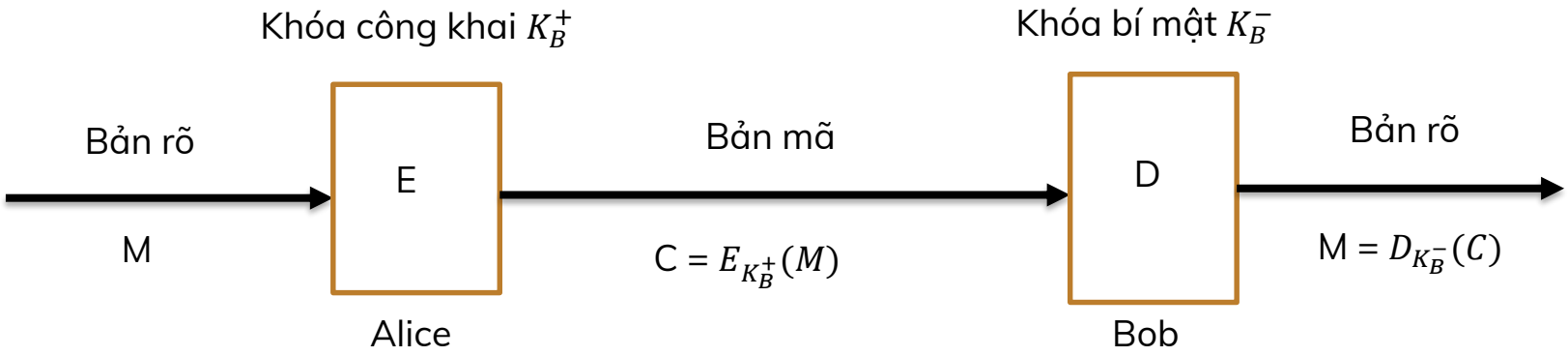
Tổng quan hệ mã hóa công khai và RSA

Khái niệm chung mã hóa công khai

- Mã hóa khóa công khai là một dạng mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa bí mật trước đó.
- Việc mã hóa công khai được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai (Public key) và khóa riêng (Private key) hay khóa bí mật (Secret key).
- Là mật mã hóa khoa bất đối xứng: 2 khóa dùng để mã hóa và giải mã không giống nhau.
- Trong mật mã khóa công khai, khóa riêng cần phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Điều quan trọng đối với hệ thống là không thể (hoặc rất khó) tìm ra khóa bí mật nếu chỉ biết khóa công khai.

Khái niệm chung mã hóa công khai

Quy trình



Mục đích sử dụng hệ thống mã hóa công khai

- **Mã hóa**: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được
- **Tạo chữ ký số**: cho phép kiểm tra một văn bản xem nó có phải được tạo với một khóa bí mật nào đó hay không.
- **Thỏa thuận khóa**: cho phép thiết lập khóa để trao đổi thông tin mật giữa hai bên

Ưu, nhược điểm của hệ mã hóa công khai

Ưu điểm:

- Thuật toán được viết một lần, công khai cho nhiều lần dùng, cho nhiều người dùng, họ chỉ cần giữ bí mật khóa riêng của mình.
- Khi biết các tham số ban đầu của hệ mã hóa, việc tính ra cặp khoá công khai và bí mật phải là “dễ”, tức là trong thời gian đa thức.
- Khả năng lộ khóa bí mật khó hơn vì chỉ có một người giữ gìn. Nếu thám mã biết khoá công khai, cố gắng tìm khoá bí mật, thì chúng phải đương đầu với bài toán “khó”.
- Nếu thám mã biết khoá công khai và bản mã C, thì việc tìm ra bản rõ M cũng là bài toán “khó”, số phép thử là vô cùng lớn, không khả thi.

Nhược điểm:

- Hệ mã hóa khóa công khai: mã hóa và giải mã chậm hơn hệ mã hóa khóa đối xứng

Nơi sử dụng hệ mã khóa công khai

- Thường được sử dụng chủ yếu trên các mạng công khai như Internet, khi mà việc trao chuyển khoá bí mật tương đối khó khăn.
- Đặc trưng nổi bật của hệ mã hoá công khai là khoá công khai (public key) và bản mã (ciphertext) đều có thể gửi đi trên một kênh truyền tin không an toàn.
- Do tốc độ mã hóa và giải mã chậm, nên hệ mã hóa khóa công khai chỉ dùng để mã hóa những bản tin ngắn, ví dụ như mã hóa khóa bí mật gửi đi.
- Được sử dụng cho cặp người dùng thỏa thuận khóa bí mật của Hệ mã hóa khóa riêng.

Hệ mã RSA

Tổng quan về RSA

- RSA là hệ mật mã khóa công khai phổ biến và cũng đa năng nhất trong thực tế, phát minh bởi Rivest, Shamir & Adleman (1977).
- Là chuẩn mật mã bất thành văn đối với PKC, cung cấp đảm bảo tính mật, xác thực và chữ ký điện tử.
- Cơ sở thuật toán RSA dựa trên tính khó của bài toán phân tích các số lớn ra thừa số nguyên tố: không tồn tại thuật toán thời gian đa thức (theo độ dài của biểu diễn nhị phân của số đó) cho bài toán này.

Tổng quan về RSA

- **Ý tưởng:** Xây dựng thuật toán sinh và giải mã trên cơ sở phép toan lấy lũy thừa đồng dư trên trường $Z_n = \{0, 1, 2, \dots, n-1\}$
- **Quy trình áp dụng RSA:**
  - + Lựa chọn (sinh) cặp khoa công khai và bí mật
  - + Thực hiện thuật toán mã hóa và giải mã.

Mã hóa & Giải mã RSA

Thuật toán sinh khóa

Bước 1: Sinh 2 số nguyên tố lớn p, q

Bước 2: Tính n=p.q và $\varphi(n) = (p-1).(q-1)$

Bước 3: Chọn một số ngẫu nhiên e, $1 < e < \varphi(n)$, sao cho $\gcd(e, \varphi(n)) = 1$

Bước 4: Sử dụng thuật toán Euclid để tìm số d, $1 < d < \varphi(n)$, sao cho $e.d \equiv 1 \pmod{\varphi(n)}$

Bước 5: Khóa công khai là $K_B^+ = (n, e)$, khóa bí mật là $K_B^- = (n, d)$.

Nguyên tắc thực hiện

1 Việc mã hóa thực hiện theo công thức:

- Theo phương án 1: Mã hóa bảo mật, ta có:

$$C = E(M, K_B^+) = M^e \bmod n$$

- Theo phương án 2: Mã hóa chứng thực, ta có:

$$C = E(M, K_B^-) = M^d \bmod n$$

Nguyên tắc thực hiện

2 Việc giải mã thực hiện theo công thức:

- Theo phương án 1: Giải mã bảo mật, ta có:

$$C = E(M, K_B^+) = M^e \bmod n$$

- Theo phương án 2: Giải mã chứng thực, ta có:

$$C = E(M, K_B^-) = M^d \bmod n$$

Ví dụ

Bản rõ M = 15 với p = 11, q = 3, e = 3.

1) Có 2 số nguyên tố p = 11, q = 3 nên n = p.q = 11.3 = 33

2) $\varphi(n) = (p-1).(q-1) = (11 - 1).(3 - 1) = 20$

3) Chọn 1 số e nguyên tố cùng nhau với 20 thì e = 3 thỏa mãn

4) Tính d là nghịch đảo của e trong modulo n tức là

$$d.e \equiv 1 \pmod{20}$$

Ta có: $d.3 \equiv 1 \pmod{20}$ thì d = 7 (thuật toán Euclid)

Hoặc Vì 7.3 = 21 mà 21 = 20.1 + 1 tức là 21 = 1 mod 20.

5) Khóa công khai $K_B^+ = (n, e) = (3, 33)$

Khóa bí mật $K_B^- = (n, d) = (7, 33)$

Ví dụ

Mã hóa bảo mật:

- Mã hóa bản rõ M = 15:

$C = M^e \bmod n = 15^3 \bmod 33 = 9$

- Mã hóa bản mã C = 9:

$M = C^d \bmod n = 9^7 \bmod 33 = 15$

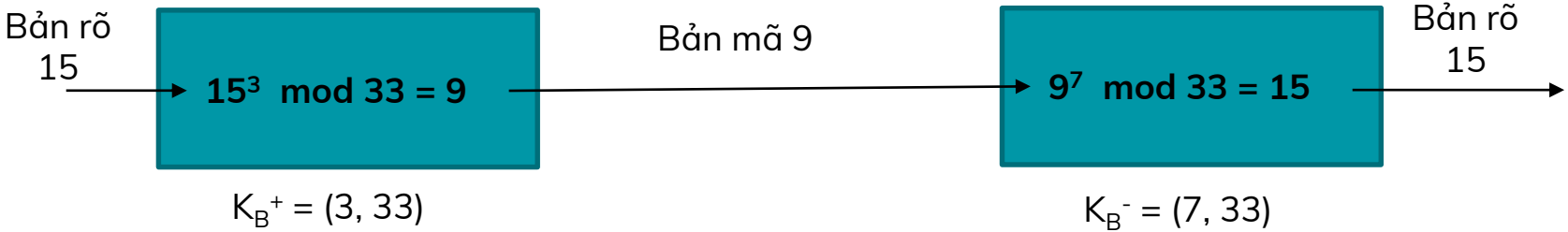
Mã hóa chứng thực:

- Mã hóa bản rõ M = 15:

$C = M^d \bmod n = 15^7 \bmod 33 = 27$

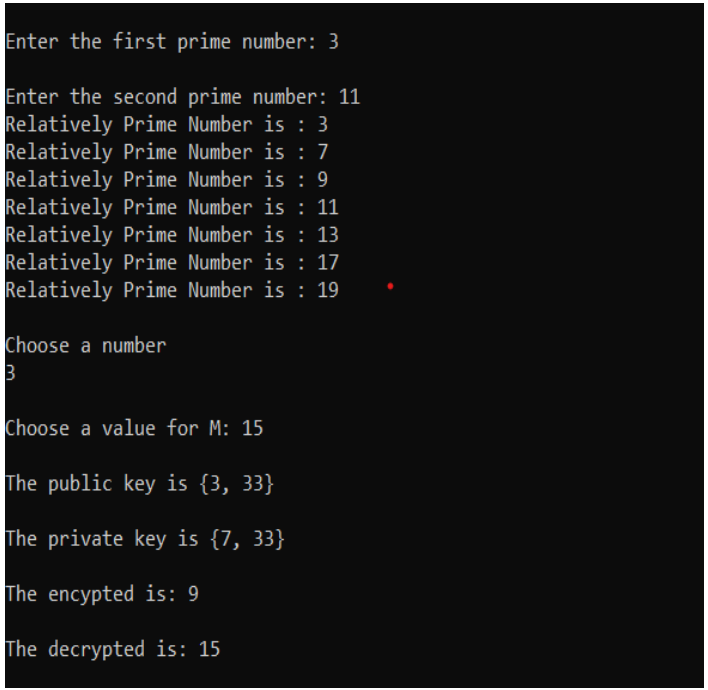
- Mã hóa bản mã C = 9:

$M = C^e \bmod n = 9^7 \bmod 33 = 15$

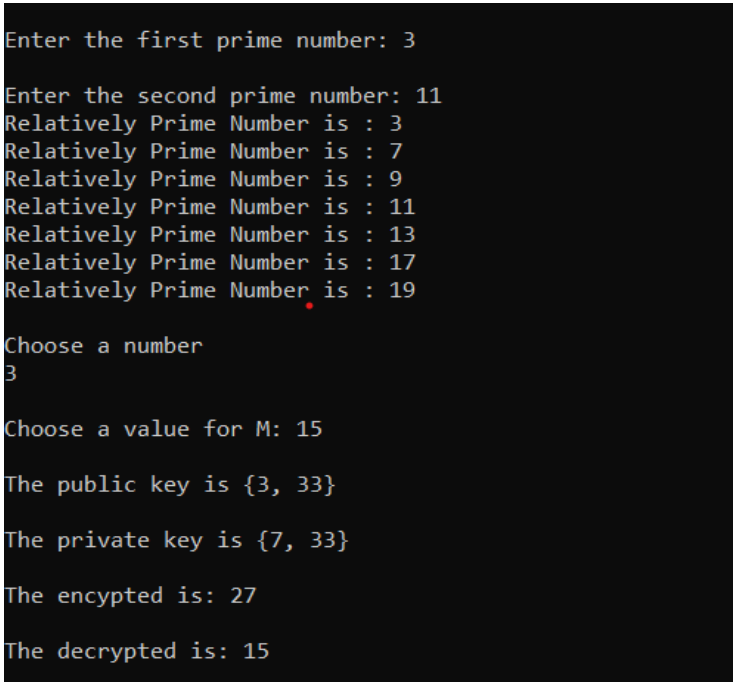


Ví dụ: Chạy chương trình đối chiếu kết quả

Mã hóa bảo mật



Mã hóa chứng thực



Một số phương pháp tấn công RSA

Một số giả thuyết ngầm định

- N – RSA modulus
- e – số mũ mã hóa (encryption exponent)
- d – số mũ giải mã (decryption exponent)
- M – thông điệp số nguyên (message integer), $M \in Z_N^*$
- Alice, Bob là đại diện hai bên truyền thông điệp cho nhau, Marvin là kẻ tấn công (attacker)
- Hàm RSA là một ánh xạ f: x → x^e mod N. Nếu d cho trước, hàm đó có thể dễ dàng nghịch đảo được bằng cách dùng phương trình trên. Chúng ta coi d như là một cửa sập (trapdoor) để nghịch đảo hàm f.

Một số giả thuyết ngầm định

- Nghiên cứu độ khó của hàm ngược (nghịch đảo) RSA khi không cho trước cửa sập d và nghiên cứu phương pháp tấn công RSA trong trường hợp này.
- Quan tâm đến thuật toán hữu hiệu có thời gian bé, tính bậc của n^c với điều kiện $n = \log_2 N$ và c là một hằng số nhỏ ($c < 5$).
- Về mặt lý thuyết, nếu cho trước (N,e,C), rất khó để tìm ra thông tin về M.
- Tấn công vét cạn (brute-force attack) bằng cách phân tích các modulus, thời gian chạy với số nguyên n-bít là $e^{((c+O(1))n^{\frac{1}{3}}(\log n)^{\frac{2}{3}})}$ trong đó c < 2.

Phân tích thành số nguyên lớn

- Vấn đề phân tích một số nguyên tố lớn thành tích các số nguyên tố khác nhau là bài toán rất hấp dẫn và đã được nhiều nhà toán học quan tâm nghiên cứu. Trong đó nhóm em sẽ chỉ tập trung nghiên cứu trong trường hợp N là tích của hai số nguyên tố phân biệt.
- ***Mệnh đề 1:*** Giả sử N là một số tự nhiên không chính phương (perfect square), tức N không phải là bình phương đúng của một số nguyên tố, thỏa mãn điều kiện:

$$N - 1 > \varphi(N) > N - N^{\frac{2}{3}}$$

Khi đó N là tích của 2 số nguyên tố phân biệt.

Phân tích thành số nguyên lớn

Chứng minh:

Thật vậy, rõ ràng N không phải là số nguyên tố vì nếu N là số nguyên tố thì $\varphi(N) = N - 1$, trái với giả thiết.

Do giả thiết N không phải là bình phương của một số nguyên tố. Như vậy nếu N không phải là tích của 2 số nguyên tố phân biệt thì nó phải là tích của nhiều hơn 2 số nguyên tố (không cần phân biệt).

Giả sử p là số nguyên tố nhỏ nhất của tích. Khi đó $p \leq N^{\frac{1}{3}}$ do đó chúng ta có:

$$\varphi(N) \leq N\left(1 - \frac{1}{p}\right) \leq N\left(1 - N^{-\frac{1}{3}}\right) = N - N^{\frac{2}{3}}$$

Điều này mâu thuẫn với giả thiết. Vậy N = p.q trong đó p, q là 2 số nguyên tố lẻ, phân biệt.

Chú ý: Mệnh đề đảo của mệnh đề 1 cũng đúng.

Phân tích thành số nguyên lớn

Mệnh đề 2:

Với (N,e) là khóa công khai của RSA. Cho trước khóa riêng d, người ta có thể phân tích thành nhân tử modul N = pq một cách hiệu quả. Ngược lại cho các thừa số của N, người ta có thể khôi phục được d một cách có hiệu quả.

Từ các mệnh đề ở trên người ta đã đưa ra một số tấn công vào RSA.

Các cách tấn công cơ bản

Modul chung

- Để tránh việc phân tích modul N = pq khác nhau cho các người dùng khác nhau, chúng ta lấy N chung cho tất cả. Cùng một N được sử dụng cho tất cả người sử dụng. Trung tâm xác thực có thể cung cấp cho người sử dụng i với một cặp e_i, d_i và người sử dụng i có một khóa công khai là (N, e_i) và khóa riêng là (N, d_i).

- Ban đầu có một bản mã C = M^{ea}mod N cung cấp cho Alice không được mã hóa bởi Bob, vì Bob không có d_a. Nhưng điều này là nhầm lẫn, từ mệnh đề 1 Bob có thể sử dụng các thành phần e_b và d_b của mình để nhân tử hóa N. N bị nhân tử hóa bởi Bob có thể lấy được khóa riêng d_a của Alice từ khóa công khai e_a của cô ấy. Với cách tiếp cận này, theo Simmons chỉ ra rằng một modul RSA không nên sử dụng quá một thực thể.

Các cách tấn công cơ bản

Mù (Blinding)

- Marvin chọn ngẫu nhiên một số r ∈ Z_N^* và đặt $M' = r^e M \ mod \ N$
- Marvin nhờ Bob ký lên M'. Bob có thể cung cấp chứ ký của mình là S' lên M'.
- Từ cách tính $S' = (M')^d \ mod \ N$, Marvin có thể đơn giản tính $S = \frac{S'}{r} \ mod \ N$ để có được chữ ký của Bob là S trên M.

$$S^e = \frac{(S')^e}{r^e} = \frac{(M')^d}{r^e} = \frac{M'}{r^e} = \text{M/(mod N)}$$

Số mũ riêng bé (Low Private Exponment)

Định lí 1 (M.Wiener, 1990):

Cho N = pq với q< p< 2p. Giả sử d < $\frac{1}{3}N^{1/4}$**.** Cho trước (N,e) với ed = 1 mod $\varphi(N)$**),** Marvin có thể tìm được d hiệu quả

Đô lớn e:

Thay vì rút gọn e trong $\varphi(N)$, ta sử dụng (N, e') cho khóa công khai thỏa mãn e' = e + t.$\varphi(N)$, t là một số rất lớn.

$\Rightarrow$ Có thể sử dụng e' thay thế e để mã hóa thông điệp.

Tuy nhiên: Khi e có giá trị lớn, theo chứng minh ở trên thì số k không thể nhỏ hơn.

Một tính toán đơn giản chỉ ra rằng nếu e' > $N^{1.5}$ thì không có vấn đề gì xảy ra mặc dù số d nhỏ và tấn công ở trên không thể thực hiện được. Nhưng số e lớn sẽ tăng thời gian mã hóa

Số mũ riêng bé (Low Private Exponment)

Sử dụng CRT:

Sử dụng định lí đồng dư trung hoa (CRT): Chọn một số d sao cho cả $d_p = d \bmod (p-1)$ và $d_q = d \bmod (q-1)$ đều nhỏ hơn 128 bits.

Để giải mã nhanh bản C ta tiến hành:

+ Tính $M_p = C^{dp} \bmod p$ và $M_q = C^{dq} \bmod q$.
+ Sử dụng CRT để tính giá trị $M \in Z_N$ thỏa mãn $M_p = C^{dp} \bmod p$ và $M_q = C^{dp} \bmod q$

=>Nếu (N,e) được biết thì kẻ địch có thể tấn công N trong thời gian $O(\min(\sqrt{dp}, \sqrt{dq}))$

Chú ý: Định lí 1 đã được cải thiện bởi Boneh và Durfee, họ chỉ ra rằng với số $d < N^{0.292}$ kẻ tấn công có thể tính được d từ (N,e). Kết quả này cho thấy ranh giới của Wiener là không rõ ràng. Nó có vẻ $d < N^{0.5}$, đây là một bài toán mở?

Bài toán mở: *Cho N = pq và* $d < N^{0.5}$**.** *Cho trước (N,e) với ed = 1 mod* $\varphi(N)$**,** *Marvin có thể tìm được d không?*

Số mũ công khai bé (Low Public Exponment)

Đinh lí 2 (Donald Coppersmith công bố năm 1997):

Cho N là một số nguyên và f ∈ $Z[x]$ là một đa thức mà có độ đo là d. Đặt X = $N^{\frac{1}{d}-e}$ cho e ≥ 0. Sau đó biết (N,f) Marvin có thể tìm tất cả các số nguyên $|x_0| < X$ thỏa mãn f(x_0) = 0 mod N. Thời gian chạy phụ thuộc vào thời gian chạy thuật toán LLL với trên một lưới có khoảng cách là O(ω) $với\ \omega = min(1/e, log_2 N)$.

Hastad's Broadcast Attack:

Xem e_i là thành phần công khai bằng 3. Marvin tìm ra M rất đơn giản nếu k ≥ 3. Thực vậy, Marvin có được C_1, C_2, C_3 $thỏa$ mãn:

$$C_1 = M^3 mod\ N_1, \qquad C_2 = M^3 mod\ N_2, \qquad C_1 = M^3 mod\ N_3$$

Nên với e = 3, gửi các thông điệp giống nhau đến 3 người nhận là không an toàn. Giải pháp chống tấn công này chúng ta gắn các thông điệp trước khi mã hóa với đa thức?

Số mũ công khai bé (Low Public Exponment)

Định lí 3 (Hastad):

Cho $N_1, \dots, N_k$ là những số nguyên tố và tập $N_{min} = \min_i(N_i)$ từng đôi một. Với $g_i \in Z_{Ni}[x]$, k là đa thức có giá trị nhỏ nhất là d. Tồn tại M < N_{min} thỏa mãn: $g_i(M) = m \bmod N_i$ với tất cả i = 1,...,k. Giả thiết rằng k > d, có thể tìm M khi cho $(N_i, g_i)^k = 1$.

Định lí chỉ ra rằng một hệ thống đồng biến các đa thức nguyên tố hỗn hợp có thể giải quyết hiệu quả, giải thuyết rằng các hàm được cung cấp đầy đủ. Bằng cách cài đặt $g_i = f_i^{ei} - C_i \bmod N_i$, Marvin có thể tìm được M từ bản mã được cho với số thành viên ít nhất là d, khi đó d là giá trị lớn nhất của $e_i \deg(f_i)$ với i = 1,...,k.

=> Để chống lại tấn công broadcast ở trên chúng ta sử dụng một cặp số ngẫu nhiên thay vì gắn cứng vào một giá trị.

Số mũ công khai bé (Low Public Exponment)

Hê quả (FR):

Giả sử rằng với e = 3 và (N, e) là một khóa công khai của RSA. Cho $M_1 \neq M_2 \in Z_N^*$ *thỏa mãn* $M_1 = f(M_2) \ mod$ *N trong đó f = ax + b* $\in Z_N^*$ *là đa thức tuyến tính với b ≠ 0. Khi đó cho trước* (N, e, C_1, C_2, f)*, Marvin có thể tìm được* M_1, M_2*với thời gian là đa thức bậc hai log N*

Chứng minh:

- Với e = 3 thì giá trị gcd phải là giá trị tuyến tính.
- Thật vậy, đa thức x3 – C_2 phân tích thành p và q là phép phân tích tuyên tính và không thể rút gọn về nhân tố bậc hai (ta nhớ rằng gcd(e, ϕ (N)) = 1 và vì thế x3 – C_2 chỉ có giá trị gốc nằm trong Z_N). Khi đó g_2 không thể chia cho g_1, gcd phải là một hàm tuyến tính.

  Với e = 3 hàm gcd luôn là tuyến tính. Tuy nhiên, đối với một vài M_1, M_2 và f, gcd có thể không phải là tuyến tính, trong trường hợp này việc tấn công là thất bại.

Thành phần công khai bé

Định lí 4:

Với (N,e) là một khóa công khai của RSA, N có độ dài n-bits. Đặt m =[n/e^2]. Với M∈ Z_N^* là một thông điệp có độ dài n-m bits. $M_1 = 2^m M + r_1$ và $M_2 = 2^m M + r_2$ với điều kiện r_1 và r_2 là hai số nguyên khác nhau thỏa mãn $0 \leq r_1, r_2 < 2^m$. Nếu Marvin biết (N,e) và các bản mã hóa C_1, C_2 của M_1, M_2(nhưng không biết r_1, r_2), anh ấy có thể tìm ra M một cách có hiệu quả.

Ý tưởng:

- Ta thêm ngẫu nhiên các bít vào cuối của thông điệp, thuật toán này có thể thu được bản rõ của M. Tấn công này rất đơn giản nhưng rất nguy hiểm
- Thực tế: Khi e = 3 tấn công có thể đạt được với độ dài của các bít thêm vào là ít hơn $1/9^{th}$ độ dài của bản thông điệp. Với e = 65537 thì sự tấn công là vô ích với các modulo kích cỡ chuẩn.

Thành phần công khai bé

Tấn công bằng khóa riêng:

Với (N,d) là một khóa riêng của RSA. Giả sử rằng Marvon có thể tìm được một nhân tử trong dãy bit của d, hay một phần tử của d, hay một phần của d. Từ đó Marvin có thể khôi phục được phần còn lại của d:

Định lí 5 (BDF):

Cho (N,d) là một khóa riêng của RSA trong đó N có độ dài n bits. Biết [n/4] bits ít ý nghĩa của d, Marvin có khôi phục được d với thời gian tuyến tính $elog_2\, e$.

Định lí 6 (Coppersmith):

Giả sử số N = pq (là một modulo RSA) có n bits. Cho trước n/4 bits ít ý nghĩa nhất (hoặc n/4 bits nhiều ý nghĩa nhất) của p (gải thiết p < q). Khi đó có tồn tại một phân tích số N một cách hiệu quả.

Cài đặt tấn công

Tấn công dựa trên thời gian

- Đây là một phương pháp phá mã dựa vào một "hiệu ứng lề" sinh ra bởi quá trình giải mã RSA. Hiệu ứng lề đó là thời gian thực hiện giải mã.
- Tấn công thông minh của Kocher cho thấy rằng bằng phương pháp lựa chọn thời gian chính xác để giải mã (hoặc ký số) RSA của smartcard, Marvin có thể nhanh chóng tìm ra thành phần giải mã riêng d.
- Bài toán:

  Giả sử người phá mã có thể đo được thời gian giải mã $M = C^d \bmod n$ dùng thuật toán bình phương liên tiếp.

  + Nếu một bít của *d* là 1 thì xảy ra hai phép mô-đun
  + Nếu bít đó là 0 thì chỉ có một phép mô-đun

  Thời gian thực hiện giải mã là khác nhau. Bằng một số phép thử bản rõ chọn trước, người phá mã có thể biết được các bít của *d* là 0 hay 1 và từ đó biết được *d*.

Tấn công dựa trên thời gian

- Sử dụng thuật toán "repeated squaring algorithm" – bình phương liên tiếp tính $C = M^d \bmod N$
- Với $d = d_n d_{n-1} \ldots d_0$ là biểu diễn nhị phân của d (hay $d = \sum_{i=0}^{n} i d_i$ với $d_i \in \{0, 1\}$),

  sử dụng nhiều nhất 2n modul nhân.

  Nó dựa trên việc xét $C = \prod_{i=0}^{n} M^{2^i d_i} \bmod N$
- Thuật toán như sau:

  Đặt z = M và C = 1

  Vòng lặp for i = 0,..., n thực hiện các bước:

  + Nếu $d_i = 1$, đặt C = C - z mod N

  + Đặt $z = z^2 \bmod N$

  Tại trạng thái kết thúc, C có giá trị là $M^d \bmod N$

Tấn công dựa các lỗi ngẫu nhiên

- Quá trình cài đặt giải mã RSA thường sử dụng định lý đồng dư Trung Quốc nhằm cải thiện tốc độ tính toán M^d mod N.

- Boneh, DeMillo, và Lipton đã quan sát và thấy rằng có một lỗi nguy hiểm khi sử dụng phương pháp CRT. Vấn đề là khi sinh mã mà máy tính hoạt động không đều là nguyên nhân gây nên lỗi tính toán.

  Hay nói cách khác trong khi copy giữa các thanh ghi, một bit của dòng bít bị thất lạc. (Sự hoạt động không đều nguyên nhân có thể do xung đột điện từ hoặc cũng có thể do sâu phần cứng, các lỗi này đã sớm được tìm thầy trong các phiên bản của chíp Pentium).

  Được cung cấp một mã lỗi, kẻ tấn công có thể dễ dàng phân tích thành nhân tử modul N

Một số tấn công nhân tử hóa số N với N lớn

Tìm nhân tử lớn thứ nhất $\leq \sqrt{N}$

- **Định lý FERMAT:**

  Giả sử n là một số nguyên dương lẻ có dạng n =p.q trong đó p ≤ q và p, q là các số nguyên tố. Khi đó biểu thức n có thể viết dưới dạng: $n = t^2 - s^2$ (t, s là các số nguyên dương). Các số nguyên t, s, p và q có mối quan hệ: $t = \frac{p+q}{2}$ và $s = \frac{q-p}{2}$

Tìm nhân tử lớn thứ nhất $\leq \sqrt{N}$

Phương pháp này được xây dựng dựa vào định lý Fermat, cụ thể như sau:

- Bước 1: Khởi tạo x = $2[\sqrt{N}] + 1$, y = 1, r = $[\sqrt{N}]^2 - n$
- Bước 2: Nếu r ≤ 0 thì chuyển đến bước 4.
- Bước 3: r = r – y; y = y – 2 chuyển đến bước 2
- Bước 4: Nếu r = 0 thì thuật toán dừng

  Khi đó ta có:

  n = $[\frac{x-y}{2}][\frac{x+y-2}{2}]$ (đây là hai nhân tử của n(p,q))

  $\frac{x-y}{2}$ là phân số có giá trị lớn nhất $\leq \sqrt{N}$
- Bước 5: r = r + x

  x = x + 2

  Chuyển đến bước 3.

Phân tích thứ hai

- Độ an toàn của RSA phụ thuộc vào độ khó của phép phân tích n thành các thừa số nguyên tố p, q. Nếu trong hai số p, q; số này nhỏ hơn số kia rất nhiều thì khả năng phân tích được n là rất cao.
- Khi thiết kế, nên chọn giá trị p, q sao cho p < $\sqrt{N}$ < q và độ dài của p xấp xỉ bằng một nửa độ dài n.

  Xác suất p nằm trong khoảng ($2\sqrt[3]{n} + 1$, $\sqrt{n}$) là rất cao.
- Bài toán đặt ra là cho n số nguyên dương lẻ, d ≥ $2\sqrt[3]{n} + 1$

  Tìm nhân tử lẻ nhỏ nhất f sao cho d < f ≤ $\sqrt{n}$

Phân tích thứ hai

Thuật toán được thực hiện như sau:

- Bước 1: Đặt r = n mod d (trong đó d = $\sqrt[3]{n}$ + 1)

  r' = n mod (d-2)

  q = 4[|$\frac{n}{d-2}$|-|$\frac{n}{d}$|]

  s = [$\sqrt{n}$]
- Bước 2: Nếu đặt d > s thuật toán kết thúc với kết quả không tìm được nhân tử
- Bước 3: Đặt d = d+2; x = r; r = 2r – r'; r' = x
- Bước 4: Nếu r <0, gán r = r + d; q = q – 1

  Chuyển đến bước 6
- Bước 5: Nếu r < d, gán r = r – d; q = q -1, chuyển đến bước 5
- Bước 6: Nếu r = 0 thì d là một thừa số của n, thuật toán kết thúc ta sẽ thu được f = d

  ngược lại ta chuyển đến bước 2.

Thuật toán Pollard's (p-1) (1974)

Giả sử N là β mịn (tất cả các ước nguyên tố của nó đều $\leq \beta$)

Thuật toán được thực hiện:

- Bước 1: a=2

  kt = false

- Bước 2: for j = 2 to β do

  a = a^j mod n

  d = gcd(a-1, n)

  if 1 < d< n then

  kt = true

  break

- Bước 3: If kt = true then d là một thừa số của N

  else không tìm thấy thừa số của N

* Độ phức tạp thuật toán $O(\beta \frac{\ln n}{\ln \beta})$

Ứng dụng của RSA trong các hoạt động bảo mật và công nghệ thông tin

Ứng dụng của RSA trong bảo mật dữ liệu

Chứng thực khi truy nhập

- Khi bạn truy cập vào tài khoản cá nhân và được hệ thống yêu cầu đưa ra con số đã được gửi về địa chỉ email, số điện thoại để xác thực đăng nhập.

Truyền tải dữ liệu an toàn

- Rất nhiều mạng xã hội đã bị chỉ trích khi liên tuc có những động thái nghe lén, theo dõi hoạt động và dữ liệu của người dùng. Cũng như các trang web cũng không tránh khỏi việc lưu lại những hoạt động, hành vi truy cập của người dùng đưa ra những kết luận phục vụ công cuộc quảng cáo.

Chữ ký số/ chữ ký điện tử

- Trên thẻ ATM luôn có chữ ký điện tử được mã hóa và giao dịch khi đăng ký tạo tài khoản ngân hàng

Ứng dụng của RSA trong bảo mật dữ liệu

- Hiện nay, các trung tâm dữ liệu của Zoho được lưu trữ tại các địa điểm an toàn và được theo dõi bằng camera quan sát trong đêm, 24/7 suốt cả năm. Máy chủ và máy khách của Zoho WorkDrive được kết nối qua SSL và bảo mật bằng khóa mã hóa 2048 bit dựa trên RSA.

  - Phục hồi sau sự cố
  - Bảo vệ chống xâm nhập
  - Xác thực hai yếu tố
  - Mã hóa trong quá trình truyền và trên thiết bị lưu trữ.

Ứng dụng của RSA trong công nghệ thông tin

- Trong ngôn ngữ lập trình Java, những đoạn code RSA được sử dụng để tăng tính bảo mật cho website, ứng dụng và đảm bảo tính an toàn truy cập cho người sử dụng

Tổng kết

KẾT LUẬN

Như vậy cho đến nay người ta mới công bố được một số phương pháp tấn công vào hệ thống mật mã RSA. Trong đó, phương pháp phân tích nhân tử modul N của RSA được nhiều nhà toán học tập trung nghiên cứu hơn cả. Tuy nhiên thuật toán bình phương đã và đang được chú ý hơn, mặc dù thuật toán cũng mới chỉ giải quyết cho trường hợp modul N có độ dài không lớn lắm. Nếu độ dài modul N của RSA mà lớn hơn thì cho đến nay chưa có một thuật toán nào khả thi được công bố.

Danh Mục Tài Liệu Tham Khảo

1. Slide môn Mật mã và Độ phức tạp của thuật toán của thầy Nguyễn Đình Hân
2. Cryptography_Theory_and_Practice_4th Ed 2019
3. https://anninhmang.edu.vn/mat-rsa/
4. Cryptography_and_network_security_principals_practice_7th Ed 2017