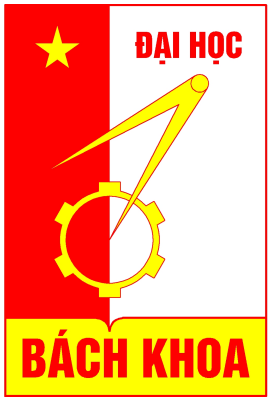
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

_____*******_____



ĐỒ ÁN TỐT NGHIỆP
NGÀNH KHOA HỌC MÁY TÍNH

Cài đặt bảo mật cho các thiết bị IoT

Sinh viên thực hiện:
Đặng Quang Trung 20134145 CNTT2.03-K58

Giảng viên:
TS. Trần Vĩnh Đức

Hà Nội 21-04-2018

Lời cảm ơn

Trước khi trình bày nội dung của đồ án này, tôi xin được gửi lời cảm ơn sâu sắc và chân thành nhất đến TS. Trần Vĩnh Đức, người đã tận tình hướng đẫn tôi trong suốt quá trình thực hiện đồ án này cũng như những năm tháng học tại trường Đại học Bách Khoa Hà Nội. Đồng thời tôi cũng xin bày tỏ lòng biết ơn đến các thầy cô trường Đại học Bách Khoa Hà Nội, Viện công nghệ thông tin và truyền thông, đặc biệt là các thầy cô bộ môn Khoa học máy tính đã tận tình chỉ dạy cho tôi trong những năm tháng học tập ở trường.

Đồng thời tôi xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn ở bên tôi, đông viên và giúp đỡ tôi trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp.

Mục lục

Bảng chữ viết tắt

Danh sách hình vẽ

Giới thiệu chung

Mật mã học là một lĩnh vực liên quan với các kỹ thuật ngôn ngữ và toán học để đảm bảo an toàn thông tin, cụ thể là trong thông tin liên lạc. Về phương diện lịch sử, mật mã học gắn liền với quá trình mã hóa, điều này có nghĩa là nó gắn với các cách thức để chuyển đổi thông tin từ dạng này sang dạng khác nhưng ở đây là từ dạng thông thường có thể nhận thức được thành dạng không thể nhận thức được, làm cho thông tin trở thành dạng không thể đọc được nếu như không có các kiến thức bí mật. Quá trình mã hóa được sử dụng chủ yếu để đảm bảo tính bí mật của các thông tin quan trọng, chẳng hạn trong công tác tình báo, quân sự hay ngoại giao cũng như các bí mật về kinh tế, thương mại. Trong những năm gần đây, lĩnh vực hoạt động của mật mã hóa đã được mở rộng: mật mã hóa hiện đại cung cấp cơ chế cho nhiều hoạt động hơn là chỉ duy nhất việc giữ bí mật và có một loạt các ứng dụng như: chứng thực khóa công khai, chữ ký số, bầu cử điện tử hay tiền điện tử. Ngoài ra, những người không có nhu cầu thiết yếu đặc biệt về tính bí mật cũng sử dụng các công nghệ mật mã hóa, thông thường được thiết kế và tạo lập sẵn trong các cơ sở hạ tầng của công nghệ tính toán và liên lạc viễn thông.

Mật mã học là một lĩnh vực liên ngành, được tạo ra từ một số lĩnh vực khác. Các dạng cổ nhất của mật mã hóa chủ yếu liên quan với các kiểu mẫu trong ngôn ngữ. Gần đây thì tầm quan trọng đã thay đổi và mật mã hóa sử dụng và gắn liền nhiều hơn với toán học, cụ thể là toán học rời rạc, bao gồm các vấn đề liên quan đến lý thuyết số, lý thuyết thông tin, độ phức tạp tính toán, thống kê và tổ hợp. Mật mã hóa cũng được coi là một nhánh của công nghệ, nhưng nó được coi là không bình thường vì nó liên quan đến các sự chống đối ngầm (xem công nghệ mật mã hóa và công nghệ an ninh). Mật mã hóa là công cụ được sử dụng trong an ninh máy tính và mạng.

Hiện nay, cùng với sự phát triển của tính toán khắp nơi, các hệ thống vạn vật kết nối (internet of things - IoT) ngày càng thu hút được sự quan tâm của các chuyên gia cũng như các nhà ứng dụng. Vấn đề an toàn thông tin trong hệ thống IoT với các thiết bị nhỏ gọn, năng lực tính toán thấp, trở thành một chủ đề nóng hiện nay. Với khả năng tính nhanh, an toàn chi phí thấp, mật mã tính toán trên đường công Elliptic tiêu biểu cho hệ mã trao đổi khóa của các tehiết bị IoT.

Nội dung chính của đồ án bao gồm 4 chương:

- **Chương 1:** Cơ sở lý thuyết
- **Chương 2:** Mật mã dựa trên đường cong Elliptic
- **Chương 3:** Chứng thư số ẩn dựa trên đường cong Elliptic
- **Chương 4:** Cài đặt cho thiết bị IoT

Chương 1

Cơ sở lý thuyết

Trong chương này sẽ trình bày cơ sở lý thuyết chung bao quát cho các hệ mật mã, nó sẽ cung cấp một số khái niệm và các kiến thức qúy giá, hỗ trợ đắc lực cho việc làm quen với lĩnh vực này.

1.1 Cơ sở mật mã

Mật mã là một lĩnh vực khoa học chuyên nghiên cứu về các phương pháp và kỹ thuật đảm bảo an toàn và bảo mật trong truyền tin liên lạc với giả thiết sự tồn tại của các thế lực thù địch, những kẻ muốn ăn cắp thông tin để lợi dụng và phá hoại. Tên gọi trong tiếng Anh, Cryptology đợưc dẫn giải nguồn gốc từ tiếng Hy lạp, trong đó kryptos nghĩa là "che dấu", logos nghĩa là "từ ngữ".

Các nhà nghiên cứu lĩnh vực này quan tâm xây dựng hoặc phân tích (để chỉ ra điểm yếu) các giao thức mật mã (cryptographic protocols), tức là các phương thức giao dịch có đảm bảo mục tiêu an toàn cho các bên tham gia (với giả thiết môi trường có kẻ đối địch, phá hoại).

Ngành Mật mã (cryptology) thường được quan niệm như sự kết hợp của 2 lĩnh vực con:

1. Sinh, chế mã mật (cryptography): nghiên cứu các kỹ thuật toán học nhằm cung cấp các công cụ hay dịch vụ đảm bảo an toàn thông tin.
2. Phá giải mã (cryptanalysis): nghiên cứu các kỹ thuật toán học phục vụ phân tích phá mật mã và/hoặc tạo ra các đoạn mã giản nhằm đánh lừa bên nhận tin

Hai lĩnh vực con này tồn tại như hai mặt đối lập, "đấu tranh để cùng phát triển" của một thể thống nhất là ngành khoa học mật mã (cryptology).

Mặc dù mật mã có thể coi là một ngành toán học phát triển cao, đòi hỏi tư duy cao để nắm được các thành tựu hiện đại của nó, nhưng cơ sở xuất phát ban đầu của nó lại là một mô hình thực tiễn khá đơn giản như sau:
hình vẽ

Như vậy trong một hệ thống mật mã khái quát sẽ có các thành phần sau:

- **Văn bản trơn**(plaintext X): tức là thông điệp nguyên gốc chưa được mã hóa.

- **Văn bản mã hóa**(ciphertext Y): tức là thông điệp đã được mã hóa.
- **Thuật toán mã hóa**(enciphering algorithm $E_z(X)$): là các giao thức hoặc hướng dẫn có tác dụng chuyển đổi văn bản trơn thành văn bản mã hóa. Đối với các hệ thống mật mã truyền thống, chỉ có người gửi thông điệp biết được thuật toán mã hóa, tuy nhiên đối với các hệ thống dùng mật mã hóa khóa công khai (Public key code - PKC), tất cả mọi người đều có thể biết thuật toán mã hóa mà không ảnh hưởng tiêu cực đến an ninh của hệ thống.
- **Khóa mã hóa**(enciphering key Z): là một hoặc nhiều đối tượng (thường là các con số hay là các hướng dẫn quan trọng nào đó) được dùng trong việc mã hóa văn bản trơn. Ngoại trừ trong hệ thống PKC, để đảm bảo bí mật an toàn thì khóa mã hóa thường chỉ được người gửi biết.
- **Thuật toán giải mã**(deciphering algorithm $D_z(Y)$): là các giao thức hoặc hướng dẫn có tác dụng chuyển đổi văn bản mã hóa trở về văn bản trơn. Để đảm bảo bí mật, chỉ có người nhận thông điệp biết được thuật toán giải mã.
- **Khóa giải mã**(deciphering key Z'): là một hoặc nhiều đối tượng (thường là các con số hay là các hướng dẫn quan trọng nào đó) được dùng trong việc giải mã văn bản bị mã hóa. Để đảm bảo bí mật, chỉ có người nhận thông điệp biết được khóa giải mã.
- **Sản phẩm mật mã**(Cryptography Product): bao gồm các hệ thống thiết bị, module, mạch tích hợp và các chương trình phần mềm mã hoá chuyên dụng có tích hợp các thuật toán mật mã, được thiết kế, chế tạo để bảo vệ thông tin giao dịch điện tử và lưu trữ dưới dạng số hoá, trong đó sử dụng "Thuật toán mã đối xứng" hoặc "Thuật toán mã không đối xứng".

1.2 Mật mã khóa đối xứng

Trong mật mã học, các thuật toán khóa đối xứng (symmetric-key algorithms) là một lớp các thuật toán mật mã hóa trong đó các khóa dùng cho việc mật mã hóa và giải mã có quan hệ rõ ràng với nhau (có thể dễ dàng tìm được một khóa nếu biết khóa kia). Mã khóa loại này không công khai.

Khóa dùng để mã hóa có liên hệ một cách rõ ràng với khóa dùng để giải mã có nghĩa chúng có thể hoàn toàn giống nhau, hoặc chỉ khác nhau nhờ một biến đổi đơn giản giữa hai khóa. Trên thực tế, các khóa này đại diện cho một bí mật được phân hưởng bởi hai bên hoặc nhiều hơn và được sử dụng để giữ gìn sự bí mật trong kênh truyền thông tin.

Thuật toán đối xứng có thể được chia ra làm hai thể loại, mật mã dòng (stream ciphers) và mật mã khối (block ciphers). Mật mã dòng mã hóa từng bit của thông điệp trong khi mật mã khối gộp một số bit lại và mật mã hóa chúng như một đơn vị. Cỡ khối được dùng thường là các khối 64 bit(128 bit). Ngày nay được ưa chuộng sử dụng hơn là mật mã khối (block cipher) – trong đó từng khối nhiều ký tự được mã hóa cùng một lúc. Trong mật mã khối, các tham số quan trọng là kích thước (độ dài khối) và kích thước khóa. Các khái niệm này được minh họa qua ví dụ sau đây.

key	000	001	010	011	100	101	110	111
0	001	111	110	000	100	010	101	011
1	001	110	111	100	011	010	000	101
2	001	000	100	101	110	111	010	011
3	100	101	110	111	000	001	010	011
4	101	110	100	010	011	001	011	111

Qua ví dụ đơn giản này (chỉ có tính chất minh họa), ta thấy rằng nếu các tham số kích thước khối và khóa qua nhỏ thì mật mã rất dễ bị phá bằng các tấn công thông qua phân tích thống kê. Chẳng hạn trong ví dụ trên, nếu kẻ thù nhận được một khối mã ciphertext 001 thì nó có thể dễ dàng suy ra plaintext tương ứng chỉ có thể là 000 hoặc 101 (nhờ thống kê trên bảng biến đổi mã).

Vì vây, các điều kiện cần cho mật mã khối an toàn là:

- Kích thước khối phải đủ lớn để chống lại các loại tấn công phá hoại bằng phương pháp thống kê. Tuy nhiên cần lưu ý rằng kích thước khối lớn sẽ làm thời gian trễ lớn.
- Không gian khóa phải đủ lớn (tức là chiều dài khóa phải đủ lớn) để chống lại tìm kiếm vét cạn.Tuy nhiên mặt khác, khóa cần phải đủ ngắn để việc làm khóa, phân phối và lưu trữ được hiệu quả.

Về các nguyên lý thiết kế mật mã khối, người ta đã ghi nhận 2 nguyên tắc cơ sở sau để có bảo mật cao, đó là việc tạo ra confusion (tính hỗn loạn, rắc rối) và diffusion (tính khuếch tán).

Confusion. (Hỗn loạn, rắc rối) Sự phụ thuộc của bản mã đối với bản rõ phải thực phức tạp để gây rắc rối, cảm giác hỗn loạn đối với kẻ thù có ý định phân tích tìm qui luật để phá mã. Quan hệ hàm số của mã-tin là phi tuyến (non-linear).

Diffusion. (Khuếch tán) Làm khuếch tán những mẫu văn bản mang đặc tính thống kê (gây ra do dư thừa của ngôn ngữ) lẫn vào toàn bộ văn bản. Nhờ đó tạo ra khó khăn cho kẻ thù trong việc dò phá mã trên cơ sở thống kê các mẫu lặp lại cao. Sự thay đổi của một bit trong một khối bản rõ phải dẫn tới sự thay đối hoàn toàn trong khối mã tạo ra.

Một cách đơn giản nhất, confusion có thể được thực hiện bằng phép thay thế (substitution) trong khi diffusion được tạo ra bằng các phép chuyển đổi chỗ (transposition/permutation) hay hoán vị. Toàn bộ sơ đồ biến đổi mật mã sẽ là một lưới các biến đổi thay thế-hoán vị (substitution-permutation network).

1.3 Mật mã khóa công khai

Giới thiệu

Mật mã hóa khóa công khai là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật).

Trong hệ mã khóa công khai, mỗi người sử dụng có hai khoá, một được gọi là khoá bí mật (secret key hay private key) và một được gọi là khoá công khai (public key). Khoá

thứ nhất chỉ mình user biết và giữ bí mật, còn khoá thứ hai thì anh ta có thể tự do phổ biến công khai. Khoá thứ nhất thường đi liền với thuật toán giải mã, còn khoá thứ hai thường đi liền với thuật toán sinh mã, tuy nhiên điều đó không phải là bắt buộc. Ta hãy ký hiệu chúng là z (khóa riêng) và Z (khóa công khai)

Hoạt động của chúng là đối xứng

$$X = D(z, E(Z, X)) \tag{1.1}$$
$$X = E(Z, D(z, X)) \tag{1.2}$$

Trong đó hệ thức (1.1) biểu tượng cho bài toán truyền tin mật: bất kỳ người sử dụng nào khác như B,C,D ... muốn gửi tin cho A chỉ việc mã hoá thông tin với khoá công khai (Z_A) của A rồi gửi đi. Chỉ có A mới có thể khoá riêng để giải mã (z_A) và đọc được tin, kẻ nghe trộm Eve không thể giải mã để lấy được tin vì không có khoá z_A.

Còn hệ thức (1.2) sẽ được sử dụng để xây dựng các hệ chữ ký điện tử trong đó thao tác Ký chính là thực hiện $E(Z_A)$ còn kiểm định chữ ký là thông qua gọi $D(z_A)$.

Hệ mật mã theo nguyên tắc nói trên được gọi là hệ mã với khoá công khai (public key cryptosystems) hay còn được gọi là mã khóa phi đối xứng (asymmetric key cryptosystems). Ta sẽ viết tắt hệ thống kiểu này bằng PKC.

Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích:

- **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 có phải đã được tạo với một khóa bí mật nào đó hay không.
- **Thỏa thuận khóa**: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.

Thông thường, các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

Hệ thống khóa công khai

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). Nó 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. Chẳng hạn, việc phân tích một hợp số là tích của 2 số nguyên tố lớn hàng trăm chữ số sẽ mất hàng ngàn năm tính toán với một máy PC trung bình có CPU khoảng trên 2Ghz.

Thuật toán RSA

Xây dựng: Chọn các tham số

1. *Chọn hai số nguyên tố lớn p và q. Tính* $n = p \times q$ *và* $m = \phi(n) = (p-1) \times (q-1)$.
2. *Chọn e,* $1 \leq e \leq m-1$*, sao cho* $gcd(e, m) = 1$.
3. *Tìm d sao cho* $e * d = 1$ *(mod m), tức là tính* $d = e^{-1}$ *(mod m), giải theo thuật toán gcd mở rộng.*

Khóa công khai (Public key) là (e, n)

Khoá dùng riêng (Private key) là d, p, q)

Giả sử X là một khối tin gốc (plaintext), Y là một khối mã tương ứng của X, và (z_A, Z_A) là các thành phần công khai và riêng của khoá của Alice

Sinh Mã. Nếu Bob muốn gửi một thông báo mã hoá cho Alice thì anh ta chỉ việc dùng khoá công khai của Alice để thực hiện:

$$Y = E_{Z_A}(X) = X^e \pm n$$

Giải mã. Khi Alice muốn giải mã Y, cô ta chỉ việc dùng khoá riêng $z_A = d$ để thực hiện như sau:

$$D_{z_A}(Y) = Y^d \pm n$$

Hệ Rabin

Hệ Rabin cũng xây dựng trên việc lấy $n = p \times q$ làm bí mật. N được coi là khoá công khai (PK) còn (p,q) là khoá bí mật (SK).

Mã hoá là việc thực hiện:

$$Y = X^2 (mod\ n)$$

còn giải mã là việc tính căn bậc hai:

$$X = \sqrt{Y} (mod\ n) \tag{1.3}$$

Có thể thấy, nếu biết $n = p \times q$ thì dễ dàng tìm được nghiệm cho phương trình này, còn nếu không thì việc tìm nghiệm là khó tương đương với bài toán PTTSNT số n.

Khi biết $N = p \times q$ thì 1.3 được giải ra có bốn nghiệm do đó để xác định được đâu là bản rõ gốc phải có mẹo để chọn được đúng giá trị cần thiết trong số 4 nghiệm đó

Hệ Rabin có một số ưu điểm so với RSA:

- Tính an toàn được chứng minh hoàn toàn tương đương với bài toán PTTSNT, nói cách khác tính ATBM của Rabin là có thể chứng minh được (provable)
- Ngoại trừ trường hợp RSA hoạt động với e nhỏ còn thuật toán sinh mã của Rabin nhanh hơn nhiều so với RSA là hệ đòi hỏi phải tính luỹ thừa. Thời gian giải mã thì tương đương nhau.

Nhược điểm: Vì phương trình giải mã cho 4 nghiệm nên làm khó dễ việc giải mã. Thông thường, bản rõ trước khi được mã hoá cần được nối thêm vào đuôi một chuỗi số xác định để làm dấu vết nhận dạng (chẳng hạn nối thêm 20 số 0 – như vậy trong số 4 nghiệm giải ra, chuỗi nào tận cùng bằng 20 con 0 thì đúng là bản rõ cần nhận). Vì lý do này nên Rabin thường được dùng chủ yếu cho chứng thực (chữ ký điện tử).

Hệ El-Gamal

Tạo khóa:

Alice chọn một số nguyên tố p và hai số nguyên ngẫu nhiên g và u, cả hai đều nhỏ hơn p. Sau đó tính

$$y = g^u (mod\ p)$$

Bây giờ khóa công khai của Alice được lấy là (p,g,y), khoá mật là u.

Sinh mã:

1. Nếu Bob muốn mã hoá một tin X để truyền cho Alice thì trước hết anh ta chọn một số ngẫu nhiên k sao cho $(k, p-1) = 1$
2. Tính
$$a = g^k (mod\ p) b = y^k X (mod\ p)$$

Mã là $Y = (a, b)$ và có độ dài gấp đôi bản rõ.

Giải mã: Alice nhận được $Y = (a, b)$ và giải ra X theo công thức sau:

$$X = \frac{b}{a^u} (mod\ p)$$

1.4 Chữ ký điện tử và Hàm băm

Chữ ký điện tử

Khái niệm chữ ký điện tử được hai nhà bác học Diffie và Hellman đề xuất trong cùng bài báo nổi tiếng của các ông khai sáng nguyên lý của hệ thống mật mã công khai (1976). Ý tưởng về mô phỏng chữ ký tay trên văn bản trong đời thường đã có từ lâu, nhưng thực sự chỉ có thể thực hiện được cùng với sự ra đời của hệ mật mã KCK (khóa công khai). Như đã biết, hệ thống mật mã đối xứng đã được sử dụng phổ biến trước đó không có tính chất đại diện duy nhất cho một cá nhân. Trong khi đó, một hệ mã hóa khóa công khai (hay còn gọi là phi đối xứng) có thể được xem là được tạo lập để giúp bảo mật truyền tin trong liên lạc giữa 1 cá nhân và phần còn lại của xã hội. Nhờ có mật mã KCK, khái niệm chữ ký điện tử mới được hiện thực hóa và giúp cho giao dịch kinh tế thương mại trong đời sống có thể đi vào số hóa hoàn toàn, qua đó thúc đẩy hoạt động dịch vụ trực tuyến trên Internet phát triển như ngày này.

Chữ ký điện tử hay chữ ký và chữ ký tay thực ra không phải hoàn toàn tương tự. Chữ ký tay là dấu vết của con người tác động lên cùng bản giấy đã mang chứa văn bản (in/viết sẵn). Phần chữ ký tay và phần văn bản có sẵn là độc lập, không có quan hệ ràng buộc nào. Do các qui luật của thế giới vật lý, người ta không thể đánh tráo chữ ký theo kiểu đơn giản là xé bỏ phần tờ giấy chứa chữ ký và ghép nối vào một phần giấy mang chữ ký tạo mới khác. Tuy nhiên trong thế giới số hóa, các qui luật vật lý này không có mặt, và bất cứ lập trình viên nào cũng có thể tha hồ cắt ghép văn bản số hóa mà không bị phát hiện.

Sơ đồ chữ ký cơ bản

Do đó, nguyên lý tạo chữ ký điện tử là khác hẳn và phức tạp hơn. Khi có một văn bản ở dạng nhị phân X, người ta phải tạo ra một chữ ký ở dạng nhị phân S sao cho S phụ thuộc hàm vào X, tức là $S = f(X)$, hơn nữa quan hệ hàm này là bí mật (có tham số khóa bí mật) đối với người ngoài. Do đó nếu có kẻ nào thử đánh tráo (tức giả mạo) chữ ký, quan hệ hàm $S = f(X)$ sẽ không còn đúng và bị phát hiện.

Tuy nhiên việc phát hiện xem một văn bản có chữ ký có là chuẩn hay bị giả mạo lại phải là một thao tác mà ai cũng làm được dễ dàng, không cần đến khóa bí mật kia (do người chủ chữ ký nắm giữ). Vì vậy hệ thống chữ ký điện tử được xây dựng trên nguyên tắc sử dụng hai thuật toán riêng rẽ cho việc tạo chữ ký và kiểm định chữ ký, thông qua việc sử dụng cặp 2 hàm toán học đối lập nhau, một cần khóa bí mật còn một thì không. Chính do điều này, mật mã khóa công khai đã được khai thác để giúp hiện thực điểm chốt của cơ chế đặc biệt này.

Giả sử Alice đã thiết lập một hệ mật mã KCK với thành phần khóa bí mật z_A và công khai Z_A, tức là có hàm sinh mã $E_{Z_A}()$ và hàm giải mã $D_{z_A}()$, khi đó Alice có thể tạo chữ ký điện tử bằng hàm $D_{z_A}()$ và bất kỳ người nào khác sẽ kiểm tra bằng hàm $E_{Z_A}()$. Cụ thể là, với văn bản nhị phân X, Alice sẽ tạo được chữ ký $S = D_{z_A}(X)$, văn bản có chữ ký sẽ là $Y = X||S$. Khi văn bản này đến tay Bob, Bob sẽ kiểm tra tính hợp lệ bằng việc tính $X' = E_{Z_A}(S)$ và đối chiếu $X = X'$? Lưu ký, Bob sẽ cần kiếm được khóa công khai của Alice, Z_A, bằng một cách nào đó.

Như vậy nếu Bob đã nhận được văn bản có chữ ký $X||S$ và dùng khóa công khai của Alice để kiểm định thành công, văn bản đó trở thành bằng chứng, ngay cả khi Alice có muốn chối cãi đã tạo ra và ký nó cũng không được. Bởi vì chỉ duy nhất Alice mới sở hữu khóa d_A bí mật để tạo ra được chữ ký hợp lệ mà thôi. Ta gọi tính chất này của chữ ký điện tử là tính không thể chối cãi được (non-repudiation). Ngay cả khi Alice có khiếu nại bị oan với lý do chữ ký tạo ra bởi một kẻ đã ăn cắp được khóa bí mật của cô ta, thì điều này cũng không thể chứng minh được.

Công chứng: Để có thể đảm bảo phòng tránh được tình trạng chữ ký giả mạo do kẻ gian ăn cắp được khóa bí mật của người bị hại, người ta đã giới thiệu thêm hệ thống công chứng – public notary. Ý tưởng thực hiện: có thêm một bên thứ ba tham gia, vô tư và có thẩm quyền hợp pháp, được gọi là công chứng viên (public notary), sẽ được thuê để ký xác nhận thêm vào sau chữ ký của Alice đối với những văn bản quan trọng mà Alice ký. Văn bản đầy đủ chữ ký sẽ có dạng $Y = X||S_A||S_N$ trong đó chữ ký của công chứng viên S_N là ký trên văn bản $X||S_A$.

Bằng chứng biên nhận: Trong truyền tin liên lạc, chữ ký điện tử có thể sử dụng để đảm bảo tính chính xác của tài liệu (bằng chữ ký của bên gửi A), và bên nhận B có thể gửi lại chữ ký của mình vào tài liệu đã nhận như là bằng chứng để A biết là B đã thực sự nhận được tài liệu đó. Nếu thủ tục này được thực hiện, sau này A có thể chứng minh được là mình đã gửi tài liệu cho B, ngay cả khi lúc đó B muốn chối cũng không được.

$A \to B : Y = E_{Z_B}(X|||D_{z_A}(X))$

B : tính $E_{z_B}(Y)$ thu được X và $S = D_{z_A}(X)$, kiểm tra xem $X =? E_{z_A}(S)$

$B \to A : Y' = E_{Z_A}(D_{z_B}(X))$

A : tính $S_B(X) = D_{z_A}(Y')$, đó chính là chữ ký của B trên X, bằng chứng xác nhận B đã nhận được tài liệu X chính xác.

Nhược điểm của hệ chữ ký cơ sở

Hệ chữ ký điện tử theo tiếp cận ban đầu nói trên, tức là sử dụng D_z để ký và E_Z để kiểm định, là khá đơn giản và phạm phải nhược điểm lớn:

- Chữ ký quá dài, dài đúng bằng tài liệu: Với văn bản dài, ta cần dùng việc chia khối rồi ký lên nhiều khối; cụ thể là $X = X_1||X_2||X_3||\dots||X_t \to S = S_A(X_1)||S_A(X_2)||S_A(X_3)||\dots||S_A(X_t))$. Rõ ràng số lượng khối trên văn bản đã ký nhiều gấp đôi ban đầu.
- Không những dài, việc thực hiện nhiều lần thuật toán KCK (ký lên từng khối) sẽ làm thủ tục ký có thể diễn ra rất lâu, thời gian tỷ lệ với độ dài văn bản. Điều này là không chấp nhận được với các giao dịch trực tuyến.
- Kẻ tấn công có thể dễ dàng phá hệ thống chữ ký này bằng kiểu tấn công lắp ghép khối (thay đổi thứ tự, thêm hay bớt khối ...).

Vì vậy hệ thống chữ ký điện tử đơn giản kiểu này đã không được sử dụng. Giải pháp đầy đủ là có thêm sự hộ trợ của hàm băm, tức là "Băm" tài liệu trước khi ký, sẽ được trình bày tiếp theo đây.

Hàm băm và ứng dụng chữ ký điện tử

Một hàm băm H sẽ lấy ở đầu vào là một thông tin X có kích thước bất kỳ và sinh kết quả ra là một chuỗi $h_X = h(X)$ có độ dài cố định, thường là nhỏ hơn nhiều so với kích thước của X. Chuỗi này thường được gọi là cốt yếu, hay cốt (digest) của thông tin X.

Ví dụ: Thông tin X có thể là một tệp độ dài hàng trăm Kb trong khi cốt của nó chỉ là một khối có độ dài 128bit. Tất nhiên, điều đó dẫn đến khả năng có thể có 2 thông tin $X \neq X'$ mà cho cùng một cốt giống nhau với một hàm băm, tức là $H(X) = H(X')$. Trường hợp này gọi là đụng độ (collision).

Tuy nhiên với hàm băm thiết kế tốt, đụng độ là gần như không thể xảy ra được trên thực tế. Nói cách khác nếu cố đi tìm, khối lượng tính toán phải thực hiện là rất lớn, không khả thi với công cụ tính toán hiện thời.

Hàm băm có ứng dụng chủ chốt trong các hệ chữ ký điện tử được sử dụng hiên nay. Thay vì ký (tức là thực hiện thuật toán D_{z_A}) lên văn bản X, Alice cần thực hiện việc ký lên h_X, như vậy văn bản đã ký sẽ có dạng $X||D_{z_A}(H(X))$.

Để đảm bảm an toàn cao, chống được tấn công giả mạo chữ ký, chúng ta cần sử dụng các hàm băm mật mã (cryptographic hash function) với các thuộc tính như sau:

1. Lấy đầu vào là một xâu với độ dài bất kỳ và sinh ra một xâu với độ dài cố định.
2. Có tính một chiều: biết X, có thể dễ dàng tính được giá trị băm h_X, nhưng không thể tính ngược được X khi chỉ biết h_X, với công cụ tính toán hiện nay (bất khả thi về tính toán).
3. Có tính phi đụng độ cao (collision free), tức là thực tế không thể tim được hai thông tin $X \neq X'$ sao cho $H(X) = H(X')$. Tất nhiên, đây là bất khả thi về mặt tính toán.

hình

Đụng độ

Rõ ràng là với không gian giá trị băm nhỏ hơn không gian tin về mặt kích thước thì chắc chắn sẽ tồn tại đụng độ (collision), nghĩa là có hai bản rõ $X \neq X'$ mà giá trị băm của chúng giống nhau nghĩa là $h_X = h_{X'}$. Điều này có thể thấy rõ ràng qua nguyên lý Diricle - Nếu có n+1 con thỏ được thả vào n cái chuồng thì phải tồn tại ít nhất một cái chuồng mà trong đó có ít ra là hai con thỏ ở chung.

Trong thực tế người ta thường chọn không gian băm cỡ khoảng 64bit, 128 bit ....Trong khi đó các văn bản thực tế lớn hơn nhiều, cỡ Kb trở lên, cho nên việc tồn tại đụng độ là chắc chắn. Tuy nhiên nếu sử dụng hàm băm mật mã có không gian băm lớn được chế tạo tốt (an toàn) thì việc tìm ra đụng độ đòi hỏi khối lượng tính toán lớn đến mức phi thực tế (infesible computation).

Việc chế tạo các hàm băm phi đụng độ là rất khó. Nhiều hàm băm được phát minh bởi các nhóm có tên tuổi trên thế giới sau một thời gian xuất hiện đã bị những người khác chỉ ra những đụng độ tồn tại và không được công nhận là an toàn nữa.

Các kỹ thuật làm hàm băm

Các kỹ thuật để chế tạo được hàm băm có thể chia ra làm ba loại:

- Dựa trên việc áp dụng các hệ mã khối theo mật mã khoá bí mật đối xứng (SKC).
- Dựa trên các phép toán số học đồng dư.
- Các hàm thiết kế băm đặc biệt.

Các hàm băm chế từ hệ SKC

Sơ đồ Rabin-Matyas-Davies-Price (RMDP)

$$\begin{aligned} &X = X_1 X_2 \dots \\ &H_0 = 0 \text{ (hay một số ngẫu nhiên nào đó)} \\ &H_i = E_{x_i}(H_{i-1}) \end{aligned}$$

Ở đây, tất nhiên các TIN phải được chặt thành các khối có kích cỡ bằng khoá của hệ mã E. Giá trị băm là $H(X) = (H_0, H_t)$.

Người ta chứng minh được rằng với không gian băm chỉ là 64bit thì $H(X)$ không phải là một chiều, tức là cho $Y = H(X)$, việc tìm ngược được X là khả thi.

Sơ đồ Davies-Meyer (DM hash)

$$\begin{aligned} &X = X_1 X_2 \dots \\ &H_0 = \text{vector khởi tạo là một số ngẫu nhiên nào đó} \\ &H_i = E_{x_i}(H_{i-1}) \oplus H_{i-1} \end{aligned}$$

- Việc xây dựng các hàm băm từ các mã khối đòi hỏi phải có phân tích tính an toàn một cách cẩn thận.

- DM được coi như là an toàn nếu sử dụng với các mã khối kích thước 128bit.
- Không có hệ nào khác đã được đề xuất mà được chứng minh là an toàn.

Các hàm băm dựa trên các phép toán số học đồng dư

QCMDC (Quadratic Congruential Manipulation Detection Code)

Được đề xuất bởi Jueneman (1983).

Bản rõ được chia thành các khối m bit. H_0 là giá trị khởi đầu được chọn ngẫu nhiên và giữ bí mật (vì thế vẫn được gọi là hàm băm có khóa - keyed hash function).

Các bước xây dựng hàm băm như sau:

M là một số nguyên tố sao cho $M \geq 2^{m-1}$,
$H_i = (H_i - 1 + X_i)^2 \pmod{\text{M}}$
M là luỹ thừa của 2.

Hệ này đã bị phá (Coppersmith).

Davies-Price (1985)

Chia văn bản thành các khối có m-d bit:

$X = X_1 X_2 X_3 \ldots X_n$
$H_i = (H_{i-1} \oplus X_i)^2 \pmod{\text{M}}$, $H_0 = 0$

Hệ này bị chứng minh là không đảm bảo tính một chiều (Girault)

Các hàm băm được chế tạo đặc biệt

Ngoài các kỹ thuật thông thường nói trên người ta đã tìm nhiều cách rất riêng biệt khác nhau để chế tạo ra những hàm băm có độ tin cậy cao. Thông thường những sơ đồ này rất phức tạp và có những cấu trúc đặc biệt, nên không trình bày đầy đủ ở đây. Sau đây là một số các hàm băm nổi tiếng.

MD5 (Rivest 1992)

Đây là một trong các hàm băm có tiếng nhất và được sử dụng thông dụng:

+ Nó lấy vào các khối đầu vào 512 bit và sinh ra các giá trị băm 128 bit.
+ Được tin là phi đụng độ và một chiều (one - way)
+ Thuật toán MD5 được thiết kế cho phép chạy tốt nhất trên các máy tính 32 bit. Nó sử dụng các phép toán đơn giản như phép cộng modulo 32, do đó thích hợp với việc mã hoá cho các bộ xử lý 32 bit.

SHA (Secure Hash Function)

Đây là một thuật toán được đề xuất và bảo trợ bởi cơ quan NIST để sử dụng đối với hệ chữ ký DSA (cũng là một dự chuẩn cho chữ ký điện tử). Nó cho giá trị băm là 160 bit và được thiết kế với cùng một tiếp cận như MD5.

HAVAL

Một hệ băm của Australia cho phép thay đổi kích thước giá trị băm. Cấu trúc rất giống như MD5.

Snefru Mekle (1989)

+ Là hàm băm có khóa (keyed hash function)
+ Cho phép 1 trong 2 lựa chọn kích thước giá trị băm là 128 bit và 256 bit.
+ Eli Biham đã chỉ ra một đụng độ cho trường hợp 128 bit.

1.5 Hệ thống chứng thực và Chứng thư số

Hệ thống chứng thực

Hệ thống chứng thực là một hạ tầng an ninh mạng được xây dựng trên một hạ tầng cơ sở khóa công khai (PKI) cung cấp các giải pháp đảm bảo an toàn cho các hoạt động (gọi chung là giao dịch) thông qua mạng.

Tại sao lại phải sử dụng hệ thống chứng thực?

- Hệ thống chứng thực cung cấp các dịch vụ đảm bảo an toàn cho các giao dịch thông qua mạng. Các dịch vụ cơ bản mà một hệ thống chứng thực cung cấp bao gồm:
- Dịch vụ xác thực: nhằm xác định xem ai đang giao dịch với mình.
- Dịch vụ bảo mật: đảm bảo tính bí mật của thông tin, người không có thẩm quyền không thể đọc được nội dung của thông tin.
- Dịch vụ toàn vẹn: khẳng định thông tin có bị thay đổi hay không.
- Dịch vụ chống chối bỏ: cung cấp các bằng chứng chống lại việc chối bỏ một hành động đã thực hiện hay đã diễn ra
- Như vậy sử dụng hệ thống chứng thực sẽ đảm bảo, bí mật, toàn vẹn cho thông tin được truyền qua mạng, xác thực được người dùng và chống chối bỏ các hành động hay sự kiện đã xảy ra.

Hệ thống chứng thực gồm những thành phần nào?

- Hệ thống chứng thực gồm 2 thành phần:
  - Thành phần thực hiện các nhiệm vụ về quản lý chứng thư số như: đăng ký và phát hành, thu hồi ... chứng thư số.
  - Thành phần thực hiện chức năng xác định xem một chứng thư số có hợp lệ hay không

Cơ quan chứng thực (CA) là gì?

Cơ quan chứng thực (Certification Authority - CA) có thẩm quyền cấp phát, thu hồi, quản lý chứng thư số cho các thực thể thực hiện các giao dịch an toàn. Cơ quan chứng thực là một thành phần chính của hệ thống chứng thực.

Cơ quan đăng ký (RA) là gì?

Cơ quan đăng ký (Registration Authority) là một thành phần trong hệ thống chứng thực có nhiệm vụ tiếp nhận và xác minh các yêu cầu về chứng thư số của người sử dụng đồng thời gửi các yêu cầu đã xác minh cho cơ quan chứng thực (CA) thực hiện yêu cầu đó.

Hệ thống chứng thực có những ứng dụng gì?

- Một số ứng dụng của hệ thống chứng thực:
- Nhóm các dịch vụ chính phủ điện tử e-Government:
  - Hóa đơn điện tử (E-Invoice)
  - Thuế điện tử (E-Tax Filing)
  - Hải quan điện tử (E-Customs)
  - Bầu cử điện tử (E-Voting)
  - E-Passport
  - PKI-based National ID Card
  - Các dịch vụ của chính phủ cho doanh nghiệp G2B (các ứng dụng đăng ký kê khai, thăm dò qua mạng đối với các doanh nghiệp)
  - Các dịch vụ của chính phủ cho công dân G2C (dịch vụ y tế ...).
- Nhóm các dịch vụ ngân hàng trực tuyến (Online Banking)
  - Thanh toán trực tuyến (E-Payment)
  - Tiền điện tử (E-Billing)
  - Kinh doanh chứng khoán trực tuyến (Online security trading)
  - Đấu thầu trực tuyến (E-Procurement)
  - Bảo hiểm trực tuyến (E-Insurance)
  - Quản lý tài liệu
  - Bảo mật email

Chứng thư số

Để thực hiện được các giao dịch an toàn qua mạng, các bên tham gia cần phải có "chứng thư số". Chứng thư số là một cấu trúc dữ liệu chứa các thông tin cần thiết để thực hiện các giao dịch an toàn qua mạng. Chứng thư số được lưu giữ trên máy tính dưới dạng một tập tin (file).

Nội dung chứng thư số bao gồm:

- Tên chủ thể chứng thư số.
- Khoá công khai.
- Một số thông tin khác như, tên của CA cấp chứng chỉ số đó, hạn dùng, thuật toán ký ....
- Chữ ký số của CA cấp chứng thư số đó.

Mục đích của chứng thư số dùng để nhận diện một đối tượng khi tham gia giao dịch trên mạng.

Ứng dụng chứng thư số để làm gì?

- Với chứng thư số người dùng có thể:
  - Xác định danh tính người dùng khi đăng nhập vào một hệ thống (xác thực).
  - Ký số các tài liệu Word, PDF hay một tệp liệu.
  - Mã hóa thông tin để đảm bảo bí mật khi gửi và nhận trên mạng.
  - Thực hiện các kênh liên lạc trao đổi thông tin bí mật với các thực thể trên mạng như thực hiện kênh liên lạc mật giữa người dùng với webserver.

Chương 2

Mật mã dựa trên đường cong Elliptic

2.1 Đường cong Elliptic

2.2 Mật mã dựa trên đường cong Elliptic

Chương 3

Chứng thư số ẩn dựa trên đường cong Elliptic

Chương 4

Cài đặt cho thiết bị IoT

Kết luận và hướng phát triển

Tài liệu tham khảo

[1] Sách Giáo trình Cơ sở An toàn Thông tin thầy Nguyễn Văn Khanh - Đại học Bách Khoa.

[2] Slide Trí tuệ nhân tạo thầy Nguyễn Nhật Quang. Chương 5 : Thỏa mãn ràng buộc

[3] Stuart Russell and Peter Norvig *Artificial Intelligence: A Modern Approach* 2nd edition, Prentice Hall, page 137, 2003.

[4] Ha Quang Minh and Deville Yves and Pham Quang Dung and Ha Minh Hoang, *On the min cost traveling salesman problem with drone*, arXiv preprint arXiv:1509.08764, 2015

[5] S. Banker, Amazon and drones – here is why it will work (dec 2013). URL http://www.forbes.com/sites/stevebanker/2013/12/19/amazon-drones-here-is-why-it-will-work/

[6] Nguyễn Đức Nghĩa, Nguyễn Tô Thành *Toán rời rạc* 3rd edition, Nhà xuất bản đại học quốc gia Hà Nội, page 107-108, 2006.

[7] Slide Tối ưu hóa tổ hợp thầy Nguyễn Đức Nghĩa. Chương mở đầu, Bài toán vận tải.

[8] Slide Tìm kiếm cục bộ dựa trên ràng buộc thầy Phạm Quang Dũng. Chương 5: Contraint-base local search applications.

[9] Slide Phân tích và thiết kế thuật toán thầy Nguyễn Đức Nghĩa. Chương 5: Quy hoạch động, Chương 3: Greedy Algorithms.

[10] Francesca Rossi, Peter van Beek, Toby Walsh *Handbook of Constraint Programming* 1st edition, Elsevier Science, page 107-108, 2006.

[11] Bài giảng môn tính toán tiến hóa cô Huỳnh Thị Thanh Bình. Thuật giải di truyền.

[12] Rupert Howell,Jonathon Wong *Apache OFBiz Development: The Beginner's Tutorial*, Paperback , page 53, 2008. truyền.

[13] Ruth Hoffman *Apache OFBiz Cookbook*, Paperback , page 28, 2010.

[14] Apache Ofbiz (wikipedia) https://en.wikipedia.org/wiki/Apache_OFBiz