|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  **BÁO CÁO BÀI TẬP LỚN**  ***Chủ đề:***  ***HMAC-Based One-Time Password***   |  |  |  | | --- | --- | --- | | Giảng viên hướng dẫn: | Hán Trọng Thanh |  | | Nhóm sinh viên: | Nguyễn Văn Chiến | 20172431 | |  | Nguyễn Quốc Anh | 20172397 | |  | Đỗ Quang Minh | 20172687 | |  | Chu Nguyên Hoàng | 20172564 |   Hà Nội, 5-2019 |

**LỜI NÓI ĐẦU**

Hiện nay khoa học kĩ thuật trên thế giới đang ngày càng phát triển mạnh mẽ. Khoa học kĩ thuật đóng vai trò không thể thiếu trong đời sống hiện đại ngày nay của chúng ta nó hiện diện trong từng sản phẩm, từng vật dụng mà chúng ta sử dụng hàng ngày. Tốc độ phát triển của khoa học kĩ thuật ngày nay có thể tính theo từng ngày từng giờ trên toàn cầu và Việt Nam chúng ta cần phải rất nỗ lực để có thể bắt kịp tốc độ phát triển đó. Vì vậy việc học tập và đạo tạo, kế thừa và phát triển những thành tựu khoa học cho thế hệ mới ngày nay là vô cùng quan trọng đặc biệt là thế hệ sinh viên. Việc tiếp cận đó cần phải tiếp cận cả về mặt lí thuyết lẫn thực hành, như vậy việc học tập và đào tạo mới có thể đạt được hiệu quả tốt nhất, đáp ứng được cho xã hội nguồn nhân lực khoa học kĩ thuật chất lượng cao. Nhờ đó chúng em nhận thức được việc cần phải học tập trau dồi không chỉ là kiến thức lý thuyết mà còn phải áp dụng những kiến thức đó vào thực hành qua các dự án, các bài đọc chuyên đề nâng cao của các môn học. Trong quá trình học tập học phần Lý thuyết mật mã do thầy Hán Trọng Thanh giảng dạy em đã được hướng dẫn giao đề tài bài tập lớn để thực hành tiếp cận thực tế mật mã OTP. Trong quá trình tìm hiểu chúng em gặp rất nhiều khó khăn, chúng em xin cảm ơn sự giúp đỡ, giải đáp của thầy và các bạn trong lớp, nhờ đó chúng em đã hoàn thành bài tập của mình.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[DANH MỤC HÌNH VẼ i](#_Toc59907827)

[DANH MỤC BẢNG BIỂU ii](#_Toc59907828)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 1](#_Toc59907829)

[1.1 Giới thiệu về OTP 1](#_Toc59907830)

[1.2 Sinh OTP và phân phối OTP 3](#_Toc59907831)

[1.2.1 Phương pháp sinh OTP dựa trên việc đồng bộ thời gian 4](#_Toc59907832)

[1.2.2 Phương pháp dựa trên các thuật toán số học 4](#_Toc59907833)

[1.2.3 Lưu ý khi sử dụng OTP 5](#_Toc59907834)

[1.2.4 Các phương thức phân phối OTP 5](#_Toc59907835)

[CHƯƠNG 2. TRIỂN KHAI MÃ OTP 7](#_Toc59907836)

[2.1 Thuật toán HOTP 7](#_Toc59907837)

[KẾT LUẬN 13](#_Toc59907838)

[TÀI LIỆU THAM KHẢO 14](#_Toc59907839)

# 

# DANH MỤC HÌNH VẼ

[Hình 2.1 Thuật toán HOTP 8](#_Toc59907822)

# DANH MỤC BẢNG BIỂU

[Bảng 1‑1 Các phiên mật khẩu OTP 4](#_Toc59907810)

# 

# GIỚI THIỆU CHUNG

Chương này sẽ giới thiệu tổng quan về mã OTP (One Time Password) gồm khái niệm, ứng dụng và các phương pháp sinh mã OTP.

## Giới thiệu về OTP

OTP – One Time Password là loại mật khẩu chỉ dùng một lần và chỉ có giá trị cho một phiên đăng nhập hoặc cho một giao dịch trong một khoản thời gian nhất định. OTP thường được tạo ra dựa trên các thông tin chia sẻ trước giữa hai bên xác thực, hoặc các sự kiện diễn ra đồng thời ở cả hai bên.

Lợi ích của OTP là nó chống được tấn công phát lại, nghĩa là nếu có một ai đó có thể lấy được thông tin về OTP trong một phiên làm việc thì cũng không thể sử dụng nó để đăng nhập vào lần kế tiếp. OTP được các điệp viên CIA, KGB, MI6... dùng phổ biến trong hai cuộc thế chiến. Lý do thứ nhất là vì nó đơn giản: mã & giải mã chỉ cần dùng đến tính nhẩm (có thể dùng thêm bút chì và giấy), lý do thứ hai là nó rất an toàn. Tuy đã được dùng rất lâu từ trước nhưng mãi đến khoảng năm 1940, phương pháp này mới được chứng mình bằng lý thuyết về tính an toàn tuyệt đối của nó. Chứng minh được đưa ra đồng thời và độc lập bởi Claude Shannon (nhà toán học Mỹ, cha đẻ lý thuyết thông tin) và Vladimir Kotelnikov (viện sĩ khoa học Liên bang Nga, kỹ sư chế tạo rađa).

Có một cách sử dụng OTP đặc biệt gọi là chia sẻ bí mật (secret splitting), sau khi mã hoá, văn bản gốc bị hủy thay vì khóa, sau đó khóa và văn bản mã hoá được đưa cho hai người khác nhau cất giữ. Chỉ khi hai người này cũng đồng ý nối hai "khoá" lại với nhau thì mới giải mã ra được văn bản gốc. Tương tự, có thể chia xẻ bí mật cho 3, 4,... người bằng cách sử dụng 2, 3,... khoá. Đây là cách bảo vệ các tài nguyên đặc biệt quan trọng, trách nhiệm bảo vệ đó được chia xẻ cho nhiều người, tuy nhiên lưu ý rằng nếu chỉ một phần của bí mật bị mất đi, thì bí mật đó cũng sẽ mất đi vĩnh viễn.

OTP là phương pháp mã hoá tuyệt đối an toàn nếu được sử dụng đúng cách, và là phương pháp tuyệt đối an toàn duy nhất cho đến thời điểm hiện tại. Văn bản được mã hoá với OTP không cho biết bất kỳ thông tin gì về văn bản gốc, ngoại trừ độ dài. Với một văn bản đã mã hoá cho trước, chúng ta có thể nghĩ ra các chuỗi khoá để "giải mã" nó về bất kỳ văn bản nào chúng ta muốn! Các phương pháp mã hoá mới sau này như DES (Data Encryption Standard), AES (Advanced Encryption Standard), PGP (Pretty Good Privacy), PKI (Public Key Infastructure)... tuy tiện dụng và có nhiều ưu điểm khác, nhưng về mặt lý thuyết không phải là không phá được. Nhưng trong sử dụng thực tế, có những lý do sau khiến OTP trở nên không an toàn:

* Chuỗi khóa OTP không thực sự ngẫu nhiên (các nhân viên thư ký của KGB tạo ra OTP bằng cách gõ ngẫu nhiên lên máy đánh chữ, nhưng xu hướng gõ phím của tay người vẫn có những pattern nhất định).
* Việc cất giữ và tiêu huỹ OTP có quá nhiều yếu tố rủi ro (đã có tình huống CIA giải được mã nhờ một cuốn sổ OTP đã bị đốt nhưng chưa cháy hết).
* Mỗi trang OTP chỉ được dùng một lần (đã có lúc trong tình hình khẩn cấp, nhân viên KGB bất cẩn dùng một trang OTP cho nhiều lần mã hoá, dẫn đến việc CIA giải được khoảng 1% trong số những thông điệp gửi bởi KGB trong những năm 1945 ~ 1950).

Điểm yếu nhất của OTP nằm trong quá trình trao đổi khoá (key exchange), đó là một trong những lý do hình thành phương pháp public key rất tiện dụng sau này.

Lỗ hổng bảo mật OTP:

* Cách thứ nhất hacker lừa nạn nhân nhập mã SMS OTP vào một website giả mạo để lấy OTP
* Cách thứ hai lừa nạn nhân cài phần mềm gián điệp chiếm quyền điều khiển của thiết bị di động.
* Cách thứ ba, tấn công vào các máy trung gian trong quá trình truyền nhận

Khắc phục:

* Yêu cầu người dung không truy cập vào các website lạ
* Không tải những ứng dụng lạ, chưa xác minh
* Ngân hang trực tiếp gửi, truyền tin cho máy của người dungf không cần qua bên nhà mạng trung gian
* Sử dụng hard token

Đến bây giờ, khi những phương tiện mã hoá và truyền thông đã quá hiện đại, người ta vẫn còn tiếp tục dùng OTP cho những kênh thông tin thuộc loại top secret (như đường dây hotline Washington DC - Moscow, liên lạc với tàu ngầm...) vì tính tuyệt đối an toàn đã được chứng minh lý thuyết của nó.

## Sinh OTP và phân phối OTP

Thuật toán sinh OTP thường sử dụng các phép toán liên quan tới cách sinh số ngẫu nhiên. Điều này là cần thiết vì nếu không ta có thể dễ dàng dự đoán được OTP sinh cho phiên kế tiếp từ việc quan sát những OTP của những phiên làm việc trước.

Vì pass được thay đổi liên tục nên hacker sẽ khó lấy được pass hơn rất nhiều. Có rất nhiều kỹ thuật được sử dụng trong OTP, tuy nhiên thông dụng nhất là 3 loại sau :

Loại thứ nhất : sử dụng các hàm toán học để tạo pass mới dựa trên pass cũ

Loại thứ hai : sử dụng các hàm toán học để tạo pass mới nhưng không dựa trên pass cũ mà dựa trên các thông tin về sự giao dịch của phiên cũ hoặc tạo ra một số ngẫu nhiên.

Loại thứ ba : sử dụng tín hiệu đồng bộ thời gian giữa Server và Client để tạo ra pass mới.

Thiết bị sinh password một lần - OTP (One Time Password) là loại khóa được dùng phổ biến nhất hiện nay vì rẻ và dễ dùng. Như tên gọi, OTP chỉ có giá trị sử dụng một lần nên tính bảo mật cao: sau khi người dùng gõ vào và đăng nhập thành công thì password này hết hiệu lực (lần đăng nhập sau sẽ dùng password khác); tin tặc nếu có lấy trộm password này cũng không đăng nhập hệ thống được.

Các phương thức phân phối OTP tới người dùng cũng rất đa dạng và phong phú. Một số hệ thống có thể sử dụng một thiết bị etoken đặc biệt - đã được đồng bộ trước với server xác thực. Thiết bị etoken này sẽ sinh ra OTP có giá trị trùng khớp với giá trị OTP sinh ra ở phía server và hiển thị trên một màn hình nhỏ. Một số hệ thống khác thì sử dụng các phần mềm chạy trên điện thoại di động của người dùng để sinh OTP thay vì thiết bị etoken, ngoài ra hệ thống server xác thực có thể thực hiện việc gửi tin nhắn báo OTP tới số điện thoại di động đã đăng ký trước của người dùng. Một số hệ thống khác lại sử dụng những mật khảu OTP đã được in ra giấy, người sử dụng sẽ sử dụng những OTP này để thực hiện xác thực.

### Phương pháp sinh OTP dựa trên việc đồng bộ thời gian

Loại đồng bộ thời gian tạo ra mã số khó đoán (mật mã hay khóa) dựa vào đồng hồ trong và mã số này được xác thực với điều kiện đồng hồ trong của thiết bị OTP đồng bộ với máy chủ xác thực. Do sự xê dịch của đồng hồ, việc đồng bộ tuyệt đối chính xác là không thể nên máy chủ xác thực phải chấp nhận các khóa có sự sai lệch đôi chút. Điều quan trọng đó là thu hẹp hết mức “khung cửa” này để giảm thiểu khả năng bị tấn công. Đa phần các nhà cung cấp thiết bị OTP áp dụng phương thức cộng dồn thời gian xê dịch để điều chỉnh với mỗi xác thực thành công. Thiết bị OTP đồng bộ thời gian có thể phải cân chỉnh lại nếu không được sử dụng một thời gian dài.

### Phương pháp dựa trên các thuật toán số học

Phương pháp này sinh OTP trên cơ sở các OTP đã sử dụng trước đó của người dùng. Một ví dụ cho thuật toán này là thuật toán của Leslie Lamport – thuật toán sử dụng các hàm một chiều f. Hệ thống OTP làm việc dựa trên một giá trị mầm khởi tạo s để sinh mật khẩu lần đầu tiên. Mật khẩu OTP được sử dụng cho phiên làm việc đầu tiên sẽ được tính như sau:

Trong đó – với là số lần áp dụng hàm *f* lên giá trị mầm *s*. Khi đó ta sẽ có lần lượt các mật khẩu OTP cho các phiên như ở bảng dưới đây.

Bảng 1‑1 Các phiên mật khẩu OTP

|  |  |  |  |
| --- | --- | --- | --- |
| Phiên 1 | Phiên 2 | Phiên 3 | Phiên 4 |
| OTP1 = fN(s) | OTP2 = fN-1(s) | OTP3 = fN-2(s) | OTP4= fN-3(s) |

Nếu một kẻ tấn công nào đó có thể bắt được giá trị OTP của một phiên làm việc nào đó thì hắn có thể dùng để đăng nhập vào hệ thống khi mà giá trị OTP này còn hợp lệ, tuy nhiên khi OTP không còn hợp lệ nữa thì hắn không thể thực hiện việc đăng nhập nữa và để lấy được giá trị OTP cho lần đăng nhập kế tiếp thì hắn sẽ phải đối mặt với việc tính hàm f ngược. Do hàm f đã được chọn là hàm một chiều nên việc này là cực kì khó để thực hiện. Nếu hàm f là một hàm băm mật mã thì việc tính toán là cực kì khó khăn thậm chí là không thể.

SMS cũng có thể được dùng làm phương tiện sinh OTP như khóa hai. Có hai dạng khóa SMS:

*Với phương thức SMS tức thời*, trước tiên người dùng thực hiện đăng nhập với password như bình thường, nếu thành công máy chủ xác thực sẽ sinh OTP và gửi ngay SMS cho người dùng để hoàn tất khóa xác thực thứ hai. Giao thức SMS không bảo đảm gửi tin thành công hay gửi kịp thời. Điều này có thể làm giảm tính khả dụng của việc xác thực hai khóa dựa trên SMS.

*Với khóa SMS lô* thì khác, người dùng nhận một danh sách mã số trước khi đăng nhập. Mỗi mã số gắn với một ký tự hay số dòng. Người dùng cần nhập vào mã số tương ứng với dòng yêu cầu để làm khóa xác thực thứ hai. SMS lô có thể giảm đáng kể chi phí SMS và khắc phục được vấn đề chất lượng dịch vụ của giao thức SMS vì danh sách được gửi đến người dùng trước khi họ thực sự cần đến. Tuy nhiên khuyết điểm của SMS lô là khả năng bị tấn công lớn hơn.

### Lưu ý khi sử dụng OTP

Thời gian hợp lệ của OTP cũng cần phải được cân nhắc tính toán – tùy thuộc vào các yếu tố như: phương thức phân phối OTP, cách thức tạo OTP, tốc độ đường truyền… Thông thường trong các hệ thống thì OTP có thời gian hợp lệ là 5 phút.

### Các phương thức phân phối OTP

***Phân phối qua SMS:*** Đây là hình thức phân phối phổ OTP phổ biến nhất do kênh phân phối này gần như có ở khắp mọi nơi do mức độ phổ biến của điện thoại di động như hiện nay. Phân phối OTP qua SMS có lợi thế là tận dụng được cơ sở hạ tầng sẵn có của mạng điện thoại di động, mức độ tiện dụng… do đó sẽ giảm đáng kể chi phí để triển khai. Tuy nhiên có điều cần phải lưu ý là chi phí cho việc gửi tin nhắn SMS là khá cao – đặc biệt khi thực hiện roamming và SMS thường không được mã hóa hoặc mã hóa với thuật toán mã yếu ( A5/x).

***Phân phối qua thiết bị etoken:*** Hình thức phân phối này chủ yếu dành cho phương pháp sinh OTP dựa trên thời gian. Khách hàng sẽ được cung cấp một thiết bị etoken – thiết bị này cũng được cài đặt thuật toán sinh OTP như ở trên server và đã được đồng bộ hóa với server.

***Phân phối qua email***

# TRIỂN KHAI MÃ OTP

Chương này trình bày về cách triển khai mã OTP dựa trên sử dụng thuật toán HOTP (HMAC-Based One-Time Password) dựa trên HMAC (Hash-based Message Authentication Code)

## Thuật toán HOTP

HOTP là một thuật toán sinh mật khẩu OTP dựa trên hàm hash SHA-1. Cách sinh mật khẩu OTP sử dụng thuật toán là

Trong công thức (2.1):

* K: Là giá trị chia sẻ bí mật giữa Client và Server.
* C: Là bộ đếm đã được đồng bộ giữa Client và Server, C có độ dài 8 bytes.
* Truncate( ): Là hàm tách chuỗi, thực hiện việc trích xuất kết quả từ hàm Hash để có được mật khẩu OTP.

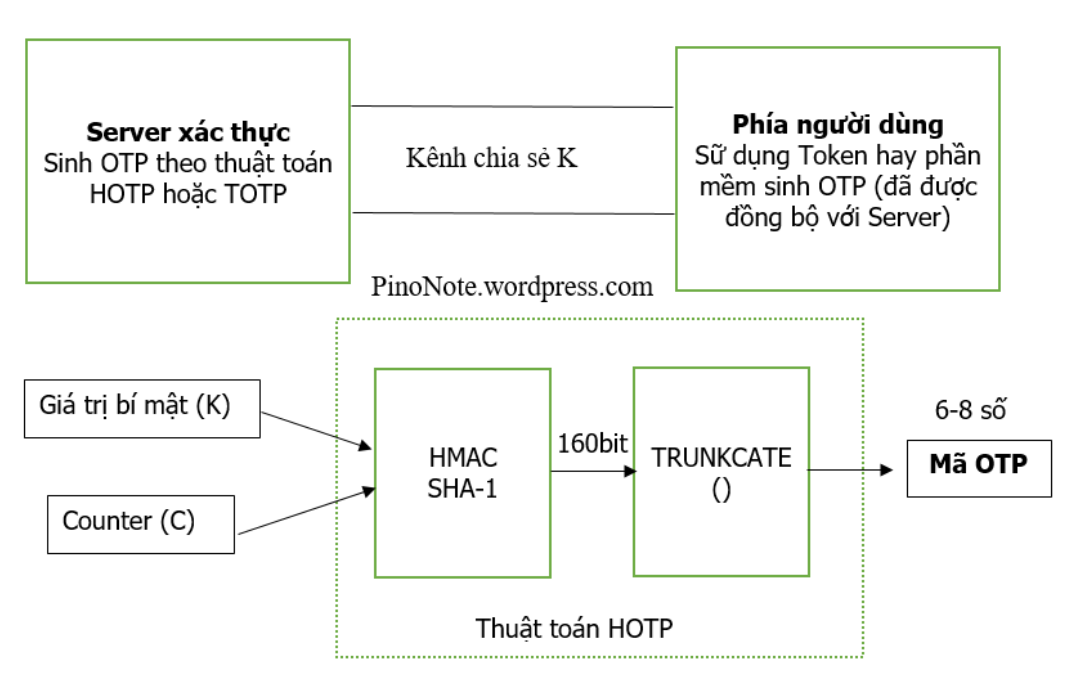
Phương pháp tách chuỗi như sau:

Trong công thức (2):

* C1 = 0x36 (36 trong hệ thập lục phân).
* C2 = 0x5c (5C trong hệ thập lục phân).
* HMAC\_SHA-1(K,C) là một hàm tính toán dựa trên thuật toán HMAC kết hợp với hàm băm SHA-1 của giá trị K và bộ đếm C.

Kết quả đầu ra của hàm *HMAC\_SHA-1(K,C)* cho ta một giá trị có độ dài là 160 bits = 20 bytes, chúng ta dùng hàm tách chuỗi (Truncate) để tách từ chuổi 160 bits thành một chuỗi mới có độ dài 32 bit, sau đó tính modulo để được mật khẩu OTP. Cụ thể như sau: từ kết quả đầu ra 160 bit của hàm HMAC\_SHA1(K,C), ta lấy 4 bit thấp của byte cuối cùng chuyển sang cơ số 10 để tìm vị trí offset, sau đó ta được chuỗi 4 bytes = 32 bit tính từ vị trí offset. Giá trị mật khẩu được tính theo công thức sau:

Trong công thức 3: d là số chữ số của OTP, thông thường một mật khẩu OTP sinh ra có độ dài từ 6 đến 8 chữ số (ví dụ là: 123456,…).



Hình 2.1 Thuật toán HOTP

Ở đây, giá trị bí mật K được chia sẻ giữa máy chủ xác thực và máy trạm phía người dùng qua một kênh liên lạc nào đó. Sau đó máy chủ xác thực sử dụng giá trị bí mật K, bộ đếm C để sinh mật khẩu OTP theo công thức (2.2), máy trạm phía người dùng cũng sử dụng thiết bị sinh mật khẩu OTP như Token để sinh mật khẩu OTP (Token lưu giá trị bí mật K được chia sẻ và bộ đếm C). Để xác thực, máy chủ sẽ so sánh mật khẩu OTP sinh ra trên máy chủ với mật khẩu OTP sinh ra từ máy trạm phía người dùng, nếu giống nhau, người dùng trên máy trạm xác thực thành công.

Trong nhiều trường hợp, vấn đề xác thực giữa máy chủ xác thực và máy trạm phía người dùng sẽ sảy ra quá trình mất đồng bộ giá trị mật khẩu vì vậy, muốn người dùng đăng nhập hệ thống thực thành công, chúng ta có thể phân tích quá trình đồng bộ giá trị sinh mật khẩu OTP giữa máy chủ xác thực và máy trạm phía người dùng đối với thuật toán sinh OTP dựa trên sự kết hợp của HMAC và hàm băm SHA-1 như sau:

Thông thường, trên máy chủ xác thực của bên cung cấp dịch vụ cho người dùng sẽ thực hiện sinh mật khẩu OTP. Trên máy trạm phía người dùng sẽ sử dụng ứng dụng sinh mật khẩu OTP, sau đó khi xác thực, máy chủ sẽ kiểm tra mật khẩu OTP sinh ra trên máy trạm từ phía người dùng có giống với mật khẩu OTP được sinh ra trên máy chủ hay không. Nếu mật khẩu OTP giống nhau, người dùng sẽ được xác thực và sử dụng dịch vụ của hệ thống. Trong một số trường hợp, vấn đề mất đồng bộ xảy ra khi người dùng sử dụng thiết bị sinh OTP như Token và ấn nút sinh OTP nhiều lần nhưng không sử dụng hoặc trong quá trình đăng nhập bị lỗi,… dẫn đến mật khẩu OTP sinh ra từ phía người dùng không giống với mật khẩu sinh ra từ phía máy chủ xác thực, khi đó người dùng xác thực không thành công. Để giải quyết được vấn đề này, quá trình đồng bộ giá trị sinh mật khẩu OTP trên máy chủ xác thực và máy trạm phía người dùng sẽ được thực hiện như sau:

Giả sử người dùng có một thiết bị phần cứng Token sinh mật khẩu OTP và mật khẩu OTP được bởi hàm: HMAC\_SHA1(Key, Counter\_Client). Máy chủ xác thực sinh mật khẩu OTP theo công thức: *HMAC\_SHA1(Key,Counter\_Server).* Người dùng muốn được xác thực và sử dụng dịch vụ của hệ thống thì phải đồng bộ Counter của máy trạm và máy chủ. Giả sử tại thời điểm hiện tại: Counter\_Client = Counter\_Server + a, với a<W (W là cửa sổ đồng bộ). Ban đầu, Client gửi OTP hiện thời tới Server. Server kiểm tra OTP vừa được gửi từ Client có là một giá trị trong tập các giá trị:

OTP=HMAC\_SHA1(Key,Counter\_Server+1)

OTP=HMAC\_SHA1(Key,Counter\_Server+2),…

OTP=HMAC\_SHA1(Key,Counter\_Server+W)

Trong đó sẽ có một giá trị OTP=HMAC\_SHA1(Key,Counter\_Server+a) giống với của OTP từ máy trạm đã gửi lên. Máy chủ xác thực đúng máy trạm, sau đó cập nhật giá trị cho Counter lúc này là Counter\_Server=Counter\_Server+a. Tại thời điểm này máy trạm và máy chủ có cùng giá trị Counter, Counter được đồng bộ để cho các phiên xác thực sau.

Đối với thuật toán HOTP đã trình bày ở phần trên, mật khẩu OTP được sinh ra dựa vào giá trị bí mật (K) được chia sẻ và một giá trị đếm (C). Còn đối với thuật toán TOTP được định nghĩa RFC 6238 là một thuật toán cũng tương tự như HOTP nhưng dựa trên thời gian T(Time) thay vì giá trị đếm C (Counter). Với thuật toán TOTP sinh mật khẩu OTP dựa theo thời gian thì giá trị T về thời gian được tính như sau:

Trong công thức (2.4):

* là giá trị thời gian hiện tại được tính theo thời gian Unix (được tính từ thời điểm của Unix Epoch là ngày 01/01/1970 theo UTC (giờ chuẩn quốc tế).
* : Là giá trị thời gian ban đầu (thường chọn T0 = 0).
* *X*: Là bước thời gian, đây là tham số quyết định thời gian hợp lệ của mật khẩu OTP.
* *T* chính là kết quả tính (đã lấy phần nguyên) từ công thức tính toán trên.

Với thuật toán TOTP thì thời gian chuẩn hợp lệ của mật khẩu OTP là 30 giây (X=30), thời gian có hiệu lực của mỗi lần sử dụng mật khẩu OTP là 30 giây được chọn phù hợp với yêu cầu về bảo mật và khả năng sử dụng. Thuật toán TOTP dựa trên thuật toán HOTP thay giá trị đếm (C) bằng giá trị thời gian (T):

Đối với thuật toán TOTP độ dài của mật khẩu OTP được tính như sau:

Trong công thức (2.6): d là số chữ số của mật khẩu OTP, thông thường một mật khẩu OTP sinh ra có độ dài từ 6 đến 8 chữ số (ví dụ như : 123456,…).

Tương tự như vấn đề đã trình bày trong phần thuật toán HOTP, vấn đề mất đồng bộ về thời gian hay giá trị đếm khi sinh mật khẩu OTP có thể xảy ra dẫn đến mật khẩu sinh OTP trên máy chủ xác thực và máy trạm phía người dùng không giống nhau, người dùng sẽ xác thực không thành công để tiếp tục sử dụng dịch vụ. Khi đó, vấn đề đồng bộ lại thời gian hay giá trị đếm được thực hiện, đối với thuật toán TOTP khi sinh mật khẩu OTP là vấn đề đồng bộ về thời gian giữa máy trạm phía người dùng và máy chủ xác thực. Theo công thức (2.5) tính toán giá trị thời gian *T* cần phải có giá trị thời gian hiện tại , nếu thời gian hiện tại không được đồng bộ thì người dùng trên máy trạm sẽ không xác thực hợp lệ được với máy chủ xác thực.

Giả sử thời gian hiện tại trong đồng hồ bên máy trạm khi thực hiện việc xác thực với máy chủ xác thực là *T1*, và thời gian phía máy chủ là *T’1*. Như vậy, thời gian trên máy chủ với máy trạm không trùng.

– Giả sử thời gian ban đầu được chọn là *T0 = 0* và bước thời gian *X=30* giây, khi đó giá trị T trên máy trạm được tính là:

– Client sẽ sinh mật khẩu OTP dựa theo công thức *HOTP(K,TClient).*

– Đối với máy chủ xác thực thì: và Server sẽ sinh mật khẩu một OTP dựa theo công thức *HOTP(K,TServer).*

Khi đó, mật khẩu OTP trên máy trạm dùng để xác thực với máy chủ sẽ không phù hợp và xác thực sẽ không thành công. Để việc xác thực được thành công, máy chủ sẽ thực hiện tính toán và thay đổi giá trị sao cho phù hợp.

– Client sinh mật khẩu OTP theo công thức: *HOTP(K,TClient)* không phù hợp với mật khẩu OTP sinh ra trên máy chủ: *HOTP(K,TServer)*. Máy chủ sẽ tính giá trị khác lần lượt là: *HOTP(K,TServer + a)* với a là giá trị thời gian tính bằng giây.

– Khi tính đến một giá trị *a* nào đó như *a = a1* và *OTP= HOTP(K,TSever + a1)* thấy giống với mật khẩu OTP được sinh ra trên máy trạm, máy chủ sẽ cho phép người dùng trên máy trạm xác thực.

Ví dụ về sinh mật khẩu OTP theo thuật toán TOTP:

Giả sử tính được HMAC\_SHA-1(Secret (K),Time (T)) (160bit) là:

1f |86 | 98 | 69| 0e | 02 | ca | 16 | 61 | 85 | 50 | ef | 7f | 19 | da | 8e | 94 | 5b | 55 | 5a

Sau đó, sử dụng hàm tách chuỗi *Truncate(HMAC\_SHA-1(K,C))* (32 bit). Cụ thể tách như sau:

– Lấy 4 bits thấp của byte cuối cùng, trong ví dụ trên thì byte cuối cùng là một giá trị hexa là 5a (01011010 theo cơ số 2), 4 bits thấp nhất của byte cuối cùng này là 1010 tương ứng là 10 (trong hệ cơ số 10).

– Lấy tiếp 4 bytes từ vị trí Offset 10 (đếm từ 0) ta sẽ có: 50 | ef | 7f | 19

– Giá trị có được là: 0x50ef7f19, đổi sang cơ số 10 ta được giá trị mới là: 1357872921. Đây chính là chuỗi vừa được hàm *truncate ()* tách ra từ mã xác thực thông điệp MAC.

Thông thường mật khẩu OTP hiện nay có độ dài từ 6 đến 8 chữ số, theo thuật toán sinh OTP dựa trên đồng bộ về thời gian (TOTP) ta có thể có được mật khẩu OTP mới có độ dài tùy chọn theo công thức (2.6) Dựa vào công thức (2.6) sinh được mật khẩu OTP có độ dài 6 chữ số tương ứng là: 872921.Đây chính là mật khẩu OTP được sinh ra và hiển thị trên ứng dụng của người dùng, mật khẩu OTP này sẽ thay đổi theo thời gian (mặc định là sau 30 giây sẽ thay đổi). Người dùng sẽ nhập mật khẩu này để xác thực với máy chủ xác thực. Máy chủ xác thức sẽ sinh mật khẩu OTP và kiểm tra mật khẩu OTP được sinh ra trên ứng dụng phía người dùng, nếu giống nhau người dùng xác thực thành công.

# KẾT LUẬN

Báo cáo này đã trình bày tổng quan về mã OTP và các cách sinh, phân phối mã OTP cùng với đó là cách triển khai mã OTP bằng thuật toán HOTP. Nhờ việc thực hành làm bài tập này, chúng em có những hiểu biết thực tế hơn về lý thuyết mật mã nhờ đó có cái nhìn rõ hơn, thực tế hơn trong lĩnh vực này. Trong quá trình tìm hiểu, đọc tài liệu chúng em gặp rất nhiều khó khăn không chỉ về kiến thức chuyên môn mà cả vấn đề về ngoại ngữ vì vậy không tránh khỏi sai sót trong cách hiểu vấn đề. Em xin cảm ơn thầy Hán Trọng Thanh cùng sự giúp đỡ của các bạn trong lớp đã giúp chúng em hoàn thành đề tài.

# TÀI LIỆU THAM KHẢO

1. Dương, N. H. (2020, 12). Pinonote. Retrieved from Pinonote: https://pinonote.wordpress.com/2018/11/27/thuat-toan-hmac-based-one-time-password-algorithm-hotp-va-time-based-one-time-password-totp-trong-google-authenticator/ (truy cập lần cuối 20/12/2020)
2. Ebrahimi, A. (2020, 12). ONE-TIME PASSWORD. Retrieved from AtaEbrahimi: www.AtaEbrahimi.com (truy cập lần cuối 20/12/2020)