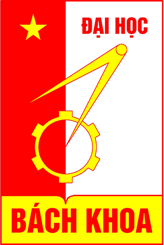
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐÀO TẠO QUỐC TẾ - SIE

\*\*\*\*\*

****

# BÁO CÁO MÔN AN NINH MẠNG

ĐỀ TÀI: ONE TIME PASSWORD

\*\*\*\*\*\*\*\*\*\*\*

**GVHD:** *ThS. Bùi Trọng Tùng*

Lớp : LTU13B

Danh sách sinh viên thực hiện:

1. Nguyễn Thị Vân Anh - 20148275

2. Nguyễn Xuân Hội - 20148103

3. Nguyễn Bình Minh - 20148165

Hà Nội, ngày 27 tháng 04 năm 2018

# DANH MỤC BẢNG VÀ HÌNH VẼ

|  |  |  |
| --- | --- | --- |
| Tên | Nội dung | Trang |
| Hình 1.1 | Mật khẩu là gì? | 7 |
| Hình 2.1 | Sinh OTP dựa trên S/Key | 13 |
| Hình 2.2 | Xác thực của S/Key OTP | 14 |
| Hình 2.3 | Thuật toán HOTP | 16 |
| Hình 2.4 | Thuật toán TOTP | 18 |
| Hình 2.5 | Minh họa thẻ EMV | 20 |
| Hình 2.6 | Minh họa thiết bị E-Token | 20 |
| Hình 2.7 | Chức năng tạo mã TOTP trên điện thoại  của ứng dụng facebook | 21 |
| Hình 3.0 | Ví dụ một số vụ việc tấn công người dùng | 25 |
| Hình 3.1 | Ví dụ một số vụ việc tấn công người dùng | 26 |
| Hình 3.2 | Khai thác lỗ hổng smart OTP | 26 |
| Hình 3.3 | Tấn công OTP bằng cách thu thập dữ liệu người dùng | 28 |
| Hình 4.1 | Đăng ký tại client (App) | 28 |
| Hình 4.2 | Cở sở dữ liệu | 29 |
| Hình 4.3 | Đăng nhập tại client | 30 |
| Hình 4.4 | Bước xác thực OTP sau khi Login thành công | 31 |

# DANH MỤC KÝ HIỆU VÀ VIẾT TẮT

|  |  |
| --- | --- |
| Ký hiệu | Ý nghĩa |
| OTP | One Time Password |
| CSDL | Cơ sở dữ liệu |
| HMAC | Hashing for Message Authentication Code |
| SHA | Secure Hash Algorithm |
| HOTP | HMAC-based One-Time Password |
| TOTP | Time-based One-Time Password |

# MỤC LỤC

DANH MỤC BẢNG VÀ HÌNH VẼ2

DANH MỤC KÝ HIỆU VÀ VIẾT TẮT3

MỤC LỤC4

LỜI NÓI ĐẦU6

I. Tổng quan về OTP và ứng dụng7

1.1. Khái quát7

1.1.1 Mật khẩu là gì?7

1.1.2 Phương pháp xác thực sử dụng mật khẩu7

1.1.3 Độ an toàn của mật khẩu8

1.2. Single static passwords – mật khẩu đơn tĩnh9

1.3. One-Time Passwords9

1.4. Phân loại OTP10

1.4.1 Phân loại theo cơ chế sử dụng OTP10

1.4.2 Phân loại theo phương pháp sinh OTP11

II. CÁC PHƯƠNG PHÁP SINH VÀ CHUYỂN GIAO OTP12

2.1. Các phương pháp sinh OTP12

2.1.1 Sinh OTP dựa trên hệ thống S/Key 13

2.1.2 Sinh OTP dựa trên thuật toán HOTP15

2.1.3 Sinh OTP dựa trên thuật toán TOTP 16

2.2. Các phương pháp chuyển giao OTP17

2.2.1 Chuyển giao OTP qua tin nhắn SMS17

2.2.2 Chuyển giao OTP qua Token19

2.2.3 Chuyển giao OTP qua ứng dụng trên smart phone21

III. OTP trong xác thực giao dịch ngân hàng 21

3.1. Sử dụng giao dịch thông qua Application của các ngân hàng 21

3.2. OTP trong các giao dịch trực tuyến22

3.3. Một số vụ việc tấn công xác thực người dùng 22

3.4. Một số kịch bản tấn công trong vụ việc của Ngân hàng Vietcombank 201623

IV. Ứng dụng demo OTP26

4.1. Tổng quan về ứng dụng27

4.2. Phân tích ứng dụng28

KẾT LUẬN31

# LỜI NÓI ĐẦU

Từ lâu, mật khẩu (password) được sử dụng rộng rãi trong khâu đăng nhập (log-on) để xác thực người dùng truy nhập vào các hệ thống máy tính và mạng. Từ đăng nhập vào các phần mềm ứng dụng trên máy tính cá nhân đến đăng nhập vào máy chủ công ty và cả website của các tổ chức tài chính, ngân hàng, phương tiện chính để xác thực người dùng chỉ là mật khẩu (tên đăng nhập hay username - cũng là một dạng password - không có ý nghĩa bảo mật vì thường không được giữ bí mật). Tuy nhiên, hầu hết các chuyên gia bảo mật đều nhận định password không còn an toàn trước các thủ đoạn tấn công tinh vi hiện nay. Mật khẩu có thể bị nghe lén, bị đánh cắp, hoặc bị phá mã (với các mật được mã hóa hoặc băm) và sau đó có thể bị lạm dụng tương đối dễ dàng.

Mật khẩu sử dụng một lần - OTP được giới thiệu để tăng cường độ an toàn trong quá trình xác thực người dùng, xác thực các giao dịch, đặc biệt là các giao dịch thanh toán trực tuyến trong các hệ thống ngân hàng.

An toàn thông tin (ATTT) đang phát triển nhanh chóng, trở thành một chuyên ngành lớn của ngành Công nghệ Thông tin. Các vấn đề cụ thể của ATTT như an ninh mạng, an toàn phần mềm nói chung hay ứng dụng Web… trở thành những vấn đề nóng trong việc phát triển hệ thống Công nghệ Thông tin.

Lời cảm ơn đến **ThS. Bùi Trọng Tùng** – Giảng Viên Đại Học Bách Khoa Hà Nội, người đã truyền đạt kiến thức về an ninh mạng đến với sinh viên chúng em.

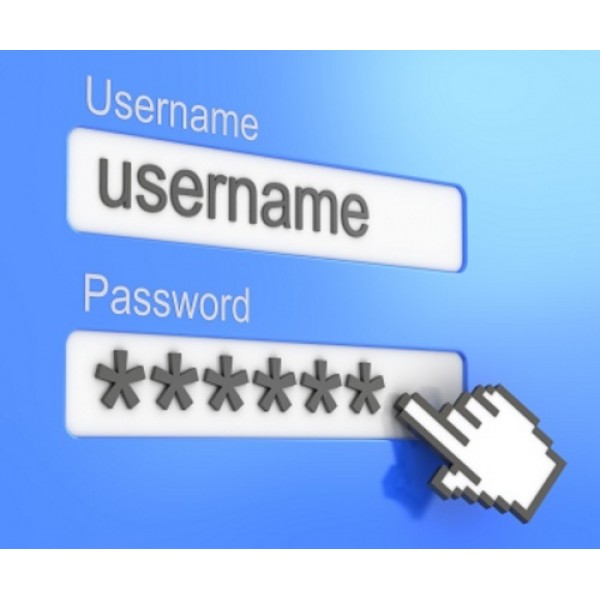
# **I. Tổng quan về OTP**

# **1.1. Khái quát**

# **1.1.1. Mật khẩu là gì ?**

Mật khẩu (password) thường là một xâu, chuỗi, loạt các ký tự mà dịch vụ Internet, phần mềm, hệ thống máy tính yêu cầu người sử dụng nhập vào bằng bàn phím trước khi có thể tiếp tục sử dụng một số tính năng nhất định, hay để xác minh danh tính của người dùng trong quá trình xác thực.

Mật khẩu thường đi cặp với tên truy cập (username) khi hệ thống cần phân biệt các người sử dụng khác nhau.



Hình 1.1: Mật khẩu là gì?

# **1.1.2. Phương pháp xác thực sử dụng mật khẩu**

Để đảm bảo an toàn, mật khẩu cần được giữ bí mật và chỉ bản thân người dùng mới biết. Mật khẩu thường được sử dụng trong một thời gian dài và trao đổi thường xuyên giữa máy khách (client) của người sử dụng với máy chủ (server). Sau khi người dùng gõ mật khẩu của mình, bên phía client sẽ xác thực mật khẩu với server. Nếu mật khẩu đúng server sẽ xác nhận và cấp quyền truy nhập tương ứng cho client và qua đó tới người dùng.

* Một số điểm yếu trên hệ thống xác thực bằng mật khẩu:
* Lưu trữ mật khẩu trong CSDL không an toàn
  + Database là một trong những nơi hay bị tấn công, dễ làm thất thoát dữ liệu.
* Truyền mật khẩu trên kênh không an toàn
  + Mọi địa chỉ trang web thường bắt đầu bằng các ký tự **HTTP** và trang web đó chưa được bảo mật.
* Người dùng không cẩn trọng:
  + Sử dụng mật khẩu yếu : sử dụng các dãy số dễ nhớ, các chữ cái dễ đoán, thân thuộc với người dùng...
  + Ghi chép mật khẩu vào văn bản
  + Chia sẻ mật khẩu cho người khác (vô tình hoặc cố ý)
* Lưu trữ mật khẩu:
* Lưu mật khẩu dưới dạng rõ: dễ đọc, dễ hiểu nhất khi bị người khác phát hiện.
  + Nguy cơ mất an toàn cao nhất
  + Tốn thời gian
* Lưu mật khẩu dưới dạng bản mã: mật khẩu đã được mã hóa
  + An toàn khi sử dụng hệ mật mã tốt, bảo vệ khóa giải mã an toàn
  + Hạn chế: cần thao tác giải mã bất cứ khi nào cần xác thực
* Lưu mật khẩu dưới dạng mã băm: Sử dụng một hàm băm được hỗ trợ của hệ quản trị CSDL để mã hóa chuỗi mật khẩu ban đầu thành một chuỗi được mã hóa sau đó mới lưu vào cơ sở dữ liệu. Các hàm băm có đặc điểm là các hàm một chiều. Dữ liệu sau khi mã hóa không thể giải mã thành dữ liệu chuỗi mật khẩu ban đầu được. Dù ai đó có kiểm soát được dữ liệu đi nữa cũng không biết mật mã của người dùng là gì. Tuy nhiên, có thể dõ tìm thông qua thuật toán vét cạn. Bởi người dùng hay sử dụng mật khẩu dễ nhớ và các hacker biết điều này do đó sẽ lập một danh sách các mật khẩu phổ biến nhất (weak password dictionary), sau đó dùng danh sách để thử sai và dò ra mật khẩu ban đầu.
  + Chi phí thấp hơn
  + Hạn chế: nguy cơ bị tấn công dò đoán dựa trên từ điển., vét cạn. Có thể hạn chế bằng cách đưa thêm “salt” vào mật khẩu trước khi băm
* Sử dụng máy chủ lưu trữ:
  + Giải pháp 1: Người thẩm tra yêu cầu máy chủ chuyển mật khẩu để xác thực
  + Giải pháp 2: Người thẩm tra đưa cho máy chủ thông tin người dùng. Máy chủ xác thực và thông báo lại kết quả

**1.1.3. Độ an toàn của mật khẩu**

Nếu sử dụng các mật khẩu quá đơn giản hay quá ngắn, việc sử dụng phương pháp tấn công dựa trên từ điển hoặc vét cạn có thể giúp tin tặc dễ dàng dò ra mật khẩu của người dùng mà không mất quá nhiều thời gian.

Trên thực tế, một mật khẩu an toàn cần thỏa mãn các yêu cầu sau:

* Độ dài của mật khẩu phải từ 8 ký tự trở lên. Khi độ dài mật khẩu đủ lớn, khả năng bị đoán hoặc tấn công vét cạn giảm đi.
* Không dùng một mật khẩu cho nhiều tài khoản.
* Mật khẩu không nên chứa các từ đơn giản, dễ đoán, như tên người thân, tên con vật yêu thích, hoặc ngày tháng năm sinh. Các mật khẩu như vậy thường dễ dàng bị dò tìm thông qua tấn công dựa trên từ điển.
* Mật khẩu phải là tổ hợp của các chữ cái in hoa, in thường, chữ số và ký tự đặc biệt (như các ký tự ? $ #,...). Khi số loại ký tự được sử dụng tăng lên, số tổ hợp mật khẩu có thể có trở lên rất lớn làm cho việc thực hiện tấn công vét cạn không khả thi.
* Dùng mật khẩu có sẵn trong USB hoặc đĩa mềm đi kèm (tránh không bị các phần mềm gián điệp lấy cắp khi gõ trực tiếp)

# **1.2. Single static passwords – mật khẩu đơn tĩnh**

Mật khẩu đơn tĩnh hay là mật khẩu truyền thống. Mật khẩu do người dùng tự đặt. Mỗi lần đăng nhập, người dùng chỉ cần nhập tên truy cập (username) là địa chỉ Email, Gmail của người dùng và mật khẩu. Trong hai yếu tố xác thực đó, thì chỉ có mật khẩu là chỉ được biết bởi người dùng, còn tên truy cập được để công khai. Vì vậy, chỉ có mật khẩu được sử đụng cho việc đảm bảo xác thực.

Mật khẩu đơn tĩnh thông thường được dùng trong khoảng thời gian lâu dài, chỉ được thay đổi khi cần thiết, người dùng cần phải tạo lại mật khẩu ( nghi ngờ có một thâm nhập khác , hoặc do người dùng muốn đổi…), hoặc có thể người dùng quên mật khẩu và cần tạo lại, hoặc mật khẩu đã hết hạn.

* Ưu điểm :
  + Tiện dụng , quá trình đăng nhập nhanh, không phức tạp.
* Nhược điểm :
* Có nhiều hệ thống sử dụng mật khẩu tĩnh đưa ra ít hoặc không có thông tin liên quan đến lịch sử hoạt động của người dùng đăng nhập tài khoản, nên người dùng không phát hiện được khi mà có sự thâm nhập khác vào tài khoản( trên thiết bị khác, thời gian đăng nhập khác …).
* Nếu hệ thống có lịch sử đăng nhập, thì người dùng cũng sẽ rất ít khi kiểm tra lịch sử. Một vài hệ thống có tính bảo mật cao hơn, là mỗi lần dăng nhập ở thiết bị khác, hay địa điểm khác sẽ thông báo cho người dùng (Gmail,Email…)
* Do mật khẩu được đặt bởi người dùng, nên mật khẩu sẽ là những thông tin quen thuộc đối với người dùng. Dẫn đến bị đoán bởi người lấy được thông tin người dùng.  Chính sách bảo mật, có đưa ra mật khẩu mạnh, gồm các chữ số, các chữ viết hoa, và các ký tự đặc biệt,… Dẫn đến gây khó nhớ cho người dùng.

# **1.3. One-Time Passwords**

Không giống như như mật khẩu tĩnh , OTP là mật khẩu được thay đổi mọi thời điểm, người dùng đăng nhập. Một OTP là một bộ các ký tự rằng có thể hoạt động như một xác thực tại duy nhất một thời điểm. Khi mật khẩu được sử dụng, nó không còn được sử dụng cho lần xác thực nào khác nữa. Thậm chí khi kẻ tấn công lấy được mật khẩu, nó có thể đã được sử dụng trước thời điểm kẻ tấn công thực thi, điều đó là vô ích với quá trình xác thực.

One-Time Passwords dựa trên phương thức xác thực hai yếu tố. Xác thực yếu tố áp dụng thêm phương thức thứ hai để xác thực người dùng, thường xuyên là đi kèm với tên truy cập và mật khẩu. Yếu tố xác thực thứ hai phải là các yếu tố thực hiện một cách vật lý( điện thoại di động, ứng dụng trên điện thoại di động, SMS…). Bởi việc sử dụng yếu tố xác thực thứ hai này, người dùng sẽ an toàn hơn.

* Ưu điểm : Tính bảo mật cao dựa trên xác thực hai yếu tố, sinh khóa ngẫu nhiên, kịch bản tấn công tốt nhất là vét cạn. Nên đảm bảo tính bí mật tuyệt đối.
* Nhược điểm:
  + Quá nhiều bước xác thực, gây không thuận tiện tới người dùng.
  + Chi phí cao hơn qua SMS gửi tin, đây là phương thức phổ biến được dùng để vận chuyển mã OTP. Khác với,vận chuyển qua Email hay Gmail, chi phí sử dụng dường như không đáng kể.

# **1.4 Phân loại OTP**

# **1.4.1. Phân loại theo cơ chế sử dụng OTP**

Về tổng thể, có 3 nhóm cơ chế sử dụng mật khẩu OTP: OTP được sinh ở phía claimant, OTP được sinh bởi cả claimant và verifier, OTP được sinh bởi verifier.

* OTP được sinh bởi claimant:

Mỗi khi tham gia vào phiên xác thực, claimant sẽ sinh ra một OTP và gửi đến verifer, verifer sẽ căn cứ vào giá trị OTP này để xác thực claimant. Ví dụ điển hình cho trường hợp này là các lược đồ xác thực sử dụng mật khẩu một lần Lamport.

* OTP được sinh bởi cả claimant và verifier:

Trong trường hợp này, hai bên xác thực sẽ thống nhất một lược đồ sinh OTP phụ thuộc vào một giá trị bí mật được chia sẻ giữa hai bên. Mỗi lần tham gia vào phiên xác thực, claimant sinh ra một OTP và gửi cho verifier. Bản thân verifier cũng sinh ra một OTP và so sánh với OTP nhận được từ claimant. Kết quả so sánh sẽ cho phép verifier kết luận về tính hợp lệ của claimant.

* OTP được sinh bởi verifier:

Trong trường hợp này OTP được sinh bởi verifier và được gửi cho claimant qua kênh liên lạc thứ hai, khác với kênh được dùng để khởi xướng phiên xác thực. Kênh thứ hai này thường là kênh tin nhắn SMS, kênh thoại hoặc kênh email. Sau khi nhận được OTP từ verifier, claimant sẽ gửi trả lại cho verifier qua kênh thứ nhất. Verifier so sánh OTP nhận được từ claimant với OTP mà nó đã gửi đi. Nếu kết quả là giống nhau thì claimant được xác thực, ngược lại thì claimant bị từ chối. Có thể thấy rằng, về thực chất OTP trong trường hợp này được sử dụng để chứng minh sự sở hữu kênh liên lạc thứ hai của claimant (ví dụ: số điện thoại di động, hộp thư điện tử) mà đã được claimant đăng ký với verifier khi thiết lập tài khoản. Cơ chế sử dụng OTP này hiện đang được áp dụng bởi nhiều ngân hàng và hệ thống giao dịch trực tuyến.

# **1.4.2. Phân loại theo phương pháp sinh OTP**

* Phương pháp sinh OTP theo thời gian

Theo phương pháp sinh OTP theo thời gian, người dùng sẽ được cấp một thiết bị sinh mã được gọi là token. Token gồm có ba thành phần chính: một mã seedcode, một đồng hồ đếm thời gian, và một thuật toán mã hóa một chiều.

* Mã seedcode là mã được nhà sản xuất cài đặt sẵn trong token. Mỗi token có một mã seedcode khác nhau. Và mã seedcode này cũng được lưu lại trong hệ thống của nhà cung cấp dịch vụ tương ứng với tên truy nhập của người dùng.
* Đồng hồ đếm thời gian là đồng hồ của token, nó được đồng bộ với đồng hồ của hệ thống trước khi giao cho người dùng. Mỗi khi người dùng bấm nút sinh mã, token sẽ lấy biến thời gian của đồng hồ. Biến thời gian được lấy chi tiết đến từng phút, hoặc 30 giây.
* Thuật toán mã hóa : sử dụng thuật toán băm SHA.
* Phương pháp sinh OTP bằng thuật toán dựa trên sự kiện

Trong cơ chế này người dùng cũng được cấp một token như ở trên, nhưng bên trong token sẽ có một bộ đếm sự kiện thay vì đồng hồ đếm thời gian. Sự kiện được nhắc đến ở đây là sự kiện mà người dùng bấm nút sinh mã trên Token. Mỗi token sẽ chứa một số mã hữu hạn, có thứ tự và không thay đổi. Số lượng các mã hữu hạn đó được gọi là cửa sổ. Kích thước của cửa sổ này càng lớn thì độ bảo mật của giải pháp càng cao.

# **II. CÁC PHƯƠNG PHÁP SINH VÀ CHUYỂN GIAO OTP**

# **2.1. Các phương thức sinh OTP**

# **2.1.1 Sinh OTP dựa trên hệ thống S/Key**

* Tóm tắt:

- S/Key OTP là phương pháp sinh OTP bằng thuật toán dựa trên mật khẩu cũ.

- S/Key OTP được thiết kế để chống lại kiểu tấn công nghe lén/phát lại.

- Có hai phía hoạt động của một hệ thống S/Key OTP. Ở phía máy khách, một OTP riêng biệt được tạo ra. Ở phía máy chủ, máy chủ phải xác minh OTP và cấp phép khi người dùng thay đổi mật khẩu.

- Thuật toán của bên xác thực và người dùng phải đồng bộ với nhau. Mỗi lần một người dùng được xác thực thành công, bên xác thực sẽ chỉ chấp nhận một OTP kế tiếp được sinh ra bởi thuật toán. OTP dựa trên S/Key chỉ có giá trị cho một lần sử dụng.

* Cách triển khai và hoạt động:

- Máy chủ là máy tính sẽ thực hiện xác thực:

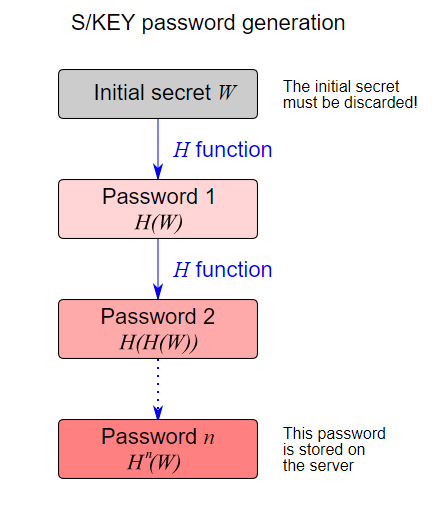
1. Bước này bắt đầu bằng khóa bí mật W. Khóa này có thể do người dùng cung cấp hoặc có thể được tạo bởi máy tính. Nếu khóa này được tiết lộ, thì bảo mật của S/Key bị xâm phạm.
2. Một hàm băm mật mã H được áp dụng n lần cho W, do đó tạo ra một chuỗi băm của n mã OTP. OTP là kết quả của việc áp dụng hàm băm mật mã:

H(W), H(H(W)), ..., Hn(W).

1. Khóa W ban đầu bị loại bỏ.
2. Người dùng được cung cấp n mật khẩu, được in theo thứ tự ngược lại.

Hn(W), Hn−1(W), ..., H(H(W)), H(W).

1. Các mật khẩu H (W), H (H (W)), ..., Hn − 1 (W) được loại bỏ khỏi máy chủ. Chỉ có mật khẩu Hn (W), mật khẩu ở đầu danh sách của người dùng, được lưu trữ trên máy chủ.



Hình 2.1: Sinh OTP dựa trên S/Key

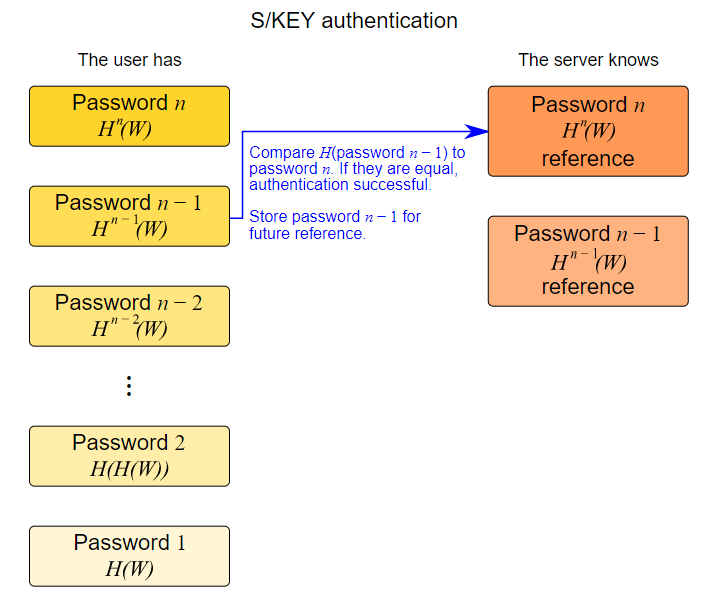
* Xác thực:

- Sau khi người dùng tạo mật khẩu, máy chủ áp dụng S/Key sinh ra n mã OTP và cung cấp cho người dùng. Người dùng lưu trữ lại n mã này để sử dụng.

- Trong nhiều trường hợp, mã OTP đầu tiên sẽ là cùng một OTP mà máy chủ đã lưu trữ. Mã OTP đầu tiên này sẽ không được sử dụng để xác thực, OTP thứ hai sẽ được sử dụng để thay thế:

* Người dùng cung cấp cho máy chủ với OTP thứ hai pwd trên danh sách và xóa bỏ mật khẩu đó.
* Máy chủ cố gắng tính H(pwd), trong đó pwd là OTP được cung cấp. Nếu H(pwd) tạo ra được mật khẩu đầu tiên (máy chủ lưu trữ), thì việc xác thực thành công. Máy chủ sau đó sẽ lưu trữ pwd làm tham chiếu hiện tại.

- Trong lần xác thực tiếp theo, người dùng sẽ cung cấp OTPi. OTP cuối cùng trên danh sách in, OTPn là OTP đầu tiên được tạo bởi máy chủ, H(W), trong đó W là khóa bí mật ban đầu. Máy chủ sẽ tính toán H(OTPi) và sẽ so sánh kết quả với OTPi-1, được lưu trữ dưới dạng tham chiếu trên máy chủ.



Hình 2.2: Xác thực của S/Key OTP

* Bảo mật:

- Tính bảo mật của S/Key dựa trên độ phức tạp của việc đảo ngược các hàm băm mật mã.

- Giả sử kẻ tấn công chiếm được một mật khẩu OTP và sử dụng xác thực thành công. Giả sử đây là OTPi, mật khẩu này đã vô ích đối với các xác thực tiếp theo, bởi vì mỗi OTP chỉ có thể được sử dụng một lần. Sẽ rất thú vị nếu kẻ tấn công tìm ra OTPi-1, vì mật khẩu này là mật khẩu sẽ được sử dụng để xác thực tiếp theo.Tuy nhiên, điều này sẽ yêu cầu đảo ngược hàm băm đã tạo mật khẩu bằng cách sử dụng OTPi(H(OTPi−1) = OTPi), điều này là cực kỳ khó thực hiện với các hàm băm mật mã hiện tại.

- S/Key có thể bị tấn công bằng một cuộc tấn công dưới dạng MITM.

# **2.1.2 Sinh OTP dựa trên thuật toán HOTP**

* Tóm tắt:

- HOTP được triển khai dựa trên các mã xác thực thông điệp bằng hàm băm HMAC và hàm băm SHA-1 thường được sử dụng. HOTP được đưa ra bởi tổ chức Initiative for Open Authentication (OATH).

- Về mặt khái niệm, HOTP tính giá trị băm SHA-1 dựa trên HMAC được mã khóa bởi một khóa bí mật được chia sẻ trên một bộ đếm.

* Cách triển khai và hoạt động:

- Những bước tính toán để sinh ra một mật khẩu HOTP như sau:

1. Hai bên người dùng đưa ra khóa bí mật dùng chung K.
2. Khởi tạo bộ đếm C = 0. Bộ đếm này phải được đồng bộ hóa giữa trình tạo HOTP (máy khách) và trình xác thực HOTP (máy chủ).
3. Giá trị của HOTP là HMAC được tính toán bằng SHA-1.

HMAC(K,m) = SHA-1(K⊕0x5c5c…∥SHA1(K⊕0x3636…∥ m))

Trong đó: m là một "thông điệp", ⊕ là phép XOR và ∥ là phép nối

1. Đặt Truncate là hàm lựa chọn 4 byte theo một cách nào đó.

HOTP(K,C) = Truncate(HMAC-SHA-1(K,C))

1. Người dùng cần xác thực gửi giá trị Truncate(H(K,C)) & 0x7FFFFFFF cho server.
2. Bên server cũng tính toán giá trị hàm Truncate() theo cách tương tự. Nếu giá trịxác thực do server tính toán khớp với giá trị người dùng cung cấp thì người dùng được xác thực.
3. Cả hai bên tăng giá trị của C.

- Với hệ thống S/Key, kết quả thường quá dài để người sử dụng nhập vào thiết bị. Thay vì chuyển kết quả thành chuỗi kí tự như S/Key, HOTP đơn giản chỉ tính ra giá trị như sau:

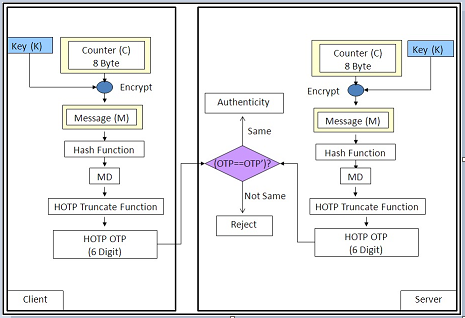
Value OTP = HOTP(K,C) mod 10d

Trong đó:

K là một khóa bí mật;

C là bộ đếm;

d là số chữ số mong muốn của kết quả OTP (thường là 6-8).



Hình 2.3: Thuật toán HOTP

* Bảo mật:

- Cách thức tấn công tốt nhất có thể chống lại HOTP là kẻ tấn công sẽ cố gắng thu thập thông tin của người dùng để từ đó xây dựng một hàm F có thể sinh ra giá trị HOTP. Tất nhiên, khả năng thành công là rất thấp.

# **2.1.3 Sinh OTP dựa trên thuật toán TOTP**

* Tóm tắt:

- Phát triển dựa trên thuật toán HMAC-based One-Time Password (HOTP) để hỗ trợ thêm yếu tố thời gian thay đổi.

- Giá trị thời gian được đưa vào nhằm xác định khoảng thời gian sống cho OTP, điều này giúp tăng tính bảo mật.

* Cách triển khai và hoạt động:

- Ta có thuật toán HOTP:

HOTP(K,C) = Hàm rút gọn(HMAC-SHA-1 (K,C))

- Thuật toán TOTP là biến thể của HOTP có thêm thuộc tính thời gian T, với T phụ thuộc vào một khoảng thời gian tham chiếu (time reference) nhất định và một bước thời gian (time step) cố định. Giá trị T này thay thế giá trị C trong HOTP.

=> TOTP = HOTP(K, T)

Trong dó:

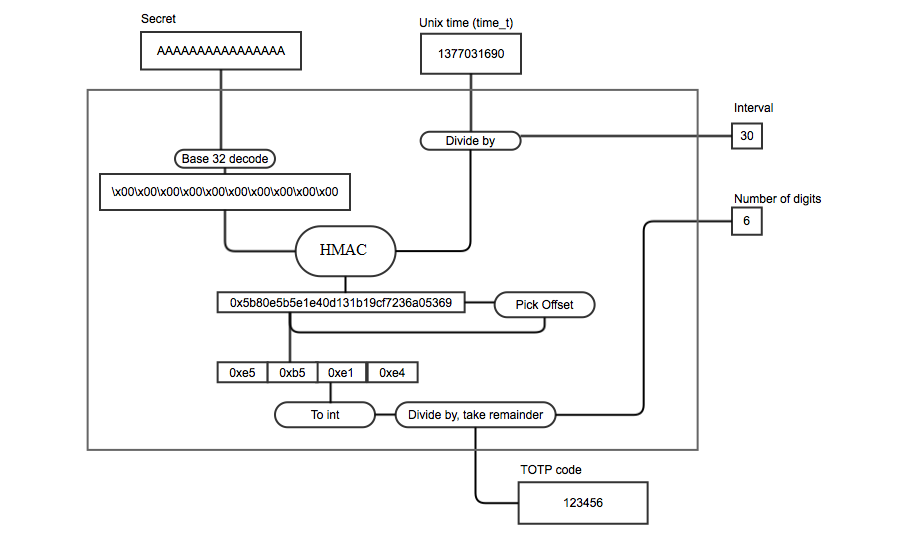
T = (Thời gian hiện tại dưới dạng Unix - T0) / X

X: thể hiện bước thời gian bằng giây (mặc định X = 30 giây) và là một tham số hệ thống.

T0: là thời gian dưới dạng Unix khi bắt đầu đếm các bước thời gian (giá trị mặc định là 0, tức là kỷ nguyên Unix-Unix epoch) và cũng là tham số hệ thống.

- *Yêu cầu thuật toán:*

* Bên chứng thực (e.g., token, soft token) và bên xác nhận (authentication or validation server) phải biết hoặc có thể lấy được thời gian hiện tại dưới dạng Unix(1) để sinh mã OTP.
* Giữa bên chứng thực và bên xác nhận phải chia sẻ cùng một khóa bí mật hoặc có kiến thức về cùng một phép biến đổi bí mật để tạo ra được một khóa bí mật giữa cả hai.
* Thuật toán phải sử dụng HOTP như một khuôn chính để phát triển lên. Nhưng TOTP phải sử dụng hàm HMAC-SHA-256 (dựa trên hàm hash SHA-256) hoặc HMAC-SHA-512 (dựa trên hàm hash SHA-512) thay vì hàm HMAC-SHA-1 (dựa trên hàm hash SHA-1) của HOTP.
* Bên chứng thực và bên xác nhận phải sử dụng cùng giá trị bước thời gian (time-step) X.
* Phải có một khóa bí mật duy nhất cho mỗi bên chứng thực.
* Các khóa nên được tạo ngẫu nhiên.
* Để hoàn chỉnh thuật toán nó phải hỗ trợ một giá trị thời gian T lớn hơn một số nguyên 32 bit khi vượt quá năm 2038. Giá trị của các tham số hệ thống X và T0 được thiết lập trước trong quá trình cấu hình và được trao đổi giữa bên chứng thực và bên xác nhận như là một phần của bước cấu hình.



Hình 2.4: Thuật toán TOTP

* Xác thực:

- Người sử dụng tính giá trị T, sau đó gửi cho bên xác thực HOTP(K,T) trong đó HOTP là thuật toán sinh OTP dựa trên các hàm băm.

- Bên xác thực có thể xác nhận mật khẩu đó hợp lệ hay không, vì bất kỳ mật khẩu nào được sinh ra bên ngoài khoảng thời gian từ T0 đến thời gian hiện tại đều khác nhau. Và vì có một khóa được dùng chung nên nhà xác thực có thể chắc chắn rằng mật khẩu được sinh ra bởi một người đang sở hữu khóa chia sẻ đó.

* Bảo mật:

- Việc triển khai không giới hạn các lần đăng nhập khiến kể tấn công có thể thử nhiều lần để dò mã TOTP.

- Nếu kẻ tấn công đánh cắp được khóa bí mật K thì có thể tạo mã TOTP mới hợp lệ theo ý muốn. Điều này rất là nguy hiểm nếu kẻ tấn công có được cơ sở dữ liệu xác thực lớn.

# **2.2. Các phương pháp chuyển giao OTP**

# **2.2.1 Chuyển giao OTP qua tin nhắn SMS**

Một công nghệ phổ biến thường được sử dụng cho việc cung cấp các OTP là tin nhắn văn bản. Từ các hệ thống xác thực tin nhắn SMS , một mật khẩu OTP được sinh và gửi thông qua một tin nhắn văn bản đến điện thoại di động của người dùng. Tương tự như với thẻ bảo mật, người dùng phải nhập OTP đã nhận được nhập vào giao diện đăng nhập nhằm xác minh tính chính xác và khẳng định đủ an toàn để truy nhập hệ thống dịch vụ hoặc ứng dụng.

Phương pháp này là một trong các phương pháp thông dụng nhất mà các ngân hàng thường sử dụng. Đầu tiên, khách hàng đăng ký dịch vụ cùng với số điện thoại cho ngân hàng.Khi thực hiện giao dịch, khách hàng nhập thông tin thanh toán, máy chủ của ngân hàng sẽ tạo OTP và gửi cho khách hàng thông qua một tổng đài bằng tin nhắn SMS. Sau khi khách hàng nhận được tin nhắn chứa OTP , khách hàng sẽ phải nhập OTP vào để máy chủ xác thực. Máy chủ kiểm tra OTP, nếu đúng và trong thời gian hợp lệ thì sẽ thực hiện giao dịch, nếu không sẽ có báo lỗi.

* Ưu điểm:
  + Tin nhắn SMS là một kênh truyền thông phổ biến, có sẵn trong gần như tất cả các thiết bị cầm tay với lượng người sử dụng khá lớn. Đây cũng là một lợi thế so sánh lớn của loại công cụ này so với việc sử dụng các loại thiết bị phân phối OTP khác.
  + Có nhiều tiềm năng để phổ biến tới nhiều người tiêu dùng với tổng chi phí thấp.
* Nhược điểm:
  + Không được bảo vệ cần thiết để chống lại các cuộc tấn công mà sự tinh vi đang ngày càng gia tăng. Các tin nhắn có thể được mã hóa bằng cách sử dụng một số tiêu chuẩn như A5/x, điều mà theo báo cáo của một số nhóm tin tặc có thể giải mã trong vòng vài phút hoặc vài giây, hoặc nó không được mã hoá bởi những nhà cung cấp dịch vụ khi nhận và gửi đi tới tất cả.
  + Trong trường hợp chuyển vùng, qua nhiều hơn một nhà mạng điện thoại di động đơn lẻ rất cần phải tạo được sự tin tưởng. Vì bất cứ ai đánh cắp được thông tin này, đều có thể gắn kết với những kẻ tấn công, chẳng hạn như các cuộc tấn công “man-in-the middle”.

# **2.2.2 Chuyển giao OTP qua Token**

Token là một thiết bị dùng để xác thực người dùng thay cho cơ chế ID/Username và mật khẩu đăng nhập. Mỗi thiết bị token đều phân biệt nhau và được nhà cung cấp dịch vụ gán với một người dùng cụ thể. OTP có thể được sinh trên token.

Thiết bị Token gồm 2 loại chính: thẻ EMV và E-Token.

* Thẻ EMV: là dạng thẻ chíp có công dụng tạo OTP, không có khả năng dùng để chi tiêu như các loại thẻ thanh toán khác. EMV là chuẩn thẻ thông minh do 3 liên minh thẻ lớn nhất thế giới là Europay, Master Card và Visa International đưa ra. Các liên minh này đã khuyến cáo các nước về việc cần thiết phải chuyển đổi từ thẻ từ có tính bảo mật thấp sang thẻ chíp có tính năng bảo mật cao hơn rất nhiều. Khuyến cáo này được đưa ra từ năm 2004, sau những con số thiệt hại do gian lận thẻ ngày càng lớn. Sau mốc thời gian quy định, nếu ngân hàng nào còn sử dụng thẻ từ, có thể chịu phạt lên tới 50 ngàn USD một năm tính theo các vụ gian lận thẻ. Mức phạt này áp dụng cho cả ngân hàng phát hành thẻ và ngân hàng chấp nhận thẻ.



Hình 2.5: Minh họa thẻ EMV

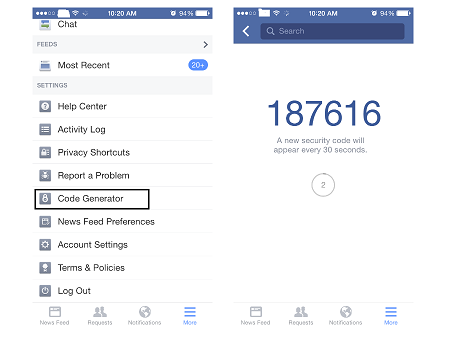
* E-Token: là thiết bị có công dụng tạo OTP dựa trên sự kết nối thiết bị với máy tính mà khách hàng đang thực hiện giao dịch. E-Token là một thiết bị nhận dạng số, được tích hợp những giải pháp phần mềm bảo mật chuyên dụng, theo đúng chuẩn quốc tế và kết nối với máy tính thông qua cổng giao tiếp USB. E-Token cho phép cả người dùng lẫn người quản trị bảo mật và quản lý hiệu quả quá trình chứng thực người dùng hệ thống bằng cách lưu trữ và sinh mật khẩu, chứng chỉ số và mã hóa tất cả thông tin đăng nhập (cả khóa chung và khóa riêng). E-Token cung cấp nền tảng bảo mật an toàn, hiệu quả, dễ sử dụng và triển khai trên diện rộng.



Hình 2.6: Minh họa thiết bị E-Token

# **2.2.3 Chuyển giao OTP qua ứng dụng trên smart phone**

Ngoài các phương pháp trên, người dùng còn có thể dùng ngay chiếc điện thoại di động của mình để cài đặt phần mềm sinh TOTP miễn phí từ nhà cung cấp dịch vụ. Sau khi cài đặt và cung cấp các thông tin đăng nhập thành công với hệ thống, người dùng có thể bắt đầu tạo mã OTP. Phương pháp này rất nhanh và tiện lợi nhưng cũng có rủi ro rất cao nếu người dùng làm mất điện thoại hoặc bị lộ OTP khi cho người khác mượn điện thoại.



Hình 2.7: Chức năng tạo mã TOTP trên điện thoại

của ứng dụng facebook

# **III. OTP trong xác thực giao dịch ngân hàng**

# **3.1. Sử dụng giao dịch thông qua Application của các ngân hàng**

Bước 1: Khách hàng đăng nhập vào hệ thống app của các ngân hàng VietinbankIpay, Vietcombank…

Bước 2: Khách hàng thực hiện các giao dịch

Bước 3: Hệ thống gửi mã OTP qua tin nhắn SMS tới số điện thoại mà khách hàng đã đăng ký

Bước 4: Khách hàng nhập mã OTP nhận được vào hệ thống để xác nhận chuyển tiền

# **3.2. OTP trong các giao dịch trực tuyến**

Bước 1: Khách hàng yêu cầu giao dịch

Bước 2: Khách hàng nhập 3 thông tin Số thẻ, chủ tk, ngày hết hạn

Bước 3: Hệ thống gửi mã OTP qua tin nhắn SMS tới số điện thoại mà khách hàng đã đăng ký

Bước 4: Khách hàng nhập mã OTP nhận được vào hệ thống để xác nhận giao dịch thành công

Đối với thẻ visa cần thêm thông tin số CVV

# **3.3. Một số vụ việc tấn công xác thực người dùng**



Hình 3.0: Ví dụ một số vụ việc tấn công người dùng



Hình 3.1: Ví dụ một số vụ việc tấn công người dùng (2)

# **3.4. Một số kịch bản tấn công trong vụ việc của Ngân hàng Vietcombank 2016**

* **Tấn công số 1: Phishing**

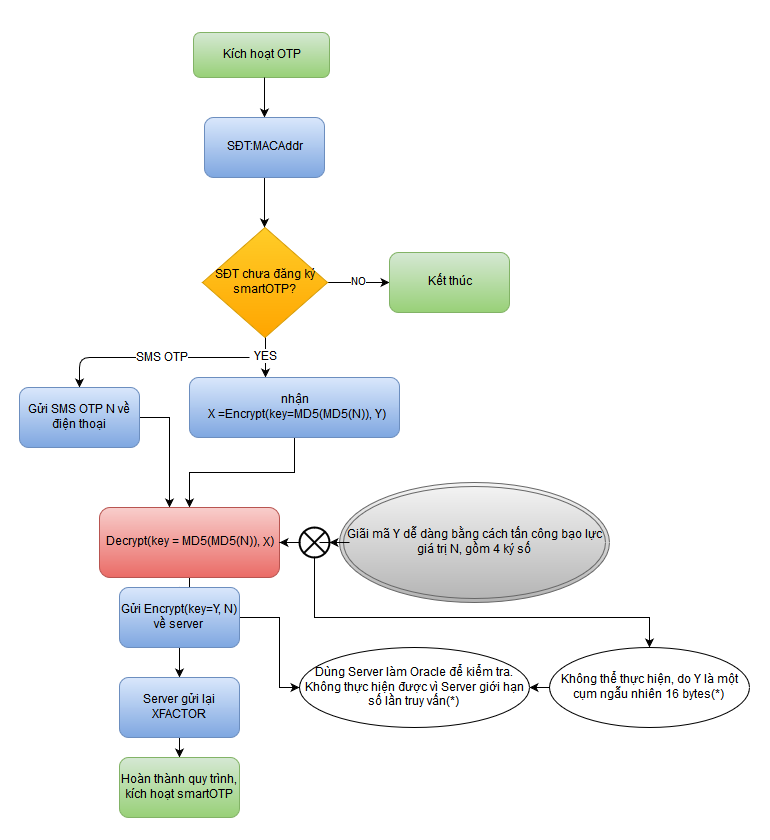
Đây là hướng tấn công đơn giản và dễ thực hiện nhất. Tại một thời điểm nào đó trước khi nạn nhân bị rút nửa tỉ đồng, kẻ xấu đã:

Bước 1: Lừa nạn nhân vào trang web giả mạo để lấy thông tin tên người dùng và mật khẩu.

Bước 2: Tiếp tục lừa nạn nhân trên giao diện trang web giả mạo để lấy SMS OTP.

Bước 3: Sử dụng SMS OTP để âm thầm kích hoạt Smart OTP. Vì Smart OTP có thể sử dụng song song với SMS OTP, nạn nhân không hề biết tài khoản đã hoàn toàn bị kiểm soát.

* **Tấn công số 2: Khai thác lỗ hổng của Smart OTP**



Hình 3.2: Khai thác lỗ hổng của smart OTP

Bước 1: Gửi số điện thoại của nạn nhân và một địa chỉ MAC bất kỳ đến máy chủ của VCB.

Bước 2: Lúc này máy chủ sẽ trả về R = Encrypt(key = MD5(MD5(N)), Y). Vì N là một chuỗi rất ngắn, chỉ có 4 chữ số, do đó kẻ tấn công có thể dò N.

* + Với mỗi giá trị dự đoán N’ kẻ tấn công sẽ tính

Y' = Decrypt(key=MD5(N’), data=R).

* + Tùy thuộc vào biên giá trị của Y mà quá trình dò này có thể thực hiện offline.
  + Nếu Y là một chuỗi hoàn toàn ngẫu nhiên, kẻ tấn công có thể sử dụng Y’ và N’ để thực hiện bước 3 và dựa vào phản hồi của máy chủ VCB để tìm N’ = N.
  + Máy chủ VCB có thể ngăn chặn sử dụng Y’ và N’ để dò N, nhưng chúng tôi không có điều kiện để xác minh.

Bước 3: Sau khi đã có N, kẻ tấn công có thể thực hiện tiếp các bước 3, 4 và 5.

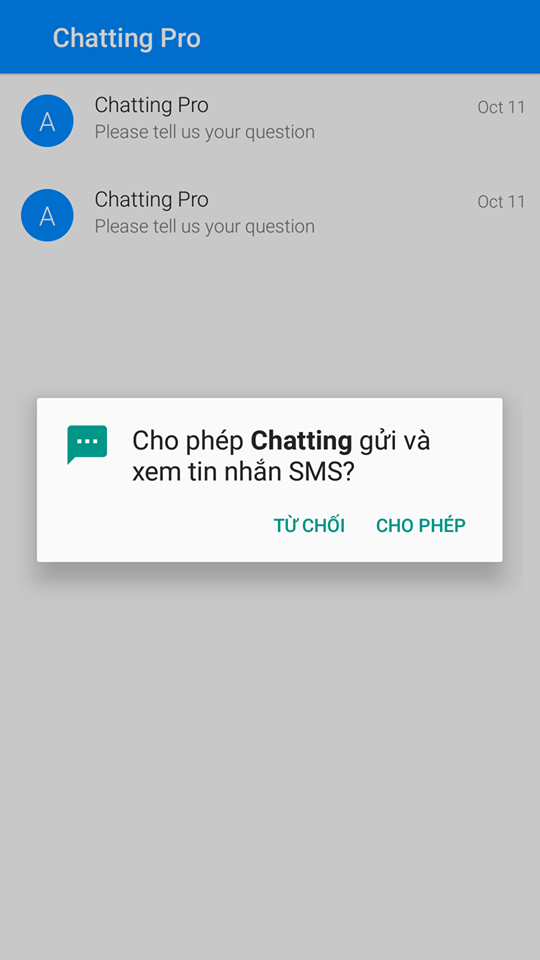
* **Tấn công số 3: Tấn công OTP bằng cách thu thập dữ liệu người dùng**

Bước 1: Người dùng cài các ứng dụng

Bước 2: Yêu cầu truy cập SMS,danh bạ

Bước 3: Khi người dùng có tin nhắn gửi đến bí mật gửi đến server của mình 🡪 Người dùng có thể bị mất mã OTP

Chi tiết cách thực hiện :



Khi người dùng cài đặt ứng dụng trên GooglePlay và chạy ứng dụng sẽ hiện lên thông báo yêu cầu truy cập vào nhưng tài nguyên của điện thoại như SMS, Danh bạ…. Nếu người dùng cho phép thì ứng dụng sẽ có quyền đọc tin nhắn của người dùng.

Đối với những chiếc điện thoại chạy hệ điều hành Android từ 6.0 trở lên bắt buộc phải xin quyền của người dùng mới được phép truy cập vào các tài nguyên đó. Thông thường người dùng khi cài đặt ứng dụng, rất ít người quan tâm đến các quyền này được cấp rất có thể thông tin người dùng bị đánh cắp một cách bí mật. Nhưng với những chiếc điện thoại chạy hệ điều hành Android từ 5.0 thì nhưng quyền này không được xin phép người dùng khi chạy ứng dụng mà mặc định được cấp khi người dùng cài ứng dụng. Khi quyền này được cấp thì ứng dụng có thể bí mật đọc tin nhắn vào gửi tới Server của họ một cách bí mật. Thậm trí nếu quyền này đã được người dùng cấp thì không cần chạy ứng dụng vẫn có thể đọc tin nhắn và gửi tới server ngầm mà người dùng không hề biết chỉ cần chiếc điện thoại đó có kết nối Internet.

Vậy chúng ta có nên tin tưởng vào những ứng dụng có quyền truy cập SMS, Danh bạ… hay không??

Video ứng dụng demo đã được đăng tải lên googlePlay được gửi kèm theo báo cáo.

# **IV. Ứng dụng demo OTP**

# **4.1. Tổng quan về ứng dụng**

- Chương trình demo gồm các tác vụ đăng ký, đăng nhập xác thực bước 2 OTP

- Xây dựng theo mô hình client-server

- Sử dụng IIS(Internet Information Services ) xây dựng server chạy trên nền window được viết bằng restful api c#

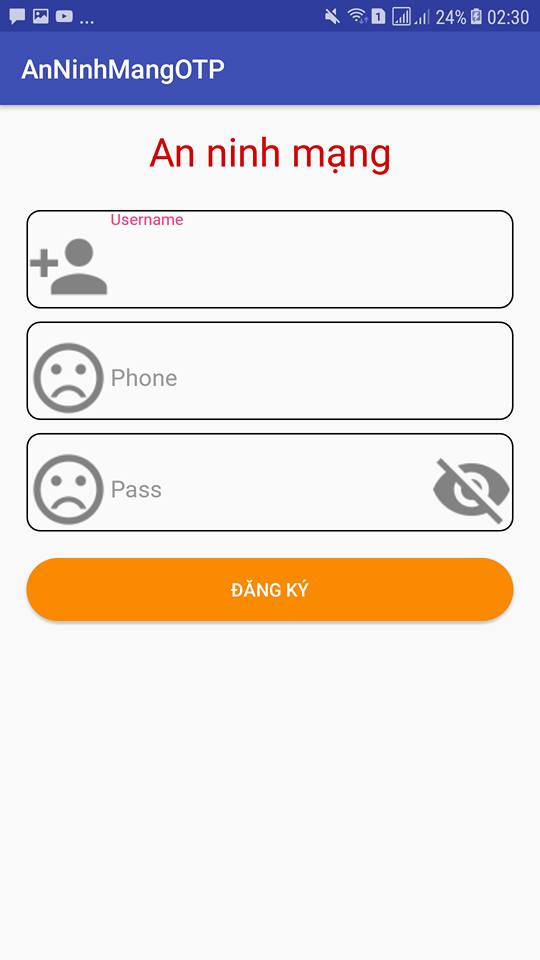
- Sử dựng hệ quản trị cơ sở dữ liệu SQL Server để lưu dữ liệu user

- Client là các ứng dụng sử dụng trên nền tảng Android viết bằng ngôn ngữ java

- Sử dụng api của speedsms.vn để gửi tín nhắn otp đến user

# **4.2. Phân tích ứng dụng**

* **Đăng ký**



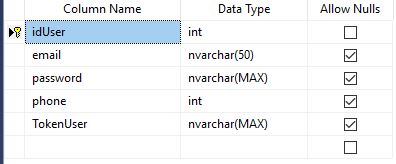
Hình 4.1: Đăng ký tại client (App)

Bước 1: Sau khi người dùng điền đẩy đủ thông tin và ấn nút đăng ký thì client (ứng dụng) yêu cầu với Server đăng ký thông tin.

Bước 2: Server sẽ kiểm tra thông tin người dùng đã yêu cầu với cơ sở dữ liệu và trả về kết quả thành công hay thất bại.

Bước 3: Nếu thành công sẽ lưu các thông tin sau vào cơ sở dữ liệu.

* + Username
  + Số điện thoại user
  + Password
  + TokenUser(Định danh cho mỗi user)



Hình 4.2: Cở sở dữ liệu

- Cách thức bảo mật password người dùng và sinh Token User

* Bảo mật password trong cơ sở dữ liệu: Sử dụng thuật toán mã hóa HMACSHA256 với một khóa bí mật:

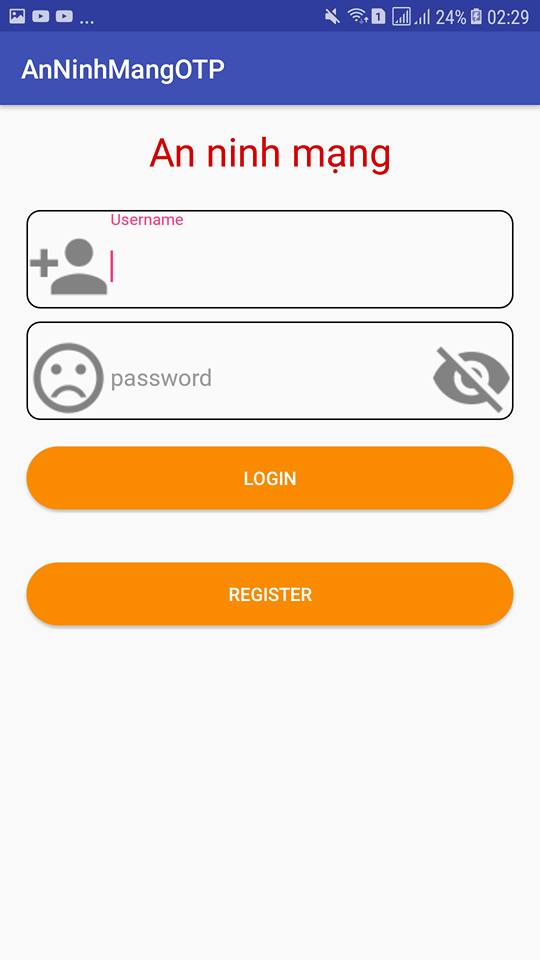
**hash\_hmac("SHA256", “Password", "secret key")**

* Cách tạo ra Token User: Token User đóng vai trò là SeekCode trong thuật toán sinh TOTP để sinh OTP.

**hash\_hmac("SHA256", “Email+ Password +**

**số ngẫu nhiên 6 chữ số", "secret key")**

* **Đăng nhập**

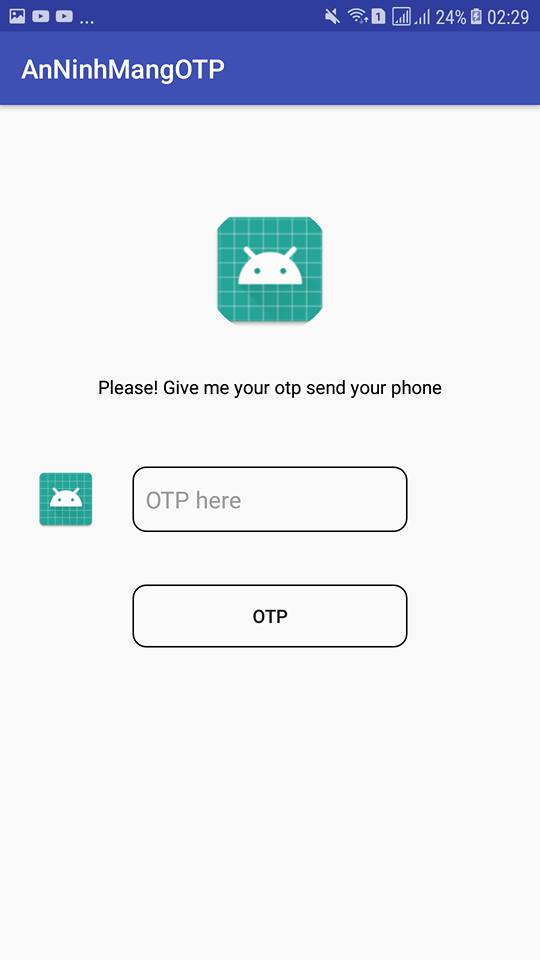


Hình 4.3: Đăng nhập tại client

Bước 1: Sau khi người dùng điền đẩy đủ thông tin sẽ yêu cầu với Server đăng nhập.

Bước 2: Server kiểm tra thông tin user trong cơ sở dữ liệu và trả về kết quả cho client bao gồm thông tin của User như email, phone, Token User.

Bước 3: Sau khi Server trả về kết quả ở bước 2. Client sẽ thực hiện yêu cầu lên sever với tham số truyền lên là Token User, yêu cầu send SMS OTP đến số điện thoại người dùng để xác thực bước đăng nhập bước 2.



Hình 4.4: Bước xác thực OTP sau khi Login thành công

Bước 4: Server kiểm tra TokenUser của user nếu hợp lệ thì tạo mã OTP theo TOTP và đồng thời gửi mã OTP về số điện thoại của người dùng và mã hóa OTP theo thuật toán HMACSHA256 với một khóa bí mật giữa client và Server và trả về chuỗi mã hóa đó cho client.

Bước 5: Sau khi nhận được OTP từ sms . Người dùng sẽ nhập mã OTP đó vào ô nhập OTP. Tại Client sẽ thực hiện mã hóa OTP mà người dùng nhập vào theo thuật toán HMACSHA256 với khóa bí mật giữa client và Server và so sánh kết quả trả về từ server. Nếu trùng khớp thì xác thực OTP thành công.

**KẾT LUẬN**

Mật khẩu sử dụng một lần (OTP) là mật khẩu chỉ được sử dụng một lần duy nhất để xác thực một giao dịch hoặc một phiên làm việc. Do OTP chỉ được sử dụng một lần nên có độ an toàn cao hơn so với mật khẩu truyền thống, tránh được các dạng tấn công như nghe lén. Đồ án nghiên cứu về mật khẩu sử dụng một lần, các kỹ thuật sinh và chuyển giao mật khẩu một lần và ứng dụng vào xác thực các giao thức trực tuyến. Cụ thể, các nội dung đồ án đã thực hiện:

- Nghiên cứu tổng quan về mật khẩu sử dụng một lần và ứng dụng của mật khẩu sử dụng một lần.

- Nghiên cứu các phương pháp sinh mật khẩu sử dụng một lầndựa trên thời gian, dựa trên thuật toán; phương pháp sinh mật khẩu bằng Token và điện thoại di động; các phương pháp chuyển giao mật khẩu sử dụng một lần bằng giấy, bằng tin nhắn SMS.

- Đồ án đã cài đặt và thử nghiệm thành công ứng dụng mật khẩu sử dụng một lần trong xác thực giao dịch ngân hàng trực tuyến sử dụng phương pháp chuyển giao OTP thông qua tin nhắn SMS và sinh OTP trên điện thoại di động.