TIME-BASED ONE-TIME PASSWORD VERIFICATION SYSTEM

**SUBMITTED BY:**

**SHREYA CHINTAWAR - 2010030155**

**TEJASWI KOTTAKKI - 2010030088**

**HARIKA NETHI - 2010030119**

**AMRUTHA VARSHINI - 2010030009**

**ABSTRACT**

This document describes an extension of One-Time Password (OTP) algorithm, namely the HMAC based One Time Password (HOTP) algorithm to support the time-based moving factor. The HOTP algorithm specifies an event-based OTP algorithm, where the moving factor is an event counter. The present work bases the moving factor on a time value. A time-based variant of the OTP algorithm provides short-lived OTP values, which are desirable for enhanced security.

The proposed algorithm can be used across a wide range of network applications, from remote Virtual Private Network (VPN) access and Wi-Fi network logon to transaction-oriented Web applications. The authors believe that a common and shared algorithm will facilitate adoption of two-factor authentication on the Internet by enabling interoperability across commercial and open-source implementations.

**INTRODUCTION**

During the Covid-19 pandemic, almost all jobs and activities to be limited, relying on the internet network. Data security in an internet network is most important and needs to be a concern for internet users.

The internet network is a public network, which is vulnerable to security attacks. Attacks may occur to retrieve user data in the form of a username and password

OTP Verification is the process of verifying a user by sending a unique password so that the user can be verified before completing a registration or payment process.

Most of the time, we get an OTP when we make an online payment, or when we forget our password, or when creating an account on any online platform.

Thus, the sole purpose of an OTP is to verify the identity of a user by sending a unique password.

OTP is classified into two types : HOTP and TOTP

HMAC-based One-time Password algorithm (HOTP) is an event-based OTP where the moving factor in each code is based on counter.

Time-based One-time Password (TOTP) is a time-based OTP where the moving factor in each code is based on time.

Every HOTP code is valid until it’s used, or until a subsequent one is validated by the server. So the chances for hacking are high.

TOTP is basically a branch of HOTP and it has a huge advantage over HOTP.

Unlike with HOTP, OTPs are generated using the number of time steps from Unix time. Usually, the time step is set to either 30 or 60 secs.

So when considering TOTP vs HOTP the obvious choice is TOTP, simply because it is more secure.

**LITERATURE REVIEW**

**Title :**

TOTP : Time-Based One-Time Password Algorithm

**Author(s) :**

David M’Raihi , Salah Machani , Mingliang Pei , Johan Rydell .

**Reference link :**

[RFC 6238 - TOTP: Time-Based One-Time Password Algorithm (ietf.org)](https://datatracker.ietf.org/doc/html/rfc6238)

**Critical Assessment :**

Previous study :

The HOTP algorithm is based on an increasing counter value and a static symmetric key known only to the token and the validation service. In order to create the HOTP value, we will use the HMAC -SHA -1 algorithm.

As the output of the HMAC-SHA-1 calculation is 160 bits, we must truncate this value to something that can be easily entered by a user.

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

Where :

* Truncate represents the function that converts an HMAC-SHA-1 value into an HOTP value.

The Key (K), the Counter (c), and Data values are hashed high-order byte first.

Current study :

Basically, we define TOTP as TOTP = HOTP(K, T), where T is an integer and represents the number of time steps between the initial counter time T0 and the current Unix time.

More specifically, T = (Current Unix time – T0) / X, where the default floor function is used in the computation.

For example, with T0 = 0 and Time Step X = 30, T = 1 if the current Unix time is 59 seconds, and T = 2 if the current Unix time is 60 seconds.

The implementation of this algorithm must support a time value T larger than 32-bit integer when it is beyond the year 2038. The value of the system parameters X and T0 are pre- established during the provisioning process and communicated between a prover and verifier as part of the provisioning step.

**IMPLEMENTATION FLOW CHART**

Diagram

Description automatically generated

Diagram

Description automatically generated

**METHODS**

**ALGORITHMS FOR TIME-BASED ONE TIME PASSWORD VERIFICATION**

There are 3 Algorithms

**1. HOTP:**

* HOTP stands for Hash-Based OTP Verification.
* HOTP is HMAC based OTP algorithm, also referred to as event-based one-time Pass.
* HMAC Algorithm stands for Hash-Based Message Authentication Code.
* HOTP is a Counter Based OTP Verification.
* This algorithm takes some time to deliver a message with a one-time OTP and can be valid for a longer period.
* HOTP is least secured compared to other OTP generation algorithms.
* PROBLEMS:
* HOTP has Synchronization Problem.
* HOTP has Security Problem.

Diagram

Description automatically generated

**2. TOTP:**

* TOTP stands for Time-Based OTP Verification.
* TOTP is Token Based where the TOTP token has a secret key and the current time value.
* TOTP will take 30 sec to deliver a code or OTP and that OTP will be valid only for 30 sec, and after 30 sec the code will get expired.
* TOTP is more secure compared to HOTP.
* There are two parameters used to generate OTP using the TOTP Algorithm
* The Shared Secret (Unique code, generally 16-32 Base 32 character long).
* The current time interval (usually 30 or 60 sec).

A picture containing text, clock

Description automatically generated

**3. OCRA:**

* OCRA or OATH challenge-response algorithm is the most reliable multi-factor authentication algorithm.
* OCRA algorithm is proved to be the safest one created by the OATH (Open Authentication initiative) as it includes one-time passcode generation alongside the secret key and a counter or time.
* The challenge-response algorithm (OCRA) can be identified as advanced HOTP.
* OCRA algorithm expanded TOTP further by introducing the challenge-response mode to calculate OTP values.

Diagram

Description automatically generated

**ALPHA TESTING**

**Case 1:**

Generated OTP and Entered OTP are matched within the time limit (30 sec).

Text

Description automatically generated

**Case 2:**

Generated OTP and Entered OTP is Mismatched.

Text

Description automatically generated

**Case 3:**

Generated OTP and Entered OTP is Matched but Time Limit Exceeded (30 sec).

Text

Description automatically generated Text

Description automatically generated

**WORK PROGRESS**

**Text

Description automatically generatedText

Description automatically generated**

**RESULTS**

Time-based One Time Password or TOTP for short, does not utilize a counter for the server-user synchronization but generates a password based on the current time. The advantage of the TOTP password is a limited lifetime, usually 30-60 seconds.

TOTP algorithm is a branch of HOTP, TOTP is based on time and HOTP is based on Counter value.

The security and strength of the TOTP algorithm depend on the properties of the underlying building block HOTP, which is a construction based on HMAC using SHA-1 as the hash function.

One of the greatest advantages of TOTP is that it does not need any Internet connection for the OTP to get verified and it is easy to use across applications and channels.

**DISCUSSION**

The conclusion of the security analysis is that, for all practical purposes, the outputs of the dynamic truncation on distinct inputs are uniformly and independently distributed strings.

The analysis demonstrates that the best possible attack against the HOTP function is the brute force attack.

As indicated in the algorithm requirement section, keys should be chosen at random or using a cryptographically strong pseudorandom generator properly seeded with a random value.

Keys should be of the length of the HMAC output to facilitate interoperability.

All the communications should take place over a secure channel, e.g., Secure Socket Layer/Transport Layer Security (SSL/TLS) or IPsec connections.

It is recommended to store the keys securely in the validation system, and, more specifically, encrypt them using tamper-resistant hardware encryption and exposing them only when required: for example, the key is decrypted when needed to verify an OTP value and re-encrypted immediately to limit exposure in the RAM to a short period of time.

The key store must be in a secure area, to avoid, as much as possible, a direct attack on the validation system and secrets database. Particularly, access to the key material should be limited to programs and processes required by the validation system only.

The time-step size has an impact on both security and usability. A larger time-step size means a larger validity window for an OTP to be accepted by a validation system.

A larger time-step size exposes a large window to attack. When an OTP is generated and exposed to a third party before it is consumed, the third party can consume the OTP within the time-step window.

Therefore, it is recommended to use a default time step size of 30 secs. This default value of 30 secs is selected between security and usability.

**CONCLUSION**

TOTP, or Time-based OTP, is basically a branch of HOTP. And it has a huge advantage over HOTP — instead of the HOTP counter, TOTP tokens use time (UNIX time plus time-steps). Like with HOTP the user and server share a seed on setup. Unlike with HOTP — after that, the OTPs are generated using the number of time steps from the UNIX time. Usually, the time step is set to either 30 or 60 secs.

OATH HOTP-compatible tokens generate OTPs that do not have an expiration period. And we have already come to the conclusion that this creates a major security vulnerability. TOTP passcodes, on the other hand, have the advantage of being valid for a limited time period — the time step. So if the generated pass is not used within the 30-60 seconds it expires and cannot be used for login.

So when considering TOTP vs HOTP the obvious choice is TOTP, simply because it is more secure.