Time-based One-time Passwords (TOTP)

What are Time-based One-time Passwords (TOTP) and how do they work?

If two users want to communicate securely in cyberspace they are able to use strong encryption such as AES to send and receive messages. The communication itself is very hard to attack for a third party. However, it requires a lot less effort to infiltrate a computer and „steal" the identity of the user in order to gain information about the communication partner. 1

This problem requires a system, which validates the identity not only by a password, but also a different factor. 2
A time-based implementation ensures, that even if the attacker is able to infiltrate and possibly decipher the passwords, he is not able to intercept the communication between the parties. For these demands the Internet Engineering Taskforce created the Time-based One-time password (TOTP) system, based on modern cryptology.

The concept behind standardized TOTP system consists of several steps in order to ensure the security and the identity of the clients

3 The two clients agree on a shared key, which is exchanged via a secure channel once.
4 They synchronize their clocks to the exact same time.
5 The key, they just exchanged, plus the time add up to a temporary key called one-time password (OTP).
6 When communication is required clients have to use their password + the TOTP to proof their identity.

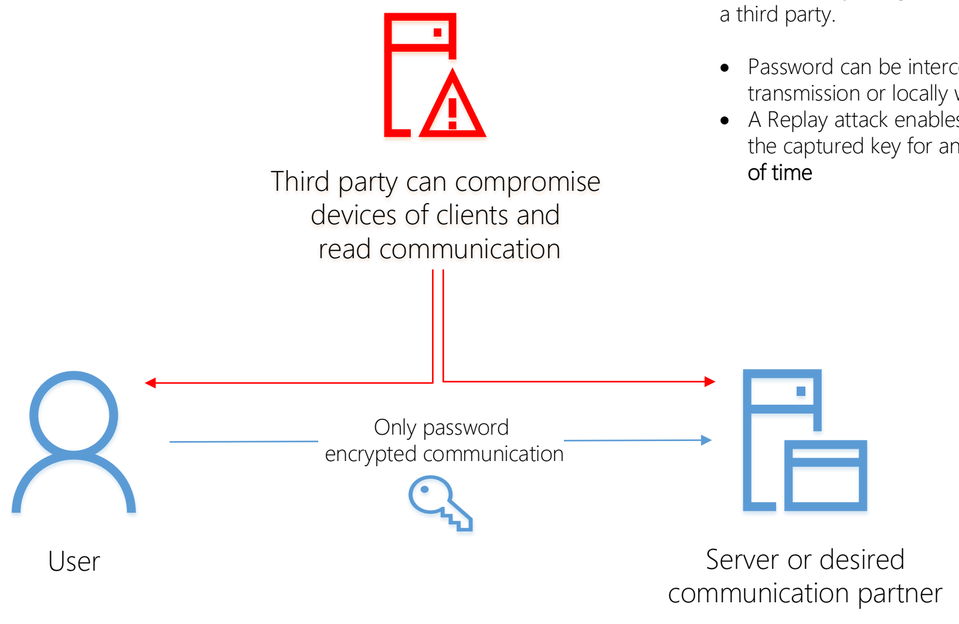
When dealing with cryptology we also have to take possible attacks into account, so it is important to question the security of the system. 7

For further information please refer to my annotated bibliography. 8

1 What are the weaknesses of a password based system?

Simple only password based security has weaknesses by design and can be attacked by a third party.

- Password can be intercepted during the transmission or locally with a keylogger
- A Replay attack enables the attacker to use the captured key for an **unlimited amount of time**



2 What is Multi-factor authentication?

Multi-factor authentication is a system to verify a users identity by two or more independent factors

The factors can be divided in three different categories:

- Hardware, which is in possession of the user (e.g. a smartphone)
- A Password known by the user
- A physical part of the user (e.g. fingerprints)

At least two of these factors combined are used in the system in order to increase security

Form of Token	Example
Paper Token	Tans for online-banking
Software Token	Google Authenticator App
Hardware Token	Yubikey, RSA Key Generator

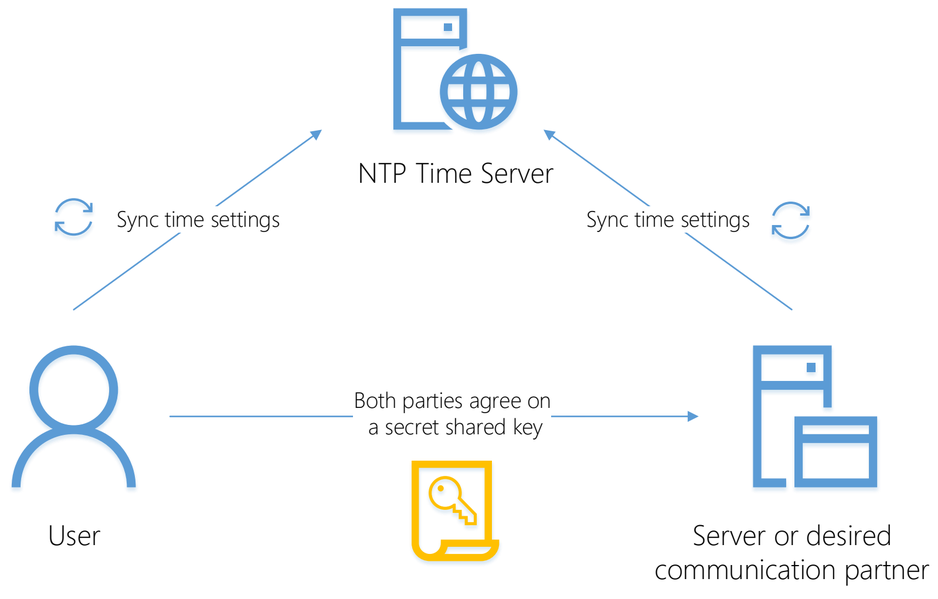
3 Secret key

The two communication partners need to agree on agree on a shared secret key, which is exchanged before the TOTP is implemented.

This can be done by programming the hardware token or sharing the key with the software token (e.g. QR Code)

4 Time Synchronization

The synchronization of the exact time between the two communication partners is critical for the TOTP to function correctly. Both sides need a correct time to generate a one-time pad, so they agree on a Network Time Protocol (NTP) to request an accurate time.
Note that not the actual time is used for the equation, but the seconds elapsed since the first digital clock was online on the 1th of January 1970 at 00:00



5 Mathematical specification of the authentication

The TOTP system is based on Hash-based message authentication code (HMAC), which uses the SHA-1 hash function to generate a random output, fixed to 160 bit. The HMAC function is displayed by

$$HMAC_k(M) = S((K \oplus opad), S((K \oplus ipad), M))$$

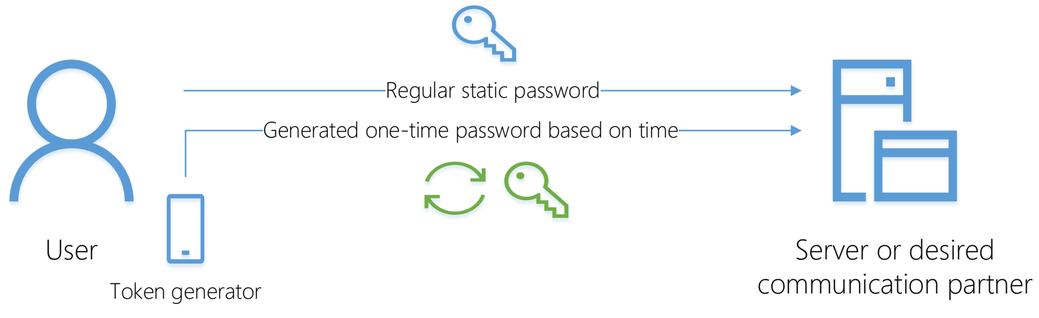
Where **M** is the message, **K** the shared key and **S** is the SHA-1 hash function. *opad* and *ipad* are rotating constants, which are added to K with an *Xor* operation.

The system then combines the shared secret and the clock of the computer. The time is used as a moving factor, changing after a defined amount time has passed.

$$HMAC\left(K, \frac{Time(x)}{P}\right) = TOTP \bmod 10^L$$

The two input factors into the cipher are defined by **K** as the shared key as the first input, the time passed since 1970 as **X**, divided by the amount of time we want the generated key to be active, displayed as **P**. The modulus 10 correlates to the SHA-1 Hash, where **L** defines the length of the key. Both parties should generate the same key and therefore are able to authenticate the signature.

6



7 Possible disadvantages of TOTP

It is vital to question, if TOTP can be broken or bypassed by an attacker

- If a client looses the token generator, he won't be able to login anymore
- If a client uses the same device to login and as key generator then the added security is lost.
- An attacker could try to spoof the TOTP login page and then has a short amount of time to authenticate with the other party himself
- The shared key could be intercepted when setting up the system

It is important to mention that the descripted attacks are theoretical and not possible if TOTP is correctly implemented

8 Conclusion and further information

TOTP makes the verification of clients much more secure and can help to harden systems against possible interceptions. Many major internet companies, such as Google, Microsoft and Facebook introduced the system to their services. To this day TOTP remains one of the most effective layer when communicating online.

If you are further interested in the topic and want to see my bibliography please scan the QR Code.