SECURITY OVERVIEW

February 2016

Version 1.2

Product Information

Organisation	Token One Pty Ltd
Product Name	TokenOne Authentication
Document Owner	Amanda Hatton

Document History

Version	Date	Author	Revision Description
0.1	2013-09-14	Mark Bugno	Initial outline with various high-level sections filled in, and other sections containing examples
0.2	2013-09-24	Mark Bugno	Incorporated changes recommended by PC & KK. Fleshed out other sections.
0.3	2013-10-17	Mark Bugno	Incorporated changes recommended by PC & KK. Polish wording.
0.4	2013-11-05	Mark Bugno	Corrected minor issues and incorporated changes recommended by PC & SE-M
0.5	2015-05-14	Amanda Hatton	Updated formatting
1.0	2015-05-25	Amanda Hatton	Release version
1.1	2015-11-05	Amanda Hatton	Update for new style guidelines
1.2	2016-02-17	Maarit Linsbauer	Co-branding for partner

Copyright ©2016 Token One Pty Ltd. All rights reserved.

For information about TokenOne trademarks, copyrights and patents refer to the TokenOne Intellectual Property Page on the TokenOne website. All other trademarks and registered trademarks are the property of their respective holders.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Token One Pty Ltd. Under the law, reproducing includes translating into another language or format. Every effort has been made to ensure that the information in this document is accurate. Token One Pty Ltd. is not responsible for printing or clerical errors. Information in this document is subject to change without notice.

TokenOne Security Overview, version 1.2 February 17, 2016

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide detailed analysis of the theory that underlies TokenOne Authentication, especially regarding security.

Where possible, the impact of security features will be quantified, and comparisons made to alternate offerings. An investigation of attack vectors, along with the mitigations provided by the TokenOne solution, is also detailed in this document.

1.2 Audience

This document is intended for security experts, software architects, and decision makers. It is suitable for anyone with at least a rudimentary understanding of cryptography and probability.

2 Background and Theory

2.1 Introduction

TokenOne is an online identity assurance system that provides cross cross-domain identity verification and authentication to integrated systems. This document focuses on the analysis of TokenOne's patented identity authentication technology: TokenOne Authentication.

TokenOne Authentication is implemented as a two-factor identity verification component solution that targets human-computer authentication. While this may loosely be described as online authentication, TokenOne Authentication is suitable for many scenarios that may not be perceived as "online" (e.g. securing ATM withdrawals; securing a door lock; etc.).

2.2 Problem Space

It is generally accepted that there are three factors (properties) for humans to authenticate themselves:

- **Knowledge Factor:** Something only the person *knows*, such as a password or PIN
- **Possession Factor:** Something only the person *has*, such as an identity card or token
- **Inherence Factor:** Something only the person *is*, such as an iris scan or voiceprint

Most human-computer authentication is single-factor – it requires only one of the above, most often a knowledge factor. This type of authentication provides only limited security. It is possible for:

- A person to share their password, or
- An attacker to observe the person entering it.

When a person enters a password, there are no checks performed to ensure that it is the person who should be using that password. Two-factor authentication, as provided by TokenOne Authentication, helps mitigate this risk, making it significantly harder for would-be attackers to compromise the security of systems.

Many online services that store confidential or valuable data now use two-factor authentication. Such services include banks (e.g. Commonwealth Bank of Australia) and email providers (e.g. Gmail). While there are a number of products on the market aimed at providing two-factor authentication, all have significant drawbacks compared to TokenOne Authenticaiton. While a number of other authentication solutions use phrases like "two-step verification" in an attempt to sound secure, TokenOne Authentication is a **genuine** strong two-factor authentication solution.

TokenOne Authentication has been engineered to support both out-of-band and in-band use. §**3.2.7** defines the risks associated with in-band use, along with steps that may be taken to mitigate these vulnerabilities. In the absence of these mitigations, out-of-band usage provides the strongest security.

2.3 Encipherment

Encipherment is a form of encryption where the plaintext of a given length is converted into ciphertext of the same length, by means of an algorithm. While the plaintext is human-readable, the ciphertext is meaningless until deciphered. Decipherment requires the application of the inverse function of the algorithm applied to encipher the plaintext:

$$f^{-1}(f(x)) = x$$

Where:

x	is the plaintext
f	is the encipherment algorithm
f(x)	is the ciphertext and
f^{-1}	is the decipherment algorithm.

Typically, the encipherment and decipherment algorithms require an additional piece of information: a cryptovariable. A cryptovariable is a shared secret, sometimes called a key. Its function is twofold:

1. It provides security for the ciphertext, even if the algorithm is discovered or known
2. It enables the algorithm to be reused, without the compromise of one use affecting the security of other uses

TokenOne Authentication uses a polyalphabetic substitution cipher, which is similar in nature to a one-time pad.

2.4 One-Time Pads

A one-time pad is both a cryptographic technique and cryptographic tool for encipherment. Provided it is implemented correctly, the ciphertext resulting from a one-time pad is impossible to cryptanalyse, so impossible to crack.

As a tool, each sheet of a one-time pad is typically comprised of a series of random characters; these form the cryptovariable. As a technique, the one-time pad acts as a stream cipher: encipherment involves modular addition of the message with the cryptovariable on a character-by-character index basis. For example, if the message *TOKENONE* was to be sent, and cryptovariable began *PSXOYEPA…*, then the encipherment may take the following form:

T	**O**	**K**	**E**	**N**	**O**	**N**	**E**	*Plaintext*
19	14	10	4	13	14	13	4	*Plaintext Character Indices*
P	S	X	O	Y	E	P	A	*Cryptovariable*
15	18	23	14	24	4	15	0	*Cryptovariable Character Indices*
34	32	33	18	37	18	28	4	*Plaintext + Cryptovariable*
8	6	7	18	11	18	2	4	*(Plaintext + Cryptovariable) % 26*
I	**G**	**H**	**S**	**L**	**S**	**C**	**E**	*Ciphertext*

Using the above algorithm, the ciphertext would be: **IGHSLSBE.** A few things should be noted here:

- Two characters that are identical in the plaintext may appear different in the ciphertext (e.g. O→G, O→S)
- Two characters that are different in the plaintext may appear identical in the ciphertext (e.g. E→S, O→S)
- Characters may appear the same in both the plaintext and ciphertext (e.g. E→E)

To decipher the ciphertext, the inverse function must be applied. This is effectively modular subtraction; as the modulus is the length of the character set, each enciphered character will map to exactly one plaintext character. For example:

I	**G**	**H**	**S**	**L**	**S**	**C**	**E**	*Ciphertext*
8	6	7	18	11	18	2	4	*Ciphertext Character Indices*
P	S	X	O	Y	E	P	A	*Cryptovariable*
15	18	23	14	24	4	15	0	*Cryptovariable Character Indices*
-7	-12	-16	4	-13	14	-13	4	*Ciphertext – Cryptovariable*
19	14	10	4	13	14	13	4	*(Ciphertext – Cryptovariable) % 26*
T	**O**	**K**	**E**	**N**	**O**	**N**	**E**	*Plaintext*

As the same cryptovariable is used to both encipher and decipher the message, one-time pads can be described as symmetrical ciphers.

For a one-time pad to be secure, two things must hold:

1. The cryptovariable must be random. If a pattern exists in the cryptovariable, it can be exploited to cryptanalyse the message.
2. The cryptovariable must be at least as long as the plaintext. If the cryptovariable is shorter than the plaintext, it will typically be used cyclically; this cycle will form a pattern, making the ciphertext susceptible to cryptanalysis.

If an attacker intercepted the ciphertext from the example above, they would not be able to decipher it without the cryptovariable. As there are no patterns in the cryptovariable, there is no additional information for the attacker to exploit. As such, any potential key that yielded valid plaintext would be just as likely of a candidate as any other. For example, if the attacker tried to decipher the ciphertext with the cryptovariable **FSLFGSRT**, they would get:

I	**G**	**H**	**S**	**L**	**S**	**C**	**E**	*Ciphertext*
8	6	7	18	11	18	2	4	*Ciphertext Character Indices*
F	S	L	F	G	S	R	T	*Cryptovariable*
5	18	11	5	6	18	17	19	*Cryptovariable Character Indices*
3	-12	-4	13	5	0	-15	-15	*Ciphertext – Cryptovariable*
3	14	22	13	5	0	11	11	*(Ciphertext – Cryptovariable) % 26*
D	**O**	**W**	**N**	**F**	**A**	**L**	**L**	*Plaintext*

Exhausting all possible 8-character cryptovariables, the attacker would yield all possible 8-character combinations of the letters A to Z. Even if the attackers suspected the message was one or more English words, the list of potential plaintext results would contain all single and combined English words with a total of 8 characters. Without knowing the context of the message, it is impossible for the attacker to guess the actual plaintext.

While the TokenOne Authentication encipherment technology is different in application (i.e. does not require modular arithmetic), it is nearly identical to a one-time pad in effect (i.e. cryptovariables are random, longer than the enciphered text, and not reused).

3 Implementation

3.1 Overview

TokenOne Authentication is an online identity assurance system that provides cross-domain identity verification and authentication to integrated systems. Flexible APIs enable integration between TokenOne Authentication and any compatible application or service. TokenOne Authentication can also be extended to integrate with existing security standards such as SAML.

Users authenticate with TokenOne Authentication using a TokenOne PIN. To determine the TokenOne PIN, both the User and User's smart device must be present at the moment of authentication. As the User's PIN is never revealed (at entry or to participating systems) a single PIN can be safely used across all authorised accounts. This provides a simple and more secure alternative to either conventional passwords or one-time passwords generated by hardware tokens, smart device apps, or sent by SMS.

For a registered User to authenticate with TokenOne Authentication, they need **both** knowledge and possession factors. When authenticated, TokenOne provides single sign-on functionality, enabling the User to access integrated systems and services with their appropriate levels of access. Simultaneously, TokenOne Authentication provides a mechanism for escalation of services. What differentiates TokenOne Authentication from the competition is the added transformative security that its OTP (one-time pad) - like implementation provides.

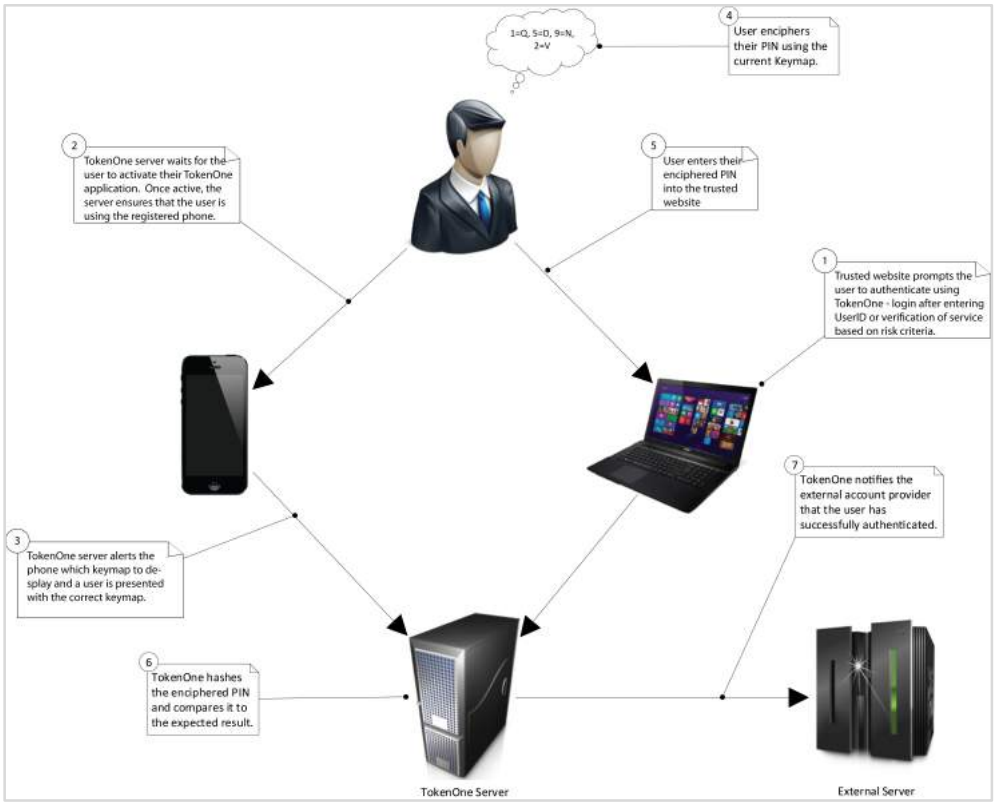

Figure 1 – Process of TokenOne Authentication service escalation

At registration, a User self-creates a PIN, which acts as a knowledge factor for future authentications. What is unique about the TokenOne Authentication implementation is that not even TokenOne (let alone the service to which the user is authenticating) knows the User's PIN, and the User never actually enters their PIN. It is impossible, from a cryptographic perspective, for an attacker to ever compromise the User's PIN without either socially engineering it from the User, or compromising multiple devices. Caveats apply for in-band authentication (see **§3.2.7**).

TokenOne Authentication requires the User to have a smart device (such as an iPhone), and access to a terminal (such as a computer). The User can use multiple terminals, however, they must always have their smart device with them: The User's smart device is their possession factor. In addition to a digital certificate, unique properties of the smart device are used to verify its presence. This increases the complexity of spoofing attacks required to compromise the device identification process. TokenOne aims to make the User's smart device a true token.

When the User authenticates with TokenOne Authentication, they first activate a KeyMap via the TokenOne app on their smart device. The KeyMap acts as a one-time cryptovariable for the authentication. The User uses the KeyMap to encipher their PIN, but is never required to enter or reveal their PIN in the process. The User then uses their out-of-band terminal to log into TokenOne Authentication, providing the enciphered PIN as their knowledge factor. Again, this does not require the User to enter or reveal their actual PIN.

If the User's smart device (ergo cryptovariable) and terminal (ergo ciphertext) are on different bands, the User's PIN (i.e. plaintext) remains secure even if an attacker compromises one of the devices.

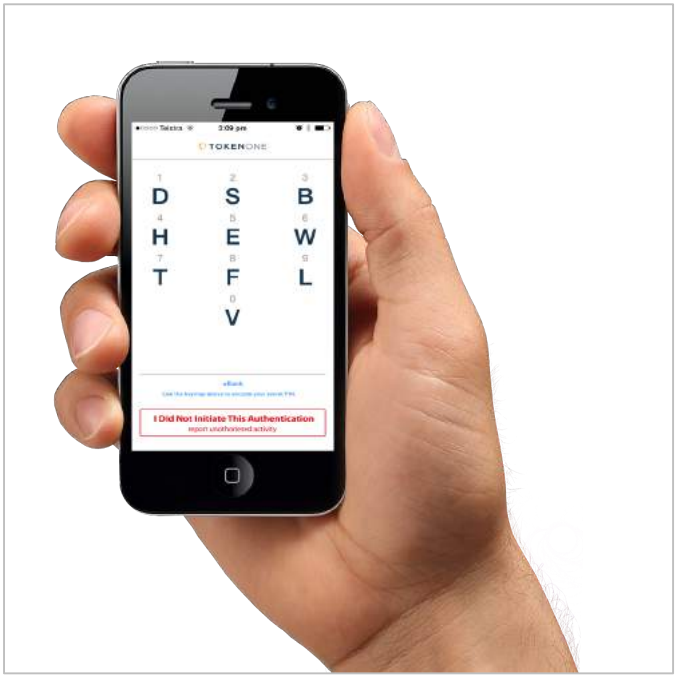

Figure 2 – A KeyMap used to encipher a User's PIN

TokenOne KeyMaps are substitution ciphers that effectively act as one-time pads. For the convenience and clarity of the customer, TokenOne Authentication prevents a single plaintext digit from mapping to multiple ciphertext characters, and multiple plaintext digits mapping to a single ciphertext character. The security ramifications of this are contained and minimised by enforcing uniqueness of all digits in the User's PIN. While this reduces the number of possible enciphered PINs, making password attacks easier; the reduction is more than overcome by the increase due to the 10-digit PIN-space being mapped to the 26-alpha-character enciphered-PIN-space.

Number of potential 4-digit PINs with repeating digits:

$$N = 10^4$$
$$= 10000$$

Number of potential 4-digit PINs with non-repeating digits:

$$N = \frac{10!}{6!}$$
$$= 5040$$

Number of potential 4-character enciphered PINs, for PINs with non-repeating digits:

$$N = \frac{26!}{22!}$$
$$= 358800$$

As illustrated above, a TokenOne PIN is over 35 times more secure (less susceptible to a password attack) than a standard ATM PIN. As the generation of TokenOne KeyMaps is not tied to an algorithm, the KeyMaps themselves are not vulnerable to algorithmic hacks.

TokenOne Authentication can be modified to support single-plaintext-digit to multiple-ciphertext-character, and multiple-plaintext-digit to single-ciphertext-character substitution. To maximise speed and simplicity for Users, this is not the default option currently, but doing so would make TokenOne Authentication a genuine one-time pad. Allowing multiple plaintext digits to map to the same ciphertext character (irrespective of the implementation of the other option) would increase the security of the ciphertext to the same level as that of a one-time pad. The permutations of an enciphered n-digit PIN would increase from:

$$P(n) = \frac{26!}{(26-n)!}$$

To:

$$P(n) = 26^n$$

In addition to the security provided by TokenOne Authentication not storing Users' PINs, TokenOne Authentication also does not store cryptovariables (KeyMaps) or ciphertext (enciphered PINs).

When the User registers their TokenOne Account they must create their own PIN. TokenOne Authentication provides the User with two different KeyMaps, each of which has a unique identifier, and the User supplies TokenOne Authentication with the encipherment of their PIN using each KeyMap.

TokenOne Authentication uses the ordinals of the User's enciphered PINs to momentarily determine the User's PIN. TokenOne Authentication then generates a series of KeyMaps for the User to use in later authentications; each KeyMap has a unique identifier. TokenOne Authentication determines the enciphered PIN for each KeyMap, then encrypts (salts and hashes) the enciphered PIN, and stores the hash of the encrypted enciphered PIN against the unique identifier for the KeyMap.

The hashing algorithm that TokenOne Authentication uses is one-way, meaning that even with the cryptovariable used in the hash, the hash cannot be decrypted to its original form. This secures Users' enciphered PINs in the event that the TokenOne Servers were compromised.

Even if an attacker were to gain unauthorised access to the TokenOne Servers, break into the database, and steal the hashes of Users' enciphered PINs, they still would not be able to decrypt them. This does not prevent attacks where the attacker attempts to guess the encryption key, however, by using unique salts but TokenOne Authentication makes such an attack unviable as it is too slow. Further, acquiring a User's enciphered PINs is not sufficient to hijack the User's TokenOne Account. Each enciphered PIN may only be used in a brief window after its corresponding KeyMap is activated, and the User's smart device is required to activate the KeyMap.

When the User activates a KeyMap, TokenOne Authentication anticipates its use for a short period of time. If the User enters an enciphered PIN during that period, TokenOne Authentication encrypts the enciphered PIN using the same salt and hash algorithm used when generating the stored value. TokenOne Authentication compares the hashes of the encrypted values, and, if they match, authenticates the User.

3.2 Security Measures

This section details the security measures that have been proactively taken to protect TokenOne Authentication and Users' identities. This section only focuses on programmatically-implemented security measures involved in the authentication process. All quantifiable security measures are included.

> **Note:** These figures do not do justice to TokenOne Authentication: the required presence of the User's smart device makes TokenOne Authentication significantly more secure than the mathematics indicates

For a full list of security measure implemented in TokenOne Authentication, see the **Solution Architecture** document.

3.2.1 Numeric-to-Alphabetic Mapping

While the User's PIN is a numeric sequence (digits 0-9), it is enciphered to an alphabetic sequence (letters A-Z). Without an attacker obtaining the KeyMap (which would require them to compromise the smart device), they will not know which 10 (of the 26) letters are mapped. The numeric-to-alpha key mapping contributes 98.6% of the overall security:

$$
\begin{aligned}
N_1 &= \frac{10!}{6!} \\
&= 5.04\times10^3 \\
N_2 &= \frac{26!}{22!} \\
&\approx 3.59\times10^5 \\
C_{\%} &= \frac{N_2 - N_1}{N_2} \\
&\approx 98.6\%
\end{aligned}
$$

3.2.2 PIN Length

By default, a Users' PIN must be 4 digits. This length was chosen, as Users are familiar with the standard 4-digit ATM PIN used by banks. It is possible to enforce longer minimum and maximum lengths.

While limiting the number of potential permutations (3-digit, 2-digit and 1-digit PIN), the excluded sequences are too obvious and would form prime candidates for attack. The number of potential PIN permutations is only reduced by 4.34%:

$$\begin{aligned} N_1 &= \sum_{d=1}^{4} \frac{26!}{(26-d)!} \\ &\approx 3.75\times10^5 \\ N_2 &= \sum_{d=4}^{4} \frac{26!}{(26-d)!} \\ &\approx 3.59\times10^5 \\ R_{\%} &= \frac{N_1 - N_2}{N_1} \\ &\approx 4.34\% \end{aligned}$$

The strength of the weakest PIN is increased by a factor of 13 800:

$$\begin{aligned} P(x)_1 &= 26^{-1} \\ &\approx 3.84\times10^{-2} \\ P(x)_2 &= \left(\frac{26!}{22!}\right)^{-1} \\ &\approx 2.79\times10^{-6} \\ I &= \frac{P(x)_1}{P(x)_2} \\ &= 1.38\times10^4 \end{aligned}$$

3.2.3 Value of a Longer PIN

Although longer PINs provide more security from password attacks; this must be weighed against the added challenge for the User to remember their PIN. Each additional digit added to a PIN, to a maximum of 10 digits, will decrease the probability of a successful password attack. The increase in security though exhibits diminishing returns for each additional digit added. Further, PINs that are too long may result in the User writing them down or choosing an easy to remember (ergo obvious) sequence. Encouraging such actions diminishes overall security.

The value of each additional digit in preventing a password attack is listed below. Setting a PIN length of more than 10 digits does not further increase the security of the PIN as the *same plaintext character* is enciphered as the *same ciphertext character* for a given KeyMap.

# Digits	# Potential Permutations	Value of Last Digit to Overall Security
1	2.60×10^{1}	100%
2	6.50×10^{2}	96.0%
3	1.56×10^{4}	95.8%
4	3.59×10^{5}	95.7%
5	7.89×10^{6}	95.5%
6	1.66×10^{8}	95.2%
7	3.32×10^{9}	95.0%
8	6.30×10^{10}	94.7%
9	1.13×10^{12}	94.4%
10	1.93×10^{13}	94.1%

3.2.4 PIN Unique Digits

Each digit of the PIN must be unique. While reducing the number of potential PIN permutations, if two ordinals contained the same digit, the two ordinals would also share the same character when enciphered. This would effectively render the PIN one digit shorter. The impact on security of this reduction in permutations is negligible, however, it greatly increases the minimum PIN strength and reduces the variance in PIN strengths.

The enforcement of unique digits reduces the number of potential four-digit PIN permutations by 49.6%:

$$\begin{aligned} N_1 &= 10^4 \\ &= 1.00{\times}10^4 \\ N_2 &= \frac{10!}{6!} \\ &= 5.04{\times}10^3 \\ R_{\%} &= \frac{N_1 - N_2}{N_1} \\ &= 49.6\% \end{aligned}$$

However, the number of potential enciphered four-digit PIN permutations is only reduced by 21.5%:

$$\begin{aligned} N_1 &= 26^4 \\ &\approx 4.57{\times}10^5 \\ N_2 &= \frac{26!}{22!} \end{aligned}$$

$$\approx 3.59\times10^{5}$$

$$R_{\%} = \frac{N_1 - N_2}{N_1}$$

$$\approx 21.5\%$$

Further, as repeated characters in the enciphered PIN represent repeated digits in the PIN, the reduction in enciphered PIN permutations is only 4.34%:

$$N_1 = \sum_{d=1}^{4} \frac{26!}{(26-d)!}$$

$$\approx 3.75\times10^{5}$$

$$N_2 = \sum_{d=4}^{4} \frac{26!}{(26-d)!}$$

$$\approx 3.59\times10^{5}$$

$$R_{\%} = \frac{N_1 - N_2}{N_1}$$

$$\approx 4.34\%$$

Adding this rule, decreases the probability of an attacker guessing the weakest PIN on their first attempt by a factor of 13 800:

$$P(x)_1 = 26^{-1}$$

$$\approx 3.84\times10^{-2}$$

$$P(x)_2 = \left(\frac{26!}{22!}\right)^{-1}$$

$$\approx 2.79\times10^{-6}$$

$$I = \frac{P(x)_1}{P(x)_2}$$

$$= 1.38\times10^{4}$$

3.2.5 PIN Non-Sequential

PINs cannot be solely comprised of a sequence of consecutive numbers (either ascending or descending). While limiting the number of potential permutations, these sequences are too obvious and would form prime candidates for attack. Further, the reduction in potential permutations is negligible:

$$R_{\%} = \frac{14}{3.59\times10^{5}}$$

$$\approx 3.90\times10^{-3}\%$$

Dictionary attacks are one of the simplest (and often first deployed) attacks in a hacker's arsenal. So it is fair to use the percentage of numbers excluded to prevent dictionary attacks as an approximation of the percentage of total permutation-space a hacker would need to cover to crack a PIN. So the exclusion of these sequential PINs can be estimated to improve the strength of the weakest PIN by a factor of 25 600. That means with the likely PIN candidates removed, it will take an attacker approximately 25 600 times as long to have a 50% probability of guessing the weakest PIN.

$$I = {R_{\%}}^{-1}$$
$$\approx 2.56\times10^{4}$$

3.2.6 PIN Non-Birthdate

Where 6-digits PINs are used, a User's PIN cannot be their date of birth in any of the formats:

- yyyyMM
- MMyyyy
- ddMMyy
- MMddyy
- yyMMdd

Whilst limiting the number of potential permutations, these sequences are too obvious and would form prime candidates for attack. Further, the reduction in potential permutations is negligible:

$$R_{\%} = \frac{5}{1.66\times10^{8}}$$
$$\approx 3.02\times10^{-6}\%$$

As the User's date of birth is stored in TokenOne Authentication for identity verification purposes, any attacker who managed to compromise the TokenOne database would have access to it. Even without access to the TokenOne database, if an attacker is performing a targeted attack, a basic level of doxing will provide the User's date of birth. As date of birth is a prime candidate for PINs, and is likely known to the attacker, it should not be permitted as a PIN.

Making the same assumptions as for dictionary attacks, the exclusion of date of birth PINs can be estimated to improve the strength of the weakest PIN by a factor of 33 200 000.

$$I = {R_{\%}}^{-1}$$
$$\approx 3.32\times10^{7}$$

While TokenOne Authentication does not store the User's PIN, it is momentarily aware of it at the point of KeyMap generation. The check of its validity is performed at this point.

3.2.7 Out-of-Band vs. In-Band KeyMaps

When transferring data, a transfer is said to be out-of-band if it occurs over a medium that is not the standard/default medium. In the context of TokenOne Authentication, out-of-band data transfers are when the User's smart device does not share an Internet connection with the User's terminal.

The opposite of out-of-band is in-band. In the context of TokenOne Authentication, in-band data transfers are when one of the following is true:

- The terminal is the smart device
- The terminal is connected to the Internet via the smart device (a wireless hotspot or tethering).
- The terminal and smart device are connected to the Internet via the same connection (e.g. they are both using the same Wi-Fi connection).

For optimal security, it is recommended that out-of-band communication be used. By enforcing out-of-band communication, the User's KeyMaps (i.e. cryptovariables) and enciphered PIN (i.e. ciphertext) are never sent/activated over the same Internet connection. The sophistication of a potentially successful attack would need to increase greatly.

3.2.7.1 In-band Attacks

For a man-in-the-middle attack to be successful, when using in-band communication, the attacker would need to compromise the User's connection, then steal or crack the User's SSL certificates.

If performed prior to the User creating their TokenOne Account, this would give the attacker access to the User's KeyMaps and ciphertext. As a result, the User's security would be completely compromised: the attacker could work out their PIN, activate KeyMaps on their behalf, and continue to identify as them.

If performed after the User created their TokenOne Account, this attack would only allow the attacker to see which KeyMap is activated and the corresponding enciphered PIN. The attacker would still need to compromise the TokenOne Server or User's smart device to get a list of the actual KeyMaps.

- The sophistication of the required attack will be lowered if the User is using Wi-Fi to connect their smart device to the Internet, as a Wi-Fi connection is easier to sniff (conduct packet inspections) and a router easier to compromise, than a cellular connection (e.g. LTE).
- The sophistication of the required attack will be lowered if the User is using their smart device as the terminal. This would allow the attacker to compromise only the smart device and gain access to the certificates for both connections.

- The sophistication of the required attack will be lowered if offline KeyMap activation is supported (see **§3.2.8**), as the attacker will not need to spoof the User's possession factor to activate a KeyMap.

3.2.7.2 Out –of-band Attacks

For a man-in-the-middle attack to be successful when using out-of-band communication, the attacker would additionally need to compromise the User's other connection. The sophistication of this attack would need to be very high where the smart device is connected directly to the Internet via its cellular connection.

By request, TokenOne Authentication can be configured to enforce out-of-band communications. When out-of-band communications are enforced, the User will not be able to log in to their TokenOne Account from the same IP address that they use to activate their KeyMap.

Where in-band communication is supported, it is strongly recommended that additional steps be put in place to prevent man-in-the-middle attacks. Mitigations against attack would include:

- Recommend users never use untrusted or unsecured Wi-Fi
- Enforce online KeyMap activation
- Strengthening encryption for the mutual handshake
- Examine the latency of connection in respect to typical latency

3.2.8 Online vs. Offline KeyMap Activation

When a User wants to authenticate with TokenOne Authentication, they must use a KeyMap to encipher their PIN. Selecting this KeyMap on their smart device activates the KeyMap – a message is then sent from the smart device to the TokenOne Server, with the serial number of the KeyMap to be activated.

When the TokenOne Server receives this message it allows the User's use of this KeyMap for a short period of time. Once that period of time elapses, the KeyMap is locked out and rendered void. This reduces the opportunity for attackers to attempt to impersonate the User, as they can only make attempts in this window. Without observing the User (physically or digitally), an attacker would be unaware of when a KeyMap was active.

As KeyMaps are stored on the User's smart device, the smart device theoretically does not require an Internet connection at the time of the user's authentication via a terminal (provided the smart device does not need to send the KeyMap activation signal).

If you enable offline KeyMap activation this reduces the certainty that the activator is in fact the User. This method reduces the need of the activator to be in possession of the User's token (i.e. smart device).

When TokenOne Authentication is configured to support offline KeyMap activation, it takes additional steps to mitigate this introduced risk.

In offline mode, when the User attempts to authenticate with TokenOne Authentication, and TokenOne Authentication cannot establish a connection to the User's smart device, it prompts the User with a KeyMap identifier at the terminal. The User then uses their smart device to select the KeyMap with matching identifier. Despite the smart device having no connection to the TokenOne Server, the server pre-activated this KeyMap when they presented the User with the prompt. So the window that the User can use the appropriate KeyMap started diminishing from shortly before the point where they were prompted.

The remainder of the authentication process is identical to the standard TokenOne Authentication process: The User uses the appropriate KeyMap to encipher their PIN, and then enters the enciphered PIN at the terminal. If the hash of the encrypted enciphered PIN matches the hash on record, the User is granted access to their TokenOne Account.

Despite this increased security, offline KeyMap activation mode does not guarantee the presence of the User's token (smart device). If an attacker had the User's KeyMaps and PIN, they could use offline Activation mode and not need further access to the User's smart device. As TokenOne Authentication attempts to establish a connection to the User's smart device, this of course would need to be a well-timed attack or be more sophisticated (such as simultaneously jamming the User's smart device).

For optimal security it is recommended that TokenOne Authentication be configured to require online KeyMap activation.

> **Note:** Offline KeyMap Activation is only available by request.

3.3 Attack Vectors and Mitigation

3.3.1 Overview

This section details attack vectors that hackers may use in their attempt to exploit TokenOne Authentication or Users' identities. In each case, where measures have been taken to mitigate the vector, details of the mitigation are provided. For each attack vector, the full impacts of a successful attack are listed, and the flow-on ramifications of these impacts are detailed.

> **Note:** While TokenOne Authentication prevents numerous remote attacks, it is neither designed nor intended to prevent vulnerabilities in the underlying hardware, nor localised 'gun to the head' type attacks

3.3.1.1 Summary

The risk assessment of each attack vector is intended to aid in the prioritisation of security and training efforts. Likelihood only considers the chance of attack, not the chance of success. Applying strong security practices, providing thorough training and remaining vigilant will significantly reduce the likelihood of attacks succeeding.

Attack	Likelihood of Attack	Potential Impact of Success	Risk
Social Engineering	Probable	Catastrophic	2 – Very High
Compromise (TokenOne Sever)	Occasional	Catastrophic	4 – Very High
Key Logging (Terminal)	Frequent	Major	**7 – High**
Man-in-the-Middle Attack (Terminal to TokenOne Server)	Occasional	Major	11 - Medium
Theft (Smart Device)	Frequent	Minor	13 – Medium
Spoofing (TokenOne Server)	Remote	Major	14 – Medium
Physical Monitoring	Probable	Minor	16 – Medium
Man-in-the-Middle Attack (Smart Device to TokenOne Server)	Occasional	Minor	**18 – Low**
Spoofing (Smart Device)	Occasional	Minor	**18 – Low**
Memory Sniffing (Smart Device)	Remote	Minor	**19 – Low**

3.3.1.2 Risk Index

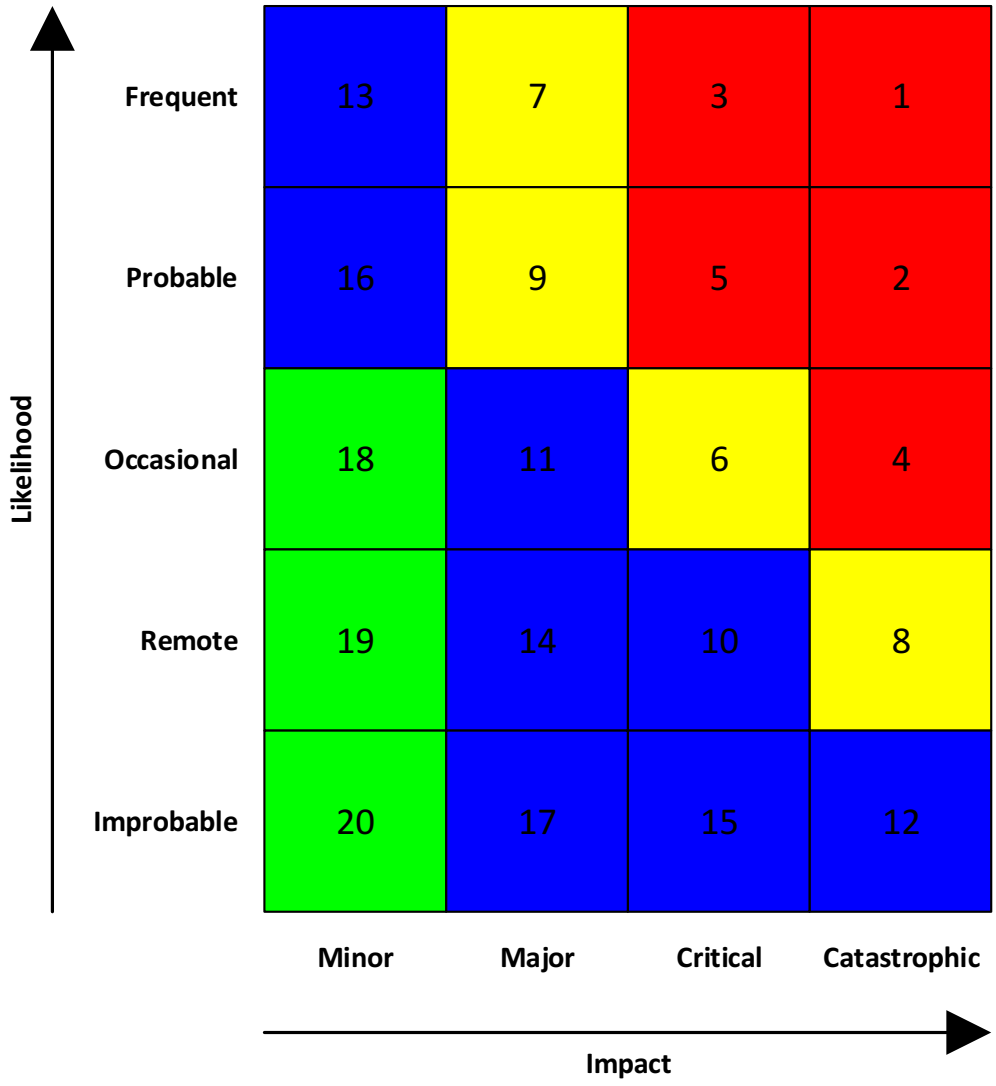

Figure 3: Risk = Likelihood × Impact

Risk Index	Risk Level	Risk Acceptability
1 to 5	Extremely High	Intolerable
6 to 9	High	Unacceptable
10 to 17	Medium	Acceptable with continuous review
18 to 20	Low	Acceptable with periodic review

3.3.1.3 Likelihood

Likelihood is the chance that an attack will be performed, irrespective of success or failure. This field attempts to encompass such things as frequency with which such attacks are seen and simplicity of attack (as opposed to sophistication).

Likelihood	Description
Frequent	Likely to occur regularly
Probable	Will occur several times in the life of the system
Occasional	Unlikely, but can be reasonably expected to occur in the life of the system
Remote	Unlikely, but possible to occur in the life of the system
Improbable	So unlikely it may not be expected

3.3.1.4 Impact

Impact is the effect on User's security if the attack succeeds. This attempts to convey how badly the User's security would be compromised.

Impact	Description
Catastrophic	Compromise of multiple User's authorised accounts
Critical	Long-term/repeatable compromise of a User's authorised accounts
Major	Temporary/short-lived compromise of a User's authorised accounts
Minor	No compromise of the User's authorised accounts, but reduces the difficulty of future compromises

3.3.1.5 Legend

Field	Use
Description	A high-level overview of the attack
Execution	How the attack is typically carried out
Mitigation	What has been done or can be done to prevent the attack or minimise its impact

Field	Use
Exposure	What is at risk if the mitigations fail and the attacker succeeds in the attack
Timing	When the attack must be performed for the associated actions to be available to the attacker. Where not present, timing is irrelevant.
Action	Options open to the attacker
Requires	Additional tasks required for an attacker to perform the action. Where not present, no additional actions are required.
Effect	What effect the attacker's action may have

3.3.2 Social Engineering

Description	Social engineering is the practice of psychological manipulation of people into performing actions or divulging sensitive information
Execution	• Pretexting • Phishing • Vishing • Baiting
Mitigation	Everyone using TokenOne Authentication should receive training on how to recognise and deal with social engineering. Security protocols should be put in place to reduce the likelihood or impact of social engineering attempts.
Exposure	The exposure to a social engineering attack is unlimited – if an employee grants an attacker administrative access to the TokenOne Server, the attacker could bring the system down, steal every User's data, impersonate every User, and so on. The likelihood of this occurring depends purely on the training Users.
Likelihood	Probable
Impact	Catastrophic

3.3.3 Compromise (TokenOne Server)

<table>
<tr><td>Description</td><td>Compromising the TokenOne Server involves the attacker gaining illicit access to it</td></tr>
<tr><td>Execution</td><td>• Social engineering
• Zero-day exploit
• Physical access</td></tr>
<tr><td>Mitigation</td><td>Physical access to the TokenOne Server should be tightly controlled. The TokenOne Server should remain patched at all times. The TokenOne Server should have proactive cyber defence, including but not limited to such things as:
• Intrusion monitoring active at all times
• Audit logs reviewed regularly
• Honeypots set up
• Pre-emptive and counter-active tactics in place and practiced.

Threat analysis should be performed regularly. People with access to the TokenOne Server should be fully trained in how to recognise and deal with social engineering attacks. Numerous moats should be set up to make social engineering attacks more challenging (e.g. use of a strong secure cryptographic key to authenticate email origin and prevent spoofing; encrypted hard drives for all portable devices). All passwords used in conjunction with the TokenOne Server should be strong. Such passwords should be changed regularly, and on employee turnover.</td></tr>
<tr><td>Exposure</td><td>
<table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td>Retrieve TokenOne certificate</td><td>Break into certificate store</td><td>Ability to spoof TokenOne Server
Ability to read messages sent to TokenOne Authentication</td></tr>
<tr><td>Retrieve database</td><td>Break into database</td><td>Ability to brute-force attack enciphered PINs</td></tr>
<tr><td>Retrieve TokenOne DLLs
Decompile TokenOne DLLs
Update obfuscated TokenOne code
Recompile TokenOne code
Replace TokenOne DLLs</td><td>Administrator access</td><td>Complete access to every User’s TokenOne and authorised accounts until the hack is caught or files replaced</td></tr>
</table>
</td></tr>
<tr><td>Likelihood</td><td>Occasional</td></tr>
<tr><td>Impact</td><td>Catastrophic</td></tr>
</table>

3.3.4 Key Logging (Terminal)

<table>
<tr><td>Description</td><td>Key logging involves capturing a User's interaction with a computer; in this case, the User's terminal. Despite the name, sophisticated key loggers can capture all forms of User interaction (viz. keyboard, mouse, gesture, microphone) along with details of what is running and what is being displayed.</td></tr>
<tr><td>Execution</td><td>• Physical access to terminal enabling manual installation of spyware
• Physical access to terminal enabling manual installation of physical logger
• Luring the user into installing spyware on the terminal</td></tr>
<tr><td>Mitigation</td><td>As a User can use any terminal for TokenOne Authentication, little can be done to guarantee security of the device. For corporate assets, security policies should be enforced that prevent installation of software, and anti-malware sweeps should be performed. Educating Users on proper security and vigilance is also highly recommended. Encouraging Users to use trusted terminals wherever possible is highly recommended.</td></tr>
<tr><td>Exposure</td><td>
<table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td>Intercept enciphered PIN</td><td>TokenOne certificate compromised
KeyMap is known</td><td>Ability to decipher User's PIN</td></tr>
<tr><td>Intercept enciphered PIN
Block user message to TokenOne
Send enciphered PIN as self</td><td>TokenOne being used for strong 2FA
Not relevant for escalation</td><td>Ability to hijack User's TokenOne session</td></tr>
<tr><td>Intercept response</td><td>TokenOne being used for strong 2FA
Not relevant for escalation</td><td>Ability to piggy-back User's TokenOne session</td></tr>
</table>
</td></tr>
<tr><td>Likelihood</td><td>Frequent</td></tr>
<tr><td>Impact</td><td>Major</td></tr>
</table>

3.3.5 Man-in-the-Middle Attack (Terminal to TokenOne Server)

<table>
<tr><td>Description</td><td>A man-in-the-middle attack involves an attacker eavesdropping on communications between two parties. In this case, the User's terminal and the TokenOne Server.</td></tr>
<tr><td>Execution</td><td>• Luring the User onto an unsecure network (Wi-Fi)
• Hack a router between user and server
• Physically install sniffer between terminal and ISP</td></tr>
<tr><td>Mitigation</td><td>The TokenOne Server and User's terminal communicate over HTTPS. As an attacker will not have access to the private keys involved without compromising either the client or server. They will be unable to decrypt the data passed between the client and server.</td></tr>
<tr><td>Exposure</td><td>
<table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td>Intercept enciphered PIN</td><td>TokenOne certificate compromised

KeyMap is known</td><td>Ability to decipher User's PIN</td></tr>
<tr><td>Intercept enciphered PIN

Block user message to TokenOne

Send enciphered PIN as self</td><td>TokenOne Authentication being used for strong 2FA

Not relevant for escalation</td><td>Ability to hijack User's TokenOne session</td></tr>
<tr><td>Intercept response</td><td>TokenOne Authentication being used for strong 2FA

Not relevant for escalation</td><td>Ability to piggy-back User's TokenOne session</td></tr>
</table>
</td></tr>
<tr><td>Likelihood</td><td>Occasional</td></tr>
<tr><td>Impact</td><td>Major</td></tr>
</table>

3.3.6 Theft (Smart Device)

<table>
<tr><td>Description</td><td>Theft involves an attacker misappropriating something. In this case, the User’s smart device.</td></tr>
<tr><td>Execution</td><td>• Taken while left unattended
• Pickpocketed
• Burglary</td></tr>
<tr><td>Mitigation</td><td>Most Users are aware of the threat of theft and practice reasonable physical security. Users should be encouraged not to leave their smart device unattended or lend it to others. Users should be encouraged to always be aware of where their smart device is; if they cannot find their smart device, they should report it missing and (if possible) remotely brick the smart device until it is recovered. Users should be encouraged to secure their smart device with a password and (if available) a program that once triggered indicates the geolocation of the smart device. Where the smart device is a corporate asset, some or all of these suggestions may be enforceable by policy.</td></tr>
<tr><td>Exposure</td><td>
<table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td rowspan="2">Retrieve KeyMaps
Return device</td><td></td><td>Ability to activate KeyMaps
Drastically reduce effort of password attack</td></tr>
<tr><td>Enciphered PIN intercepted</td><td>Ability to decipher User’s PINs</td></tr>
<tr><td>Retrieve device certificate

Return device</td><td></td><td>Able to listen in on all future communications to User
Potentially ability to spoof smart device</td></tr>
<tr><td>Retrieve smart device identifier

Return device</td><td></td><td>Potential ability to spoof smart device</td></tr>
<tr><td rowspan="2">Install spyware

Return device</td><td></td><td>Ability to listen in on all future communications to user
Ability to activate KeyMaps</td></tr>
<tr><td>Enciphered PIN intercepted</td><td>Ability to decipher User’s PINs</td></tr>
</table>
</td></tr>
<tr><td>Likelihood</td><td>Frequent</td></tr>
<tr><td>Impact</td><td>Minor</td></tr>
</table>

3.3.7 Spoofing (TokenOne Server)

<table>
<tr><td>Description</td><td>Spoofing involves an attacker pretending to be somebody or something else. In this case, the TokenOne Server.</td></tr>
<tr><td>Execution</td><td>• Luring the User into installing malware that re-routes connections to the TokenOne Server (e.g. updates hosts file)
• Physical access to terminal/smart device enabling rerouting of connections
• DNS cache poisoning</td></tr>
<tr><td>Mitigation</td><td>To mitigate DNS cache poisoning, DNSSEC should be enabled (if available). Educating Users on proper security practices is highly recommended. If the smart device or terminal is a corporate asset, it may be possible to prevent some forms of re-routing (e.g. by making the hosts file inaccessible to the User’s privilege level).</td></tr>
<tr><td>Exposure</td><td>
<table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td rowspan="3">Collect KeyMap activation</td><td></td><td>Awareness that attack window is open</td></tr>
<tr><td>TokenOne certificate compromised</td><td>Awareness of active KeyMap identifier</td></tr>
<tr><td>TokenOne certificate compromised

KeyMap is known</td><td>Ability to decipher User’s PIN</td></tr>
<tr><td>Collect smart device identifier</td><td></td><td>Potential ability to spoof smart device</td></tr>
<tr><td>Collect enciphered PIN</td><td>TokenOne certificate compromised

KeyMap is known</td><td>Ability to decipher User’s PIN</td></tr>
<tr><td>Send enciphered PIN as self</td><td>TokenOne Authentication being used for strong2FA
Not relevant for escalation</td><td>Ability to hijack User’s TokenOne session</td></tr>
<tr><td>Send request to TokenOne Server
Intercept response</td><td>TokenOne Authentication being used for strong 2FA
Not relevant for escalation</td><td>Ability to piggy-back User’s TokenOne session</td></tr>
</table>
</td></tr>
<tr><td>Likelihood</td><td>Remote</td></tr>
<tr><td>Impact</td><td>Major</td></tr>
</table>

3.3.8 Physical Monitoring

<table>
<tr><td>Description</td><td>Physical monitoring involves the in-person or remote observation of a User</td></tr>
<tr><td>Execution</td><td>• Lurk over their shoulder
• Install a camera
• Hijack their web camera</td></tr>
<tr><td>Mitigation</td><td>Educating Users on proper security practices is highly recommended. Users should be mindful of anyone observing them when using TokenOne Authentication and should avoid using TokenOne Authentication in public areas if possible. Users should be mindful of strange cameras that suddenly appear in their house or workplace. Users should use anti-spyware tools to detect and remove hijacks of their digital devices.</td></tr>
<tr><td>Exposure</td><td><table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td>Observe KeyMap</td><td>Enciphered PIN intercepted</td><td>Ability to decipher User’s PIN</td></tr>
<tr><td>Observe User’s enciphered PIN</td><td>KeyMap is known</td><td>Ability to decipher User’s PIN</td></tr>
</table></td></tr>
<tr><td>Likelihood</td><td>Probable</td></tr>
<tr><td>Impact</td><td>Minor</td></tr>
</table>

3.3.9 Man-in-the-Middle Attack (Smart Device to TokenOne Server)

Description

A man-in-the-middle attack involves an attacker eavesdropping on communications between two parties. In this case, the User's smart device and the TokenOne Server.

Execution

- Luring the User onto an unsecure network (Wi-Fi)
- Hack a router between User and server
- Spoof cell tower

Mitigation

The TokenOne Server and User's smart device confirm each other's identity by means of their SSL certificates (mutual authentication). As an attacker will not have access to the private keys of these certificates without compromising either the client or server, they cannot take the place of either party. The User and TokenOne Server communicate over HTTPS; as the User does not have the private keys used to decrypt the conversation, they cannot to read the data passed between the client and server.

Exposure

Timing	Action	Requires	Effect
Account creation	Copy/replace smart device certificate		Able to listen in on all future communications to user Potentially ability to spoof smart device
Account creation	Intercept KeyMaps		Drastically reduce effort of password attack
		Smart device spoofed	Ability to activate KeyMaps
		Enciphered PIN intercepted	Ability to decipher User's PINs
Anytime	Intercept KeyMap activation		Awareness that attack window is open
		TokenOne certificate compromised	Awareness of active KeyMap identifier
		TokenOne certificate compromised KeyMap is known	Ability to drastically reduce effort of password attack
Anytime	Intercept smart device identifier	TokenOne certificate compromised	Potential ability to spoof smart device

Likelihood	Occasional
Impact	Minor

3.3.10Spoofing (Smart Device)

Description	Spoofing involves an attacker pretending to be somebody or something else. In this case, the User's smart device.
Execution	To spoof the User's smart device, an attacker must do all of the following: • Acquiring the certificate issued to smart device by TokenOne • Acquire the secret properties of the device, such as its identifier • Build custom software to interface with the TokenOne server
Mitigation	The TokenOne Server and User's terminal communicate over HTTPS. As an attacker will not have access to the private keys involved without compromising either the client or server, they cannot decrypt the data passed between the client and server. The TokenOne team is continuing to investigate options to further guarantee the presence of the User's actual smart device.
Exposure	**Action** | **Requires** | **Effect** Activate a KeyMap | KeyMap identifier known | Ability to perform password attack.
Likelihood	Occasional
Impact	Minor

3.3.11Memory Sniffing (Smart Device)

<table>
<tr><td>Description</td><td>Memory sniffing involves an attacker accessing the data on a device; in this case, the User’s smart device.</td></tr>
<tr><td>Execution</td><td>• Physical access to smart device enabling manual installation spyware
• Luring the User into installing a Trojan on their smart device
• Physical access to smart device enabling disassembly</td></tr>
<tr><td>Mitigation</td><td>Smart device security is largely regulated by the User and the issuing body. In the case of corporate issued smart devices, certain aspects of the device (such as ability to install software) may be locked down, and certain security (such as requiring a password to unlock the device) may be instituted. Educating Users on proper security and vigilance is also highly recommended.

While certain application marketplaces are more secure than others, researchers have demonstrated all marketplaces can be compromised with time-delayed Trojans. Once a Trojan is in place, its ability to compromise the User’s security will depend on what privileges it has been granted and what unpatched security flaws exist in the device.</td></tr>
<tr><td>Exposure</td><td>
<table>
<tr><th>Action</th><th>Requires</th><th>Effect</th></tr>
<tr><td rowspan="2">Retrieve KeyMaps</td><td></td><td>Ability to activate KeyMaps
Drastically reduce effort of password attack</td></tr>
<tr><td>Enciphered PIN intercepted</td><td>Ability to decipher User’s PINs</td></tr>
<tr><td>Retrieve smart device certificate</td><td></td><td>Able to listen in on all future communications to User
Potentially ability to spoof smart device</td></tr>
<tr><td>Retrieve smart device identifier</td><td></td><td>Potential ability to spoof smart device</td></tr>
</table>
</td></tr>
<tr><td>Likelihood</td><td>Remote</td></tr>
<tr><td>Impact</td><td>Minor</td></tr>
</table>

Appendix A: Glossary

Term	Abbr.	Definition
Application Programmer Interface	API	An application program interface is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact
Authorised Account		An electronic account with a known external service provider that has been linked to a User's TokenOne Account
Automatic Teller Machine	ATM	An electronic banking outlet, that allows customers to complete basic transactions without the aid of a teller
Baiting		A social engineering attack vector where a hacker leaves Trojan-infected physical media (e.g. USB drive, CD) in a location where an employee will find it, in the hope that the employee will open it on a corporate terminal.
Brick		The process of rendering a digital device non-functional
Brute-Force Attack		An attack vector of last resort where a hacker performs an exhaustive search (e.g. testing every combination for a PIN)
Ciphertext		A message or document that has been converted to a code or code or cipher
Cryptanalysis		The process of breaching cryptographic systems by means of targeting vulnerabilities in the algorithm of implementation (side-channel attacks).
Cryptovariable		A cryptovariable (also known as a key) is a variable secret used by encryption algorithms
Device Identifier		A unique identifier belonging to the smart device
DNS Cache Poisoning		A hacking technique where fake data is introduced into a DNS name server's cache database, causing the name server to return an incorrect IP address, diverting traffic to another computer
Domain Name System	DNS	A hierarchical, distributed naming system for computers, services and other resources attached to a network
Domain Name System Secure Extensions	DNSSEC	A set of extensions to DNS that provide resolvers with origin authentication of DNS data, authenticated denial of existence, and data integrity

Term	Abbr.	Definition
Doxing		The process of gathering personal information about an individual, particularly by means of the Internet
Dynamic-Link Library	DLL	Microsoft's implementation of the shared library concept
Encipher		The process of encoding plaintext by substituting one character for another in a prescribed way
Encrypt		The process of encoding plain text by using an algorithm
Escalation		The process of obtaining further proof of identity from a User prior to performing certain actions
Geolocation		The real-world geographic location of an object
Hash		A representation of data that has been transformed by an algorithm. The hash is typically smaller than the data and of a fixed length. The hash typically cannot be turned back into the data, even with awareness of the algorithm.
Honeypot		A trap set to detect, deflect or counteract attempts at unauthorised use of information systems
HTTP Secure	HTTPS	HTTP layered on top of the SSL/TSL protocol. The foundation of secure data communications for the web.
Hypertext Transfer Protocol	HTTP	The foundation of data communications for the web
Identity Verification	IDV	A digital service that investigates and corroborates details of a person. Also, the process of that investigation and corroboration.
In-Band		Communication that occurs via the default medium (the medium that is used for other communication). Opposite of Out-of-Band.
Internet Protocol	IP	The principal communications protocol in the Internet Protocol Suite for relaying datagrams across network boundaries
Internet Service Provider	ISP	An organisation that offers Users access to the Internet
IP Address		A numerical label assigned to each device in a computer network
Jamming		The process of preventing a wireless or cellular device from connecting to its network by increasing the noise-to-signal ratio

Term	Abbr.	Definition
KeyMap		A cryptovariable generated by TokenOne Authentication, mapping the digits 0-9 to 10 random alphabetic characters (A-Z)
Logger		Software or hardware used to record interactions between human and computer
Long Term Evolution	LTE	A standard for wireless communication of high-speed data for mobile phones and data terminals
Man-in-the-Middle	MITM	An attack vector where a hacker eavesdrops on communications between two parties. Typically performed by compromising a router, luring a user onto an unsecure network, or compromising the computer of one or other party.
Moat		A generic term for any defence mechanism (digital or physical) used in the protection of an asset
Multi-Factor Authentication	MFA	An approach to authentication that requires at least two of the three authentication factors: • Knowledge factor (something only the user knows) • Possession factor (something only the user has) • Inherence factor (something only the user is)
One-Time Pad	OTP	A cryptographic tool enabling encipherment, where each character in the plaintext is encoded by modular addition with a character from a secret random key. The key must be at least as long as the plaintext, non-repeating and non-predictable.
One-Time Password		A password that is used once only
Out-of-Band		Communication that occurs via a non-default medium (not via the medium used for other communication). Opposite of In-Band.
Password Attack		An attack vector where an attacker guesses the user's password, whether via a dictionary attack or a brute force attack
Personal Identification Number	PIN	A secret numeric password
Phishing		A social engineering attack vector where a hacker attempts to acquire privileged information by masquerading as a trustworthy entity in electronic communication
Plaintext		The unencrypted form of an encrypted message
Polyalphabetic Substitution Cipher		A Substitution Cipher using a number of substitutions at different points in the message

Term	Abbr.	Definition
Pretexting		A social engineering attack vector where an hacker invents a scenario to engage the targeted victim in a manner that increases the chances a victim will divulge privileged information or perform actions that would otherwise be unlikely
Private Key		A 'secret' used in asymmetric cryptography
Proactive Cyber Defence		To actively oppose anticipated attacks against computers and networks
Salt		Random data added to the input of one-way hash functions, used to defend against rainbow table attacks
Secure Sockets Layer	SSL	A cryptographic protocol designed to provide communication security over the Internet
Security Assertion Mark-up Language	SAML	An XML-based open standard data format for exchanging authentication and authorisation data between parties
Short Message Service	SMS	A text messaging service component of mobile communication systems
Single Sign-On	SSO	An access control pattern where a User signs into (authenticating) one system is also authenticated to other independent systems
Smart Device		An electronic device, such as a tablet or smartphone, connected to the Internet
Sniffer		A computer program that can intercept and log traffic passing over a digital network
Spoofing		An attack vector where one or more details of a transmission are changed, so that the attacker appears to be someone else
Spyware		Software that aids in gathering digital information
SSL Certificate		A digital credential issued by a trusted party and used in identification of the recipient
Strong Two-Factor Authentication	Strong 2FA	An approach to authentication that requires at least two of the three authentication factors: • Knowledge factor (something only the User knows) • Possession factor (something only the User has) • Inherence factor (something only the User is) For a *strong t*wo-factor Authentication with TokenOne Authentication you never enter, store or transmit the Knowledge Factor (your Password) making it a *strong* two-factor Authentication method.

Term	Abbr.	Definition
Substitution Cipher		A method of encoding where units of plaintext are replaced by ciphertext, according to a regular system. The "units" may be single letters (the most common) or pairs of letters, triplets of letters, or even mixtures of the above.
Symmetrical Cipher		An algorithm that uses the same cryptovariable for both encoding and decoding
TokenOne		Where the term "TokenOne" is used without further qualification in a document means the TokenOne solution as a whole. This includes the implementation of TokenOne Authentication.
TokenOne Authentication		Where the term "TokenOne Authentication" is used, it refers to the theory behind TokenOne's authentication component
Two-factor Authentication		An approach to authentication that requires two of the three authentication factors: • **Knowledge factor** (something only the User knows) • **Possession factor** (something only the User has) • **Inherence factor** (something only the User is)
Trojan		A computer program that provides back-door access to any machine where it is installed
True Token		A possession factor that cannot be spoofed
Vishing		A social engineering attack vector where a hacker attempts to acquire privileged information by masquerading as a trustworthy entity in vocal/telephonic communication
Wi-Fi		A technology that allows an electronic device to connect to a network without physical cables
Zero-Day Exploit		Vulnerability in a computer program that has no patch (often because the vulnerability is unknown to the vendor)