A Secure Mobile OTP Token

by: Fred Cheng

FPC Consultancy
International Technological University
fretchenting@yahoo.com
Topics

• Introduction
• Background & Selected Related Works
• Base Cipher
• Implementing a Secure Mobile OTP Token
• Security Analysis
• Conclusion
Introduction

• Using One-time Password (OTP) for remote authentication becomes popular.
• It is natural to have mobile phone as an OTP token.
• This paper proposes an encryption cipher to build a secure Mobile OTP token that can resist certain security attacks.
• The token also preserves full compliance and interoperability with existing infrastructure.
Why OTP & OTP Token?

- **OTP**: One-time Password
  - For network remote authentication

- **Security weakness with basic authentication**
  - Publicly known UserID
    - John Dole at ACE Corp. → j.dole@ace.com
  - Static Password
Password Security Attack

Attacks
- Dictionary & Brute-force Attack
- MITM Replay Attack: Capture the static password
- Seed-tracing (MITM), Shoulder-surfing, & …

Solutions
- Increase password complexity (OTP)
- Dynamic password (OTP)
- No simple and easy solution
Dynamic Password

- Proposed by Leslie Lamport
  - In his landmark 1981 ACM Paper

- The Algorithm
  - Using a One-way Function – $F$
    - An initial seed $x$ & event counter $i$
  - Dynamic session password
    - $F(x), F(F(x)), \ldots, F^i(x)$
  - Each password is only used once in one session.

- The beginning of a One-time Password (OTP) development.

OTP Token

• Various OTP algorithms and implementations were introduced and marketed
  – Expensive, in-compatible & non-interoperable

• OATH: Initiative for Open AuTHenticaTion
  – Free Standard, Compatibility, Interoperability & Low Cost
  – OTP Algorithm – RFC4226

\[ \text{OTP} = \text{Truncate (HMAC-SHA-1(K, C))} \]
Variety of OTP Tokens

- Hardware
- Software
- Mobile OTP Token
  - Embeds OTP function in cellular phone
Stand-alone Mobile OTP Token

• Cellular phone is an OTP Token
  – Generating OTP code
  – Replacing the dedicated H/W or S/W OTP token

• Software based token
  – Seed (K) and Counter (C) are stored inside phone

• Ref: [1] [7] [8]
Out of Band Transceiver

- Cellular Network
  - Secure out of band channel
- Phone (using SMS)
  - Transceiver of the OTP code
- Seed (K) and C storage
  - At server or computer
- Limitation
  - Unreliable & untimely SMS
  - Cellular service coverage

- Ref: [2] [9] [10]
Mobile Authenticator

• Authentication
  – Provided by cellular system
  – New Protocols [9][10]
  – SIM (user credentials)
• Phone
  – Contains SIM
• Limitation
  – Cellular service coverage
• Ref: [3][13][14][15][16][17][18]
# Mobile OTP Solutions

<table>
<thead>
<tr>
<th>Items</th>
<th>Stand Alone Token</th>
<th>Out of Band TXR</th>
<th>Mobile Authenticator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role of phone</td>
<td>Computational platform</td>
<td>Transceiver of OTP code</td>
<td>Part of the authenticator</td>
</tr>
<tr>
<td>OTP generation</td>
<td>Phone</td>
<td>Phone or Server</td>
<td>Phone and Server</td>
</tr>
<tr>
<td>OTP submission</td>
<td>Through PC</td>
<td>SIM &amp; SMS</td>
<td>SIM &amp; Protocol</td>
</tr>
<tr>
<td>Type of phone</td>
<td>2.5G &amp; up</td>
<td>2.5G &amp; up</td>
<td>3G &amp; up</td>
</tr>
<tr>
<td>Simple usability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Low phone $/m</td>
<td>Yes (zero)</td>
<td>No (SMS Plan)</td>
<td>No (3G Data Plan)</td>
</tr>
<tr>
<td>No cellular limitation</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compatibility &amp; Interoperability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No system change</td>
<td>Yes</td>
<td>No (additional H/W,S/W )</td>
<td>No (complex system)</td>
</tr>
<tr>
<td>Protect secrets</td>
<td>No</td>
<td>Yes</td>
<td>Yes/ No</td>
</tr>
<tr>
<td>MITM attack safe</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Shoulder-surfing attack safe</td>
<td>No</td>
<td>No</td>
<td>?</td>
</tr>
</tbody>
</table>

Mobilware 2010
July 1, 2010
A Secure Mobile OTP Token
Fred Cheng
Low Cost Mobile OTP Token

• Stand-alone Mobile Token has its merits
  – Works with existing authentication infrastructure
  – Low cost – deployment & supporting
  – No cell coverage limitation

• Need to solve
  – Protecting the secrecy
    • Seed (K) and counter value (C)
  – Protecting security attacks
    • MITM seed tracing
    • Shoulder-surfing
Requirements & Solutions

- An Event-based OATH Mobile OTP Token
- Security Protection

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preserve phone power</td>
<td>Less computation &amp; local code</td>
</tr>
<tr>
<td>Compatibility &amp; Interoperability</td>
<td>Using same OTP algorithm</td>
</tr>
<tr>
<td>Protect local confidential data</td>
<td>Rubbing Encryption Algorithm</td>
</tr>
<tr>
<td>Resist security attack</td>
<td>Rubbing Encryption Algorithm</td>
</tr>
<tr>
<td>OTP</td>
<td>OATH OTP</td>
</tr>
<tr>
<td>Seed Tracing</td>
<td>New solution</td>
</tr>
<tr>
<td>Shoulder-surfing</td>
<td>New solution</td>
</tr>
</tbody>
</table>
Rubbing Encryption Algorithm

- A secure scramble algorithm that uses complex key

**Features**
- Key embedded in H/W token
- Decrypting w/o Entering Key
- Long & Complex Key
- Secure Scramble Algorithm

**Benefits**
- No need to memorize key
- Using long & complex key
- High security with short plaintext
- No complex computation
REAL Cipher - Math

Given a numeric image $X$ containing $T$ characters selected from $Y$ numerals,

$$x_1 \ x_2 \ x_3 \ \ldots \ \ldots \ x_i \ \ldots \ x_t$$

The occurrence possibility ($P_i$) of a numeral $Y_i$ (assumed appears $N_{yi}$ times) is,

$$P_i = \frac{N_{yi}}{T} \quad (1)$$

Following Shannon Entropy Theory, Image $X$’s uncertainty $H(X)$ is as follows.

$$H(X) = - \sum_{i=1}^{T} P_i \cdot (\log_2 P_i) \quad (2)$$

When each numeral has equal chance to be displayed and $N_{yi}$ are all equal ($N$), image $X$ reaches a Equiprobable state and has the highest uncertainty. [20]

$$T = NY, \quad (3)$$

$$P_i = \frac{N_{yi}}{T} = \frac{N}{NY} = \frac{1}{Y} = P. \quad (4)$$
REAL Cipher - Math

Substituting (4) into (2), Image X’s uncertainty $H(X)$ becomes

$$H(X) = \frac{T \log_2 Y}{Y}.$$  \hspace{1cm} (5)

Following similar procedure, each symbol’s uncertainty $H(S)$ can be found as follows

$$H(S) = \frac{T \log_2 Y}{Y^2}.$$  \hspace{1cm} (6)
REAL Operating Procedure

1. A user signs up at server with her credential
   - TO Encrypt
     - Follow the numerical (1) … (12) step
   - To Decrypt
     - Follow the alphabet [A] … [L] step

2. Draft REAL H/W Token & Key
3. Generating OATH OTP Code
4. Encrypting with REAL Key
5. Generating REAL Image
6. Generating DATA
7. Obtaining HI
8. Generating OFFSET
9. Bit XOR
10. DELTA
11. Encryption
12. Secure Storing Delta Table (DT) & HI in local device

1. REAL OTP Token
2. Activation & key-in
3. User Credential

(0) [A]
(1) [B]
(2) [C]
(3) [D]
(4) [E]
(5) [F]
(6) [G]
(7) [H]
(8) [I]
(9) [J]
(10) [K]
(11) [L]
REAL Image-Token-Key (Step 1&2)

- REAL Image Generation
  - Security level setting
  - Screen size, font size, token size, usability and other factors
- REAL Hardware Token
  - Low cost, easy to carry and use
- REAL Key
  - Code pointer as key
- Token can be of multi-dimensions
  - REAL key can be in multi-dimensional form
Encryption with REAL Key (step 3~6)

- REAL key position → \( W_6 \sim W_0 \)
- Pre-generated OTP Code (= 807235)
  - \( D_5 = 8, D_4 = 0, D_3 = 7, D_2 = 2, D_1 = 3 \) and \( D_0 = 5 \)
  - Program Digit: \( W_6 = D_6 = 3 \) (odd → front key, even → backside key)
- Randomly place other numerals to make \( X \) an Equiprobable Image

<table>
<thead>
<tr>
<th>Placement of OTP Code</th>
<th>Key Locations</th>
<th>( W_6 )</th>
<th>( W_5 )</th>
<th>( W_4 )</th>
<th>( W_3 )</th>
<th>( W_2 )</th>
<th>( W_1 )</th>
<th>( W_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>D_6</td>
<td>D_5</td>
<td>D_4</td>
<td>D_3</td>
<td>D_2</td>
<td>D_1</td>
<td>D_0</td>
</tr>
<tr>
<td>II</td>
<td>II</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

Final REAL Image \( X = \text{DATA}(i) = \text{Concatenate} \ (X_{40} \sim W_i \sim X_1) \):

\[
\begin{align*}
X_{40} & \quad X_{39} & \quad \ldots & \quad X_k & \quad \ldots & \quad X_8 & \quad \ldots & \quad X_j & \quad \ldots & \quad X_0 & \quad \ldots & \quad X_{h} & \quad \ldots & \quad X_2 & \quad X_1
\end{align*}
\]
Offset & HI Generation (step 7~8)

- Offset is generated from the last Hashed Index (HI)
  - Reuse OATH OTP generation algorithm
**Delta Table Generation (Step 9~12)**

- Delta(i) = Bit Ex-OR (Offset(i), Data(i))
- Delta Table (DT)
  - Compilation of Delta(i) with HI(i)
  - Rearranging Delta(i) order according to the value of HI(i)
  - Ensure higher security with the local storage
- User_Key = HMAC-SHA-1(HMAC-SHA-1(UC, UC)), UC

**Diagram:****

- Offset(i)
  - Bit319 - Bit160
  - Bit159 - Bit0
- Bit Ex-OR
- Data(i)
  - Bit319 - Bit160
  - Bit159 - Bit0

- Delta(i)
  - Bit319 - Bit160
  - Bit159 - Bit0

- HI(i)
  - HI(i-1)
  - HI(i)
  - HI(i+1)
  - HI(i+n)

- Delta Table
  - Delta(i+k)
  - Delta(i-n)
  - Delta(i)
  - Delta(i+m)

- HI(i)
  - HI(i+k)
  - HI(i-n)
  - HI(i)
  - HI(i+m)

 Encryption with a User Key

Stored in phone

Mobilware 2010
July 1, 2010

A Secure Mobile OTP Token
Fred Cheng
REAL Decryption (Step A~L)

- Reverse previous steps
- Place H/W token over the numerical image (beginning with token’s front side 1\textsuperscript{st})
- Rubbing sequence
  - 1\textsuperscript{st} Pointer indicates front (odd number) and backside (even number) key selection
  - Reading from 2\textsuperscript{nd} pointer: Top/Down or Left/Right and clockwise

Code = 807235

Code = 478818
Security Attacks – Seed Tracing

- OTP code is generated by a known algorithm with a static SEED
- Collecting codes & comparing to code database
  - Finding Pseudo Random Sequence

\[ \text{OTP}_1, \text{OTP}_2, \text{OTP}_3, \text{OTP}_4, \text{OTP}_5, \text{OTP}_6, \ldots \rightarrow \text{OTP Seed} \]

- Solution
  - Multi-Seeding OTP (Ms.OTP)
    - To break the Pseudo Random Sequence
    - Increase OTP code randomness

Mobilware 2010
July 1, 2010
A Secure Mobile OTP Token
Fred Cheng
Multiple Seeding OTP (Ms.OTP)

- A REAL Bi-Seeding Mobile OTP Token
  - One H/W token with two encryption keys
  - One REAL key to encrypt codes from one OTP Seed
  - Front 1st pointer to show which key to rub the OTP code
- Randomly mixing OTP codes from either Seed
  - Breaking pseudo random sequence from collected codes
  - Server records the mixing pattern during the provisioning

OTP_{seed-A}, OTP_{seed-B}, OTP_{seed-A}, OTP_{seed-A}, OTP_{seed-B}, OTP_{seed-A}, OTP_{seed-B}, ...
Multiple Seeding OTP (Ms.OTP)

<table>
<thead>
<tr>
<th></th>
<th>D_6</th>
<th>D_5</th>
<th>D_4</th>
<th>D_3</th>
<th>D_2</th>
<th>D_1</th>
<th>D_0</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>Seed A, Token front side</td>
</tr>
<tr>
<td>B1</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>Seed B, Token back side</td>
</tr>
</tbody>
</table>

OTP Code = 807235

OTP Code = 478818
Security Attacks – Shoulder-surfing

- Secretly observing and collecting either OTP codes or token pointer locations
  - To trace OTP Seed or
  - To trace REAL encryption keys

Solution

- Ms.OTP → Preventing OTP Seed tracing
- Multi-Random OTP (Mr.OTP)
  - Preventing REAL key tracing
  - Breaking code to the pointer’s physical locations
Multiple Random OTP (Mr.OTP)

1\textsuperscript{st} pointer’s code ($N_{1f}$ or $N_{1b}$) value provides the seed for creating the extra randomness.

Each code value adds the 1\textsuperscript{st} pointer’s code value and drops the 10s digit if the sum is greater than 10.

\[ D_{if} = (\text{Value of } N_{1f} + \text{Value of } N_{(7-i)f}) \mod 10 \]

<table>
<thead>
<tr>
<th></th>
<th>$D_6$</th>
<th>$D_5$</th>
<th>$D_4$</th>
<th>$D_3$</th>
<th>$D_2$</th>
<th>$D_1$</th>
<th>$D_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>II</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>III</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

OTP Code = 130568
# REAL Secure Mobile OTP Token

- Resists OTP Seed-tracing and Shoulder-surfing attacks
- **REAL Secure Mobile OTP Token = Ms.OTP + Mr.OTP**

<table>
<thead>
<tr>
<th></th>
<th>D_6</th>
<th>D_5</th>
<th>D_4</th>
<th>D_3</th>
<th>D_2</th>
<th>D_1</th>
<th>D_0</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>Seed A, front side, Ms.OTP</td>
</tr>
<tr>
<td>A2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>OTP = Ms.OTP + Mr.OTP</td>
</tr>
<tr>
<td>B1</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>Seed B, back side, Ms.OTP</td>
</tr>
<tr>
<td>B2</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>OTP = Ms.OTP + Mr.OTP</td>
</tr>
</tbody>
</table>

- **OTP Code** = 130568
- **OTP Code** = 034474

Mobilware 2010
July 1, 2010
A Secure Mobile OTP Token
Fred Cheng
Other Security Analysis

• Phone is lost or stolen
  – Have REAL Image w/o H/W token (if breaks the U_K 1st)
    \[ \text{Possibility (P_1) = } \frac{1}{C(40, 6)} = 2.6 \times 10^{-7} \]
  – Decrypt codes from Delta Table directly
    Needs user credential (U_K) & HI values

• H/W token is lost, stolen or secretly copied
  – No phone (REAL Image)
    \[ \text{Possibility (P_2) = } 1 \times 10^{-7} \]

• H/W token & DT are secretly copied
  Protected by user credential (U_K)

• Brute-force guess possibility = \(1 \times 10^{-6}\)
Conclusion

• Rubbing Encryption Algorithm (REAL)
  – A multi-dimensional secure cipher with long and complex keys
  – Provides high security level encryption for short length plaintext

• A REAL Secure Mobile OTP Token
  – Securely protects local stored confidential data
  – Resists MITM Seed-tracing and Shoulder-surfing attacks
  – Low cost, compatible & interoperable with existing authentication infrastructures

• Further Work
  – To explore more apps. on REAL multi-dimension, multi-key features and improve the usability against desired security level
  – Example:
    • “A Novel Rubbing Encryption Algorithm and The Implementation of a Web-based One-time Password Token”, COMPSAC 2010. [23]
THANKS!

Q & A