Attacking and Fixing PKCS#11 Security Tokens with Tookan

Matteo Bortolozzo, Matteo Centenaro
Riccardo Focardi and Graham Steel

Università Ca’ Foscari, Venezia
and
LSV, INRIA & CNRS & ENS-Cachan
RSA Public Key Cryptographic Standard 11

Describes ‘cryptoki’: cryptographic token interface

Widely adopted in industry for authentication tokens, smartcards (and HSMs, other devices, …)
RSA Public Key Cryptographic Standard 11

Describes ‘cryptoki’: cryptographic token interface

Widely adopted in industry for authentication tokens, smartcards (and HSMs, other devices, …)

Keys (etc.) stored on the device and accessed by handles
RSA Public Key Cryptographic Standard 11

Describes ‘cryptoki’: cryptographic token interface

Widely adopted in industry for authentication tokens, smartcards (and HSMs, other devices, …)

Keys (etc.) stored on the device and accessed by handles

Attributes stored with keys to control usage
PKCS#11 Security

Section 7 of standard:
PKCS#11 Security

Section 7 of standard:

“1. Access to private objects on the token, and possibly to cryptographic functions and/or certificates on the token as well, requires a PIN.”
PKCS#11 Security

Section 7 of standard:

“1. Access to private objects on the token, and possibly to cryptographic functions and/or certificates on the token as well, requires a PIN.

2. Additional protection can be given to private keys and secret keys by marking them as “sensitive” or “unextractable”. Sensitive keys cannot be revealed in plaintext off the token, and unextractable keys cannot be revealed off the token even when encrypted”
PKCS#11 Security

Section 7 of standard:

“1. Access to private objects on the token, and possibly to cryptographic functions and/or certificates on the token as well, requires a PIN.

2. Additional protection can be given to private keys and secret keys by marking them as “sensitive” or “unextractable”. Sensitive keys cannot be revealed in plaintext off the token, and unextractable keys cannot be revealed off the token even when encrypted”

“Rogue applications and devices may also change the commands sent to the cryptographic device to obtain services other than what the application requested [but cannot] compromise keys marked “sensitive,” since a key that is sensitive will always remain sensitive. Similarly, a key that is unextractable cannot be modified to be extractable.”
Formal Model (Delaune, Kremer, S., CSF 2008)

Abstract ‘Dolev-Yao’ style

$$h(n_1, k_1)$$ - a handle $$n_1$$ for key $$k_1$$ (h is a *private symbol*)

$$a_1(n_1)$$ - setting of attribute $$a_1$$ for handle $$n_1$$

Command:

input;state $\xrightarrow{\text{new}}$ output;state'
Key Management - 1

KeyGenerate:

\[ \text{new } n,k \rightarrow h(n,k); L \]

Where \( L = \text{extract}(n), \neg \text{wrap}(n), \neg \text{unwrap}(n), \neg \text{encrypt}(n), \neg \text{decrypt}(n), \neg \text{sensitive}(n) \)
Key Management - 2

Set_Wrap : \[ h(x_1, y_1); \neg \text{wrap}(x_1) \rightarrow ; \text{wrap}(x_1) \]

Set_Encrypt : \[ h(x_1, y_1); \neg \text{encrypt}(x_1) \rightarrow ; \text{encrypt}(x_1) \]

UnSet_Wrap : \[ h(x_1, y_1); \text{wrap}(x_1) \rightarrow ; \neg \text{wrap}(x_1) \]

UnSet_Encrypt : \[ h(x_1, y_1); \text{encrypt}(x_1) \rightarrow ; \neg \text{encrypt}(x_1) \]

Some restrictions, e.g. can’t unset sensitive, can’t set extract
Key Management - 3

Wrap:

\[ h(x_1, y_1), h(x_2, y_2); \ wrap(x_1), \ \rightarrow \ \{y_2\}_{y_1} \]
\[ extract(x_2) \]

Unwrap:

\[ h(x_2, y_2), \{y_1\}_{y_2}; \ unwrap(x_2) \xrightarrow{\text{new } n_1} h(n_1, y_1); \ L \]

Where \( L = extract(n), \neg \text{wrap}(n), \neg \text{unwrap}(n), \neg \text{encrypt}(n), \neg \text{decrypt}(n), \neg \text{sensitive}(n) \)
Key Usage

Encrypt:

\[ h(x_1, y_1), y_2; \text{encrypt}(x_1) \rightarrow \{y_2\}_{y_1} \]

Decrypt:

\[ h(x_1, y_1), \{y_2\}_{y_1}; \text{decrypt}(x_1) \rightarrow y_2 \]
Fix decrypt/wrap, (and encrypt/unwrap):
Fix decrypt/wrap, (and encrypt/unwrap):

**Intruder knows:** $h(n_1, k_1), h(n_2, k_2), k_3$

**State:** sensitive($n_1$), extract($n_1$), extract($n_2$)

- **Set\_wrap:** $h(n_2, k_2) \rightarrow \text{;wrap}(n_2)$
- **Set\_wrap:** $h(n_1, k_1) \rightarrow \text{;wrap}(n_1)$
- **Wrap:** $h(n_1, k_1), h(n_2, k_2) \rightarrow \{k_2\}_{k_1}$
- **Set\_unwrap:** $h(n_1, k_1) \rightarrow \text{;unwrap}(n_1)$
- **Unwrap:** $h(n_1, k_1), \{k_2\}_{k_1} \xrightarrow{\text{new } n_3} h(n_3, k_2)$
- **Wrap:** $h(n_2, k_2), h(n_1, k_1) \rightarrow \{k_1\}_{k_2}$
- **Set\_decrypt:** $h(n_3, k_2) \rightarrow \text{;decrypt}(n_3)$
- **Decrypt:** $h(n_3, k_2), \{k_1\}_{k_2} \rightarrow k_1$
‘Tool for cryptoKi Analysis’
Configuration Language

Functions
Attributes
Always on/off
Conflicts
Tied
Templates
Flags

(see http://secgroup.ext.dsi.unive.it/tookan for full description)
<table>
<thead>
<tr>
<th>Brand</th>
<th>Model</th>
<th>Supported Functionality</th>
<th>Attacks found</th>
<th>Tookan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aladdin</td>
<td>eToken PRO</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓</td>
<td>wd</td>
</tr>
<tr>
<td>Athena</td>
<td>ASEKey</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bull</td>
<td>Trustway RCI</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓</td>
<td>wd</td>
</tr>
<tr>
<td>Eutron</td>
<td>Crypto Id. ITSEC</td>
<td>✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feitian</td>
<td>StorePass2000</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
<td>rs</td>
</tr>
<tr>
<td>Feitian</td>
<td>ePass2000</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
<td>rs</td>
</tr>
<tr>
<td>Feitian</td>
<td>ePass3003Auto</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
<td>rs</td>
</tr>
<tr>
<td>Gemalto</td>
<td>SEG</td>
<td>✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MXI</td>
<td>Stealth MXP Bio</td>
<td>✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA</td>
<td>SecurID 800</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
<td>rs</td>
</tr>
<tr>
<td>SafeNet</td>
<td>iKey 2032</td>
<td>✓ ✓ ✓ ✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Sata</td>
<td>DKey</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>rs</td>
</tr>
<tr>
<td>ACS</td>
<td>ACOS5</td>
<td>✓ ✓ ✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Athena</td>
<td>ASE Smartcard</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gemalto</td>
<td>Cyberflex V2</td>
<td>✓ ✓ ✓ ✓</td>
<td>✓</td>
<td>wd</td>
</tr>
<tr>
<td>Gemalto</td>
<td>SafeSite V1</td>
<td>✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gemalto</td>
<td>SafeSite V2</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>rs</td>
</tr>
<tr>
<td>Siemens</td>
<td>CardOS V4.3 B</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓</td>
<td>ru</td>
</tr>
</tbody>
</table>
Manufacturer Reaction

All 7 received notification at least 5 months before publication.

We offered to publish responses on project website
Manufacturer Reaction

All 7 received notification at least 5 months before publication.

We offered to publish responses on project website

RSA sent response, registered vulnerability with Mitre (CVE-2010-3321), issued security advisory 6 Oct 2010

Aladdin (now Safenet) sent a 2-page response for website
Manufacturer Reaction

All 7 received notification at least 5 months before publication.

We offered to publish responses on project website

RSA sent response, registered vulnerability with Mitre (CVE-2010-3321), issued security advisory 6 Oct 2010

Aladdin (now Safenet) sent a 2-page response for website

Bull invited me for a private meeting at their HQ
Manufacturer Reaction

All 7 received notification at least 5 months before publication.

We offered to publish responses on project website

RSA sent response, registered vulnerability with Mitre (CVE-2010-3321), issued security advisory 6 Oct 2010

Aladdin (now Safenet) sent a 2-page response for website

Bull invited me for a private meeting at their HQ

Gemalto responded to Cyberflex vulnerability, but not to SafeSite, and not to request to publish their response.

Minimal response from anyone else (e.g. requests to know who else is vulnerable)
OpencryptokiX

IBM Opencryptoki is a library including a software token

Vulnerable to many attacks (but it's a software token)
OpencryptokiX

IBM Opencryptoki is a library including a software token

Vulnerable to many attacks (but it’s a software token)

We have coded two fixed versions

- one implements config from Fröschle & Steel WITS ’09
- one is a new fix with no new crypto mechanisms

Uses a carefully chosen set of templates $G = \{wu, ed\}$, $U = \{eu\}$
OpencryptokiX

IBM Opencryptoki is a library including a software token

Vulnerable to many attacks (but it’s a software token)

We have coded two fixed versions

- one implements config from Fröschle & Steel WITS ’09
- one is a new fix with no new crypto mechanisms

Uses a carefully chosen set of templates \( G = \{wu, ed\}, \mathcal{U} = \{eu\} \)

Available to download from

http://secgroup.ext.dsi.unive.it/cryptokix
Conclusions

Tookan: our tool for formal analysis of PKCS#11 configurations

OpencryptokiX: a sandbox for trying token configurations

Bees: a library for programming PKCS#11 tokens using symbolic model language
Conclusions

Tookan: our tool for formal analysis of PKCS#11 configurations

OpencryptokiX: a sandbox for trying token configurations

Bees: a library for programming PKCS#11 tokens using symbolic model language

State of art of tokens not great (10/18 vulnerable, the rest very limited functionality)

Some manufacturers patching, no reaction from others
Conclusions

Tookan: our tool for formal analysis of PKCS#11 configurations

OpencryptokiX: a sandbox for trying token configurations

Bees: a library for programming PKCS#11 tokens using symbolic model language

State of art of tokens not great (10/18 vulnerable, the rest very limited functionality)

Some manufacturers patching, no reaction from others

Maybe we need a new standard with modern crypto? (OASIS, IEEE SISWG,...)
Conclusions

Tookan: our tool for formal analysis of PKCS#11 configurations

OpencryptokiX: a sandbox for trying token configurations

Bees: a library for programming PKCS#11 tokens using symbolic model language

State of art of tokens not great (10/18 vulnerable, the rest very limited functionality)

Some manufacturers patching, no reaction from others

Maybe we need a new standard with modern crypto? (OASIS, IEEE SISWG,...)

More details in the paper or online:

http://secgroup.ext.dsi.unive.it/tookan