

SECURE TEMPLATE

PKCS#11 ultimate fix

Bortolozzo Matteo

CREAZIONE DI CHIAVI

Quando si genera una nuova chiave simmetrica nel token può essere selezionato solo uno dei seguenti due template:

TEMPLATE	
Template 1	Template 2
MODIFIABLE = FALSE SENSITIVE = TRUE WRAP = TRUE UNWRAP = TRUE ENCRYPT = FALSE DECRYPT = FALSE	MODIFIABLE = FALSE SENSITIVE = TRUE WRAP = FALSE UNWRAP = FALSE ENCRYPT = TRUE DECRYPT = TRUE

Tabella 1.A

I rimanenti attributi possono essere settati a piacere.

Con tale configurazione viene "assorbita" la coppia di attributi in conflitto "wrap/decrypt" settati a TRUE.

Nel caso in cui si generi una coppia di chiavi asimmetriche, è necessario seguire il seguente template:

TEMPLATE	
Attributo	Chiave
MODIFIABLE = FALSE	pubblica e privata
SENSITIVE = TRUE	privata
WRAP = FALSE	pubblica
UNWRAP = FALSE	privata
ENCRYPT = TRUE	pubblica
DECRYPT = TRUE	privata

Tabella 1.B

UNWRAP, DUPLICAZIONE o IMPORT DI CHIAVI

Quando, nel caso di chiavi simmetriche, si esegue l'unwrap di una chiave o la si importa con la funzione "C_CreateObject", o la si clona con la funzione "C_CopyObject" l'unica configurazione che deve essere permessa è la seguente:

TEMPLATE
MODIFIABLE = FALSE EXTRACTABLE = FALSE SENSITIVE = TRUE WRAP = FALSE UNWRAP = TRUE ENCRYPT = TRUE DECRYPT = FALSE

Tabella 2.A

Spiegazione:

- ✓ MODIFIABLE = FALSE rende Sticky gli attributi e non consente il cambio degli attributi con il metodo nativo di PKCS#11.
- ✓ EXTRACTABLE = FALSE consente di mantenere il controllo sulla fruizione delle chiavi impedendo che la chiave, una volta scambiata con un utente, questi la scambi con una terza parte dato che non permette alla chiave di essere wrappata.
- ✓ WRAP = FALSE non permette di usare la chiave appena importata per wrappare altre chiavi dentro il Token.
- ✓ UNWRAP = TRUE consente di usare la chiave per ricevere chiavi da un utente
- ✓ DECRYPT = FALSE evito che la chiave possa essere usata per decifrare una chiave wrappata

In modo analogo se viene importata una chiave pubblica, dev'essere rispettato il seguente template:

TEMPLATE
MODIFIABLE = FALSE EXTRACTABLE = FALSE WRAP = FALSE ENCRYPT = TRUE

Tabella 2.B

Spiegazione:

- ✓ MODIFIABLE = FALSE rende Sticky gli attributi e non consente il cambio degli attributi con il metodo nativo di PKCS#11
- ✓ WRAP = FALSE non permette di usare la chiave appena importata per wrappare altre chiavi dentro il Token.

NOTA: tale configurazione degli attributi consente d'impedire il cambio del loro valore mediante il "wrap/unwrap trick". In questo modo vengono preservati i template delle chiavi crittografiche (generate o importate) presenti nelle precedenti tabelle.

ATTACCHI

Questa configurazione non permette gli attacchi wrap/decrypt

Attacco a chiave singola	
Algoritmo	Descrizione
$\text{Wrap}(K1, K1) = \{ K1 _{K1}\}$ $\text{Decrypt}(\{ K1 _{K1}\}, K1) = K1$	<p>Non è possibile dato che per farlo la chiave deve essere contemporaneamente di WRAP e di DECRYPT (Tabella 1.A). E' possibile cambiare gli attributi di K1? Ci sono due possibilità: o con il metodo nativo di PKCS#11 o con il trucco wrap/unwrap.</p> <p>Metodo nativo: non è possibile, MODIFIABLE è settato a FALSE Wrap/unwrap: la chiave è generata dentro il token e quindi può essere wrappata da un'altra chiave, tuttavia quando si esegue la unwrap il token deve impostare come attributi quelli indicati nella tabella 2.A e quindi con DECRYPT viene impostato a FALSE. Dato che K1 ha l'attributo SENSITIVE settato a TRUE non è possibile per il nemico attuare una decrypt offline di $\{ K1 _{K1}\}$ dato che non conosce il valore di K1</p>

Attacco "key separation"	
Algoritmo	Descrizione
$Wrap(K1, K2) = \{ K1 _{K2}\}$ $Decrypt(\{ K1 _{K2}\}, K2) = K1$	Come per il precedente non è possibile dato che K2 deve essere di WRAP e di DECRYPT e per i motivi di prima non è mai attuabile questa configurazione

Attacco a rinomina	
Algoritmo	Descrizione
$Unwrap(Ke _{K2}, K2) = Ke1$ set: $Ke1_{WRAP = TRUE}$ $Unwrap(Ke _{K2}, K2) = Ke2$ set: $Ke2_{DECRYPT = TRUE}$ $Wrap(K1, Ke1) = K1 _{Ke1}$ $Decrypt(K1 _{Ke1}, Ke2) = K1$	Mai verificabile dato che quando una chiave viene importata nel token, questa assume il template visto in tabella 2.A, quindi Ke1 non potrà mai avere l'attributo WRAP settato a TRUE e Ke2 non potrà mai avere l'attributo DECRYPT settato a TRUE

Attacco "off line" a chiave simmetrica	
Algoritmo	Descrizione
$Import\ Ke1\ (create\ Object)$ set: $Ke1_{WRAP = TRUE}$ $Wrap(K1, Ke1) = K1 _{Ke1}$ $Decrypt(K1 _{Ke1}, Ke1) = K1$ (off line decrypt)	<p>Principio: importo dentro il token una chiave simmetrica generata esternamente ed eseguo l'operazione di WRAP della chiave K1 per poi eseguire l'operazione di DECRYPT esternamente al token</p> <p>Non è possibile, dato che quando viene importata una chiave questa segue il template della Tabella 2.A e quindi non posso usare la chiave importata per wrappare altre chiavi</p>

Attacco "off line" a chiave asimmetrica	
Algoritmo	Descrizione
$Import\ PubKe1\ (create\ Object)$ set: $PubKe1_{WRAP = TRUE}$ $Wrap(K1, Ke1) = K1 _{Ke1}$ $Decrypt(K1 _{Ke1}, Ke1) = K1$ (off line decrypt)	<p>Principio: importo dentro il token una chiave asimmetrica generata esternamente ed eseguo l'operazione di WRAP della chiave K1 per poi eseguire l'operazione di DECRYPT con la relativa chiave privata esternamente al token</p> <p>Non è possibile, dato che quando viene importata una chiave asimmetrica questa assume il template visto in Tabella 2.A e quindi non è possibile usare la chiave importata per wrappare altre chiavi</p>

In modo analogo vengono evitati gli attacchi con le chiavi asimmetriche, dato che l'importazione di una chiave pubblica in un token segue il template visto in tabella 2.B e quindi dato che WRAP è settato a FALSE, non è possibile esportare le chiavi presenti nel token usando come chiave di wrap la chiave pubblica appena importata.

LIMITI DEI TEMPLATE

L'utilizzo di una simile politica per la gestione delle chiavi pone un forte limite all'utilizzo delle chiavi asimmetriche, dato che possono essere utilizzate solo per le operazioni crittografiche sui dati e per la firma digitale.

USABILITÀ DELLA SOLUZIONE



L'utente ROSSO vuole scambiare chiavi e dati con l'utente BLU

1. ROSSO crea una chiave (KR1) con WRAP e UNWRAP settati a TRUE (Tabella 1, template 1)
2. BLU importa¹ tale chiave (KR1) e il token gli setta il template della Tabella 2.A
NOTA: ROSSO con KR1 può solo inviare chiavi a BLU dato che non è possibile per ROSSO cambiare gli attributi di KR1 e convertirla in una chiave di ENCRYPT – DECRYPT
3. ROSSO genera una nuova chiave (KR2) e la setta con gli attributi ENCRYPT e DECRYPT a TRUE (Tabella 1, template 2)
4. ROSSO usa KR1 per eseguire il wrap di KR2. BLU usa la sua copia di KR1 per eseguire l'unwrap di KR2
NOTA: ROSSO ha 2 chiavi:
 KR1 per lo scambio di chiavi con BLU
 KR2 per la comunicazione cifrata con BLU
BLU ha 2 chiavi con template identico ma diverso valore di chiave crittografica
5. Se l'utente BLU vuole mandare un messaggio cifrato a ROSSO deve usare KR2 dato che permette così a ROSSO di decifrare il messaggio (ricordo che KR2 per ROSSO ha il template encrypt – decrypt e KR2 per BLU permette solo Unwrap ed Encrypt).

E se ROSSO vuole mandare un messaggio cifrato a BLU ?? L'utente BLU farà gli stessi passi visti in precedenza:

1. BLU crea una chiave (KB1) con WRAP e UNWRAP settati a TRUE
2. ROSSO importa¹ tale chiave (KB1) e il token gli setta il template della Tabella 2.A
3. BLU con KB1 può solo scambiare chiavi con ROSSO dato che non può cambiare gli attributi di KB1 e convertirla in una chiave di ENCRYPT – DECRYPT
4. BLU genera una nuova chiave (KB2) e la setta con gli attributi ENCRYPT e DECRYPT a TRUE
5. BLU usa KB1 per scambiare KB2 con l'utente ROSSO
6. Se ROSSO vuole mandare un messaggio cifrato a BLU deve usare KB2 dato che permette così a BLU di decifrare il messaggio (ricordo che KB2 per BLU ha il template encrypt – decrypt)

(1) Affiche questa soluzione sia attuabile è necessario che i vari utenti condividano una chiave (es.: Master Key) che consenta di iniziare lo scambio dei dati. Lo standard PKCS#11 supporta Diffie-Hellman, che può essere utilizzato per la creazione di una Master Key tra i vari utenti con il template di wrap/unwrap di Tabella 1. E' anche possibile far coincidere le chiavi KR1 e KB1 con tale Master Key.