Response to the Tookan PKCS#11 Attacks for Classic TPC IS V2 and Cyberflex V2

This document is a response from Gemalto to the paper "Attacking and Fixing PKCS#11 Security Tokens," [1] that explains how some commercial security tokens are vulnerable to attacks using the PKCS#11 standard as implemented by the vendor. These vulnerabilities revolve around exposing sensitive keys in ways that were not intended. While some of the vulnerabilities are due to weaknesses in the PKCS#11 standard, others are a result of the security token not conforming to the standard. This response focuses on two Gemalto products: Classic TPC IS v2 and Cyberflex v2 smart cards.

The attacks a1 and a2 described in the paper are deficiencies in the PKCS#11 standard, not faults with the implementation of the standard. Deficiencies or not, the specification is what guides implementers to ensure their products are interoperable. Diverging from the standard runs the risk of the security token not supporting applications and thereby limiting its adoption in the industry. However, there are ways to mitigate the vulnerabilities using industry best practices suitable for PKCS#11.

This document is organized in two parts, one covering the Gemalto Classic TPC IS v2 case, another one covering the Gemalto Cyberflex v2 case.

Gemalto Classic Client and TPC IS v2 Smart Card

The paper is describing some attacks using the PKCS#11 library provided by the Gemalto Classic Client with the Classic TPC IS v2 smartcard. Classic Client implements a subset of the PKCS#11 specifications v2.01 to interface with the Classic TPC IS v2 card.
The attacks mentioned in the Tookan document are discussed in the following paragraphs.

a1: wrap/decrypt attack based on symmetric keys

a1 attack is a key separation (wrap & decrypt) attack on a secret key. A secret key that can wrap and decrypt is used to wrap a sensitive key. A further decryption operation would reveal the sensitive key value.

The a1 attack is due to a deficiency in the PKCS#11 specifications, which do not enforce the separation of key roles to encrypt/decrypt data and wrap/unwrap.

Additionally, in the case of Classic Client with Classic TPC IS v2, two types of keys are supported:
- Session secret keys (managed in the host memory, not in the smartcard)
- Token RSA key pairs (RSA keys stored in the smartcard)

The only use case which corresponds to the attack a1 is the wrap of a session secret key1, by another session secret key2. Session secret key2 is used to wrap session secret key1, itself used to encrypt/decrypt data.
This use case, despite described in the PKCS#11 specifications, is actually not used by the applications.

[1] Bortolozzo, M. et al. "Attacking and Fixing PKCS#11 Security Tokens." 2010. MS. Http://secgroup.ext.dsi.unive.it/wp-content/uploads/2010/10/Tookan-CCS10.pdf. Secgroup. 06 Oct. 2010. Web. 30 Nov. 2010.

It is important to remind that the RSA private keys (the sensitive part of the token RSA key pairs) can never be exported from the smartcard. This is a key security feature for all Gemalto cards.

a3: sensitive keys are directly readable

As for attack a1, attack a3 is related only to PKCS#11 session secret key objects, which can only be used to encrypt/decrypt data.
That kind of objects, despite described in the PKCS#11 specifications, is actually not used by the applications.

Nevertheless this a3 attack revealed a non conformity of the Classic Client PKCS#11 library for this specific feature, which has been corrected.

The design of the Classic TPC IS v2 ensures that a RSA private key stored in the smartcard can never be exported outside the smartcard.

a4: unextractable keys are directly readable

As for attack a1, attack a4 is related only to PKCS#11 session secret key objects, which can only be used to encrypt/decrypt data.
That kind of objects, despite described in the PKCS#11 specifications, is actually not used by the applications.

Nevertheless this a4 attack revealed a non conformity of the Classic Client PKCS#11 library for this specific feature, which has been corrected.

The design of the Classic TPC IS v2 ensures that a RSA private key stored in the smartcard can never be exported outside the smartcard.

a5: sensitive/unextractable keys can be changed into nonsensitive/extractable

As for attack a1, attack a5 is related only to PKCS#11 session secret key objects, which can only be used to encrypt/decrypt data.
That kind of objects, despite described in the PKCS#11 specifications, is actually not used by the applications.

Nevertheless this a5 attack revealed a non conformity of the Classic Client PKCS#11 library for this specific feature, which has been corrected.

The design of the Classic TPC IS v2 ensures that a RSA private key stored in the smartcard can never be exported outside the smartcard.

Conclusion for Classic TPC IS v2

The attacks mentioned in the “Attacking and Fixing PKCS#11 Security Tokens” paper, for the Gemalto Classic TPC IS v2 token, are all related to the feature “wrap of a session encryption/decryption secret key by another session secret key”.
a1 attack is due to a deficiency of the PKCS#11 specifications.

a3, a4 and a5 attacks are due to some non conformity to the PKCS#11 standard, which have been corrected.

The related use case, despite described in the PKCS#11 specifications, is actually not used by the applications.

None of these attacks are involving the RSA private keys (the sensitive part of the token RSA key pairs), which, by design, can never be exported from the smartcard.

Gemalto Access Client with Cyberflex V2

The paper describes an attack (a2) using the PKCS#11 library provided by the Gemalto Access Client with the Cyberflex V2 smart card. This attack involves revealing a secret key that is wrapped by an asymmetric key. While the paper does not specify the version, the Access Client v5.3 and later series is assumed in this analysis. This Gemalto suite implements a subset of the PKCS#11 specifications v2.01 to interface with the Cyberflex V2 card.

The a2 attack illustrates a key separation vulnerability due to a deficiency in the PKCS#11 specifications. In this case, the PKCS#11 specifications do not enforce the separation of key roles to encrypt/decrypt data and wrap/unwrap. In addition, the standard does not provide a way to distinguish a wrapped key pair from encrypted data. Together, the deficiencies in the specification allow secret keys to be exposed using a key pair with the ability to both unwrap and decrypt.

With this fundamental weakness in the specifications, this attack can be avoided by using industry best practices. In this case, the wrapping key should not have both the CKA_UNWRAP and CKA_DECRYPT attributes set to TRUE. Where possible, an additional technique to avoid the attack is to generate the secret key pair on the smart card rather than downloading the wrapped secret key pair. Applications written using these conventions can be secure while the PKCS#11 implementation can remain interoperable.