3rd International Workshop

Analysis of Security APIs

July 10 -11 2009

Port Jefferson (New York - USA)

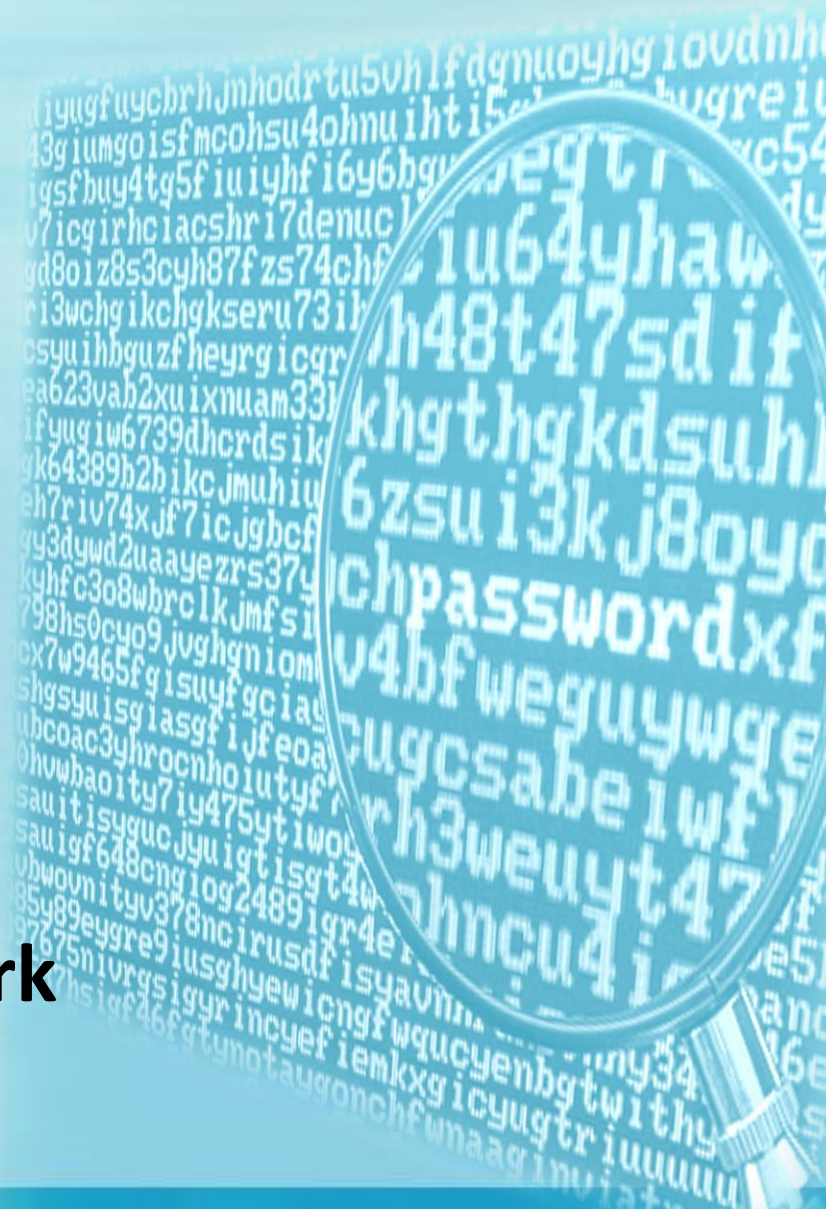
Secure your PKCS#11 token against API attacks!

(Work partially supported by Miur'07 Project SOFT: Security Oriented Formal Techniques)

Authors:

Bortolozzo Matteo (left) (speaker)
Marchetto Giovanni (right)
Focardi Riccardo
Graham Steel

TABLE OF CONTENTS



THE PKCS#11 STANDARD

P.K.C.S. is an acronym:

"Public Key Cryptography Standards"

This standard is developed by RSA Inc. for Token management

PKCS#11 describes:

- ✓ **Asymmetric cryptography**
- ✓ **Symmetric cryptography**
- ✓ **Digital signature**

THE PKCS#11 STANDARD

The targets of PKCS#11 are:

- ✓ **Provide a common interface for hardware Tokens**

(device interoperability)

- ✓ **Provide a secure device for data transfer**

(e.g., secure secret key transfer)

- ✓ **Provide a secure and protected system**

(the token is secure and it works in a insecure context)

THE PKCS#11 STANDARD

There are three types of Token object:

Data: user data (e.g., documents)

Certificate: digital certificates

Key: cryptographic keys

(the attacks involve the keys)

THE PKCS#11 STANDARD

PKCS#11 defines some operations:

- ✓ **WRAP: key encryption**
- ✓ **UNWRAP: key decryption**
- ✓ **ENCRYPT: data encryption**
- ✓ **DECRYPT: data decryption**

THE PKCS#11 STANDARD

Every token object has some attributes

Common key attributes:

- ✓ SENSITIVE
- ✓ EXTRACTABLE
- ✓ WRAP
- ✓ UNWRAP
- ✓ ENCRYPT
- ✓ DECRYPT

NOTE: attributes are modifiable after key creation

THE ATTACKS

Attack definition:

"The hardware security modules (HSMs) revealing their secrets by sending unusual sequences of commands …"

M. Bond

TARGET KEY ATTRIBUTES:

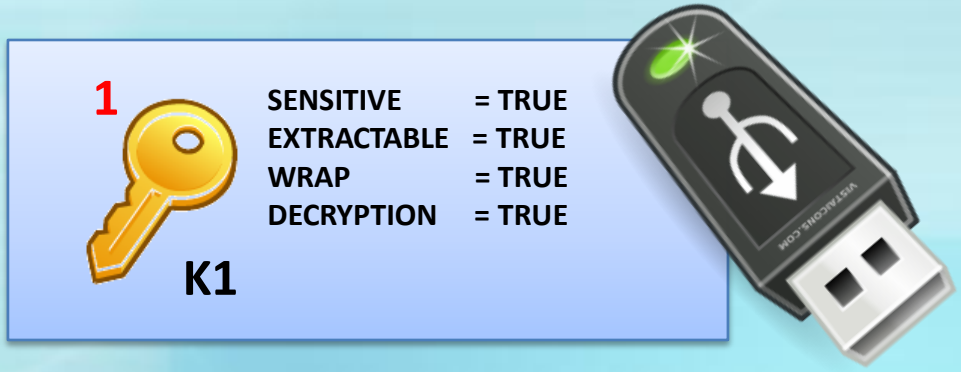
SENSITIVE	= TRUE
EXTRACTABLE	= TRUE
WRAP	= TRUE
DECRYPTION	= TRUE

SINGLE KEY ATTACK

ATTACK SEQUENCE:

1. K1 = "Target key"
2. WrappedKey = WRAP (K1, K1)
3. K1 = DECRYPT (WrappedKey, K1)

THE ATTACKS



ATTACK SEQUENCE:

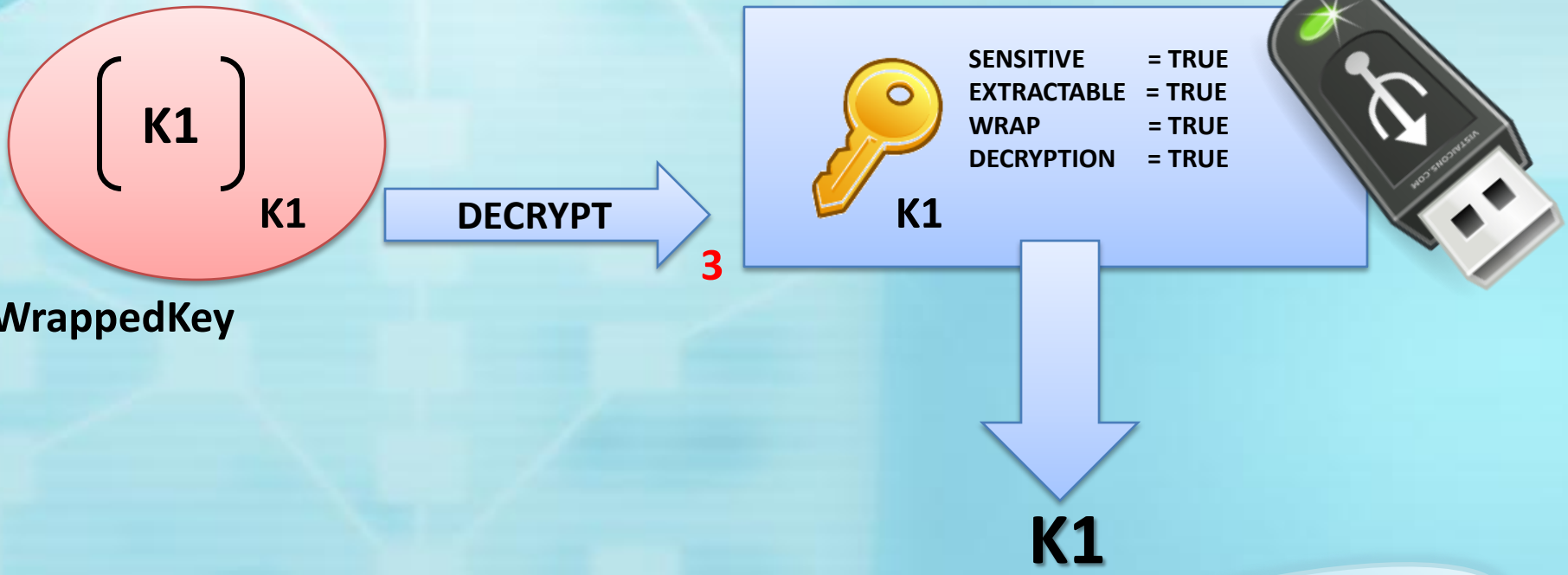
1. K1 = “Target key”
2. WrappedKey = WRAP (&K1, &K1)
3. K1 = DECRYPT (WrappedKey, &K1)

THE ATTACKS

ATTACK SEQUENCE:

1. K1 = “Target key”
2. WrappedKey = WRAP (&K1, &K1)
3. K1 = DECRYPT (WrappedKey, &K1)

THE ATTACKS



ATTACK SEQUENCE:

1. K1 = “Target key”
2. WrappedKey = WRAP (&K1, &K1)
3. K1 = DECRYPT (WrappedKey, &K1)

THE ATTACKS

TARGET KEY ATTRIBUTES:

SENSITIVE = TRUE

EXTRACTABLE = TRUE

WRAP KEY ATTRIBUTES:

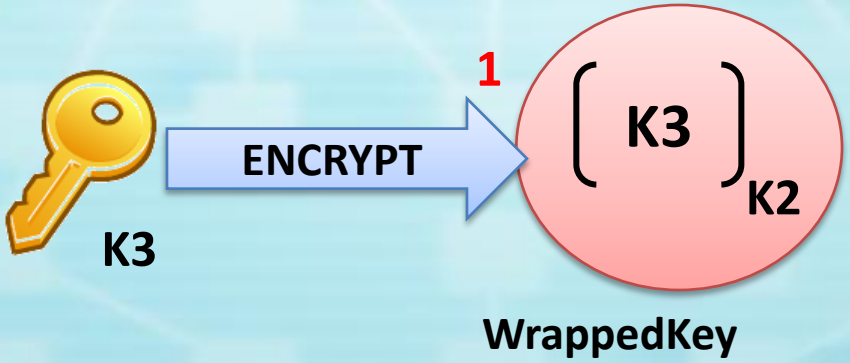
ENCRYPT = TRUE

UNWRAP = TRUE

ATTACK SEQUENCE:

1. K1 = “Target key”
2. K2 = “Wrap key”
3. K3 = “Enemy key”
4. WrappedKey = ENCRYPT (K3, &K2)
5. K4 = UNWRAP (WrappedKey, &K2)
6. K5 = UNWRAP (WrappedKey, &K2)
7. NewWrappedKey = WRAP (&K1, &K4)
8. K1 = DECRYPT (NewWrappedKey, &K5)

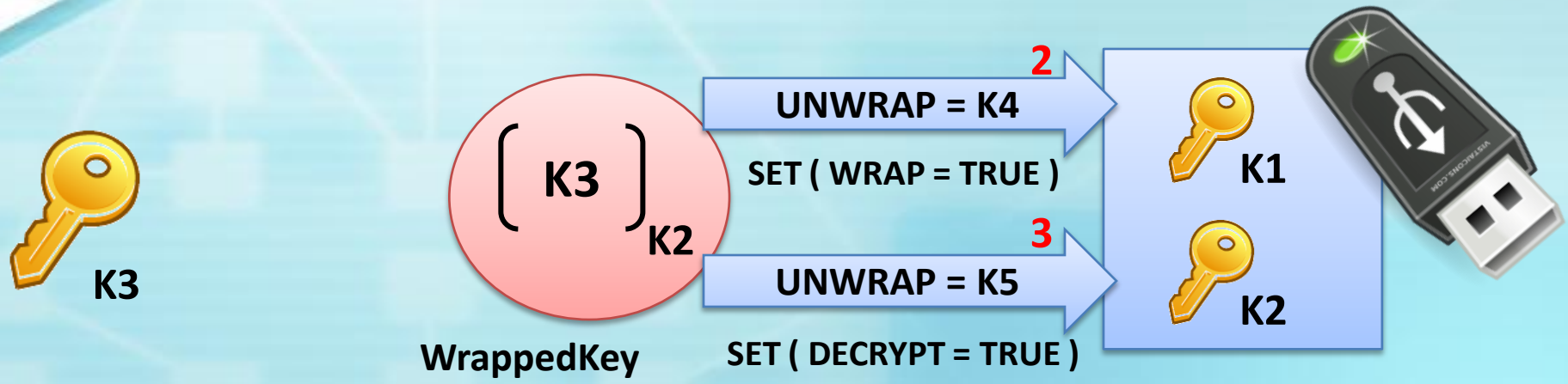
THE ATTACKS



ATTACK SEQUENCE:

1. WrappedKey = ENCRYPT (K3, &K2)
2. K4 = UNWRAP (WrappedKey, &K2)
3. K5 = UNWRAP (WrappedKey, &K2)
4. NewWrappedKey = WRAP (&K1, &K4)
5. K1 = DECRYPT (NewWrappedKey, &K5)

THE ATTACKS

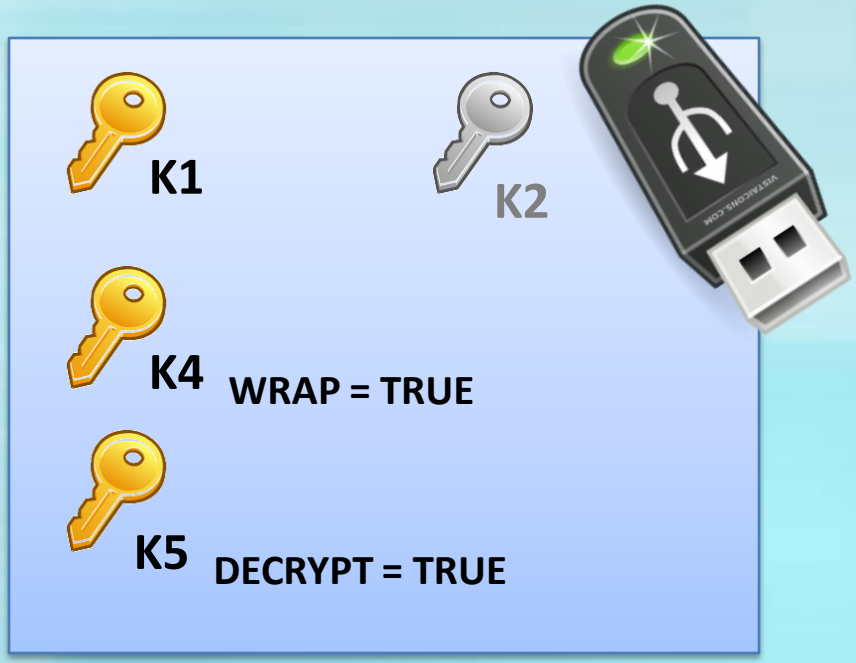


ATTACK SEQUENCE:

1. WrappedKey = ENCRYPT (K3, &K2)
2. K4 = UNWRAP (WrappedKey, &K2)
3. K5 = UNWRAP (WrappedKey, &K2)
4. NewWrappedKey = WRAP (&K1, &K4)
5. K1 = DECRYPT (NewWrappedKey, &K5)

"THREE KEY ATTACK WITH KEY RENAME"

THE ATTACKS



ATTACK SEQUENCE:

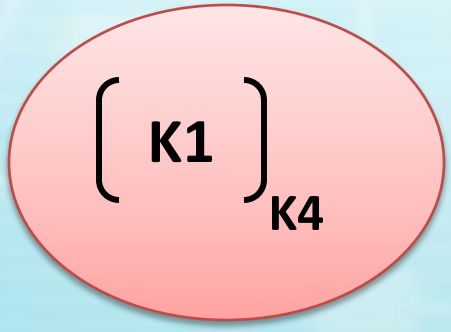
1. WrappedKey = ENCRYPT (K3, &K2)
2. &K4 = UNWRAP (WrappedKey, &K2)
3. &K5 = UNWRAP (WrappedKey, &K2)
4. NewWrappedKey = WRAP (&K1, &K4)
5. K1 = DECRYPT (NewWrappedKey, &K5)

THE ATTACKS

ATTACK SEQUENCE:

1. WrappedKey = ENCRYPT (K3, &K2)
2. &K4 = UNWRAP (WrappedKey, &K2)
3. &K5 = UNWRAP (WrappedKey, &K2)
4. NewWrappedKey = WRAP (&K1, &K4)
5. K1 = DECRYPT (NewWrappedKey, &K5)

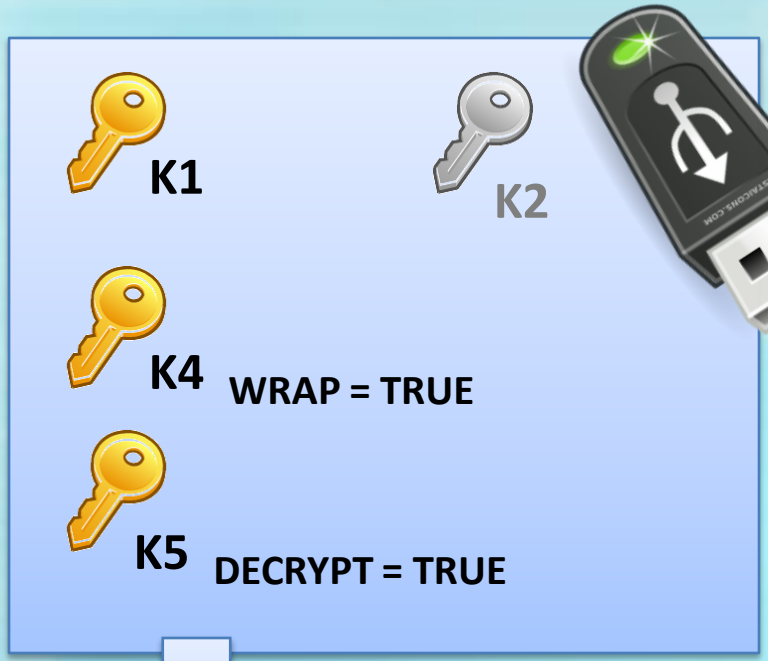
THE ATTACKS



NewWrappedKey

DECRYPT

5



ATTACK SEQUENCE:

1. WrappedKey = ENCRYPT (K3, &K2)
2. &K4 = UNWRAP (WrappedKey, &K2)
3. &K5 = UNWRAP (WrappedKey, &K2)
4. NewWrappedKey = WRAP (&K1, &K4)
5. K1 = DECRYPT (NewWrappedKey, &K5)

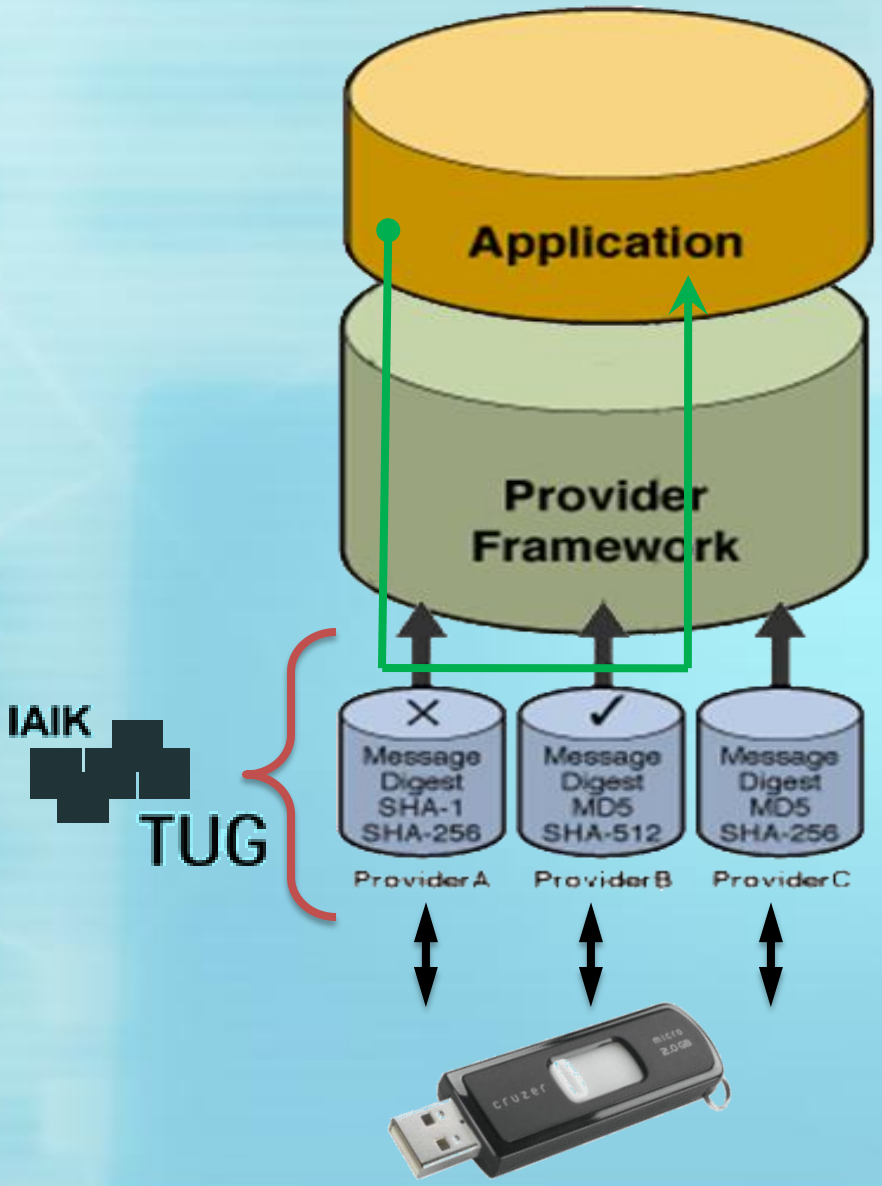
K1

"THREE KEY ATTACK WITH KEY RENAME"

IAIK LIBRARY

The IAIK library:

- ✓ **Is a university of Graz (Austria) project**
- ✓ **Does not implement PKCS#11 functionalities**
- ✓ **Is a "bridge" between Java and PKCS#11**



THE API ATTACKs!

Main window:

- ✓ Shows token functions
- ✓ Shows token objects
- ✓ Gives access to:
  - ➢ Key management
  - ➢ Digital certificate creation
  - ➢ The Attack window

THE API ATTACKs!

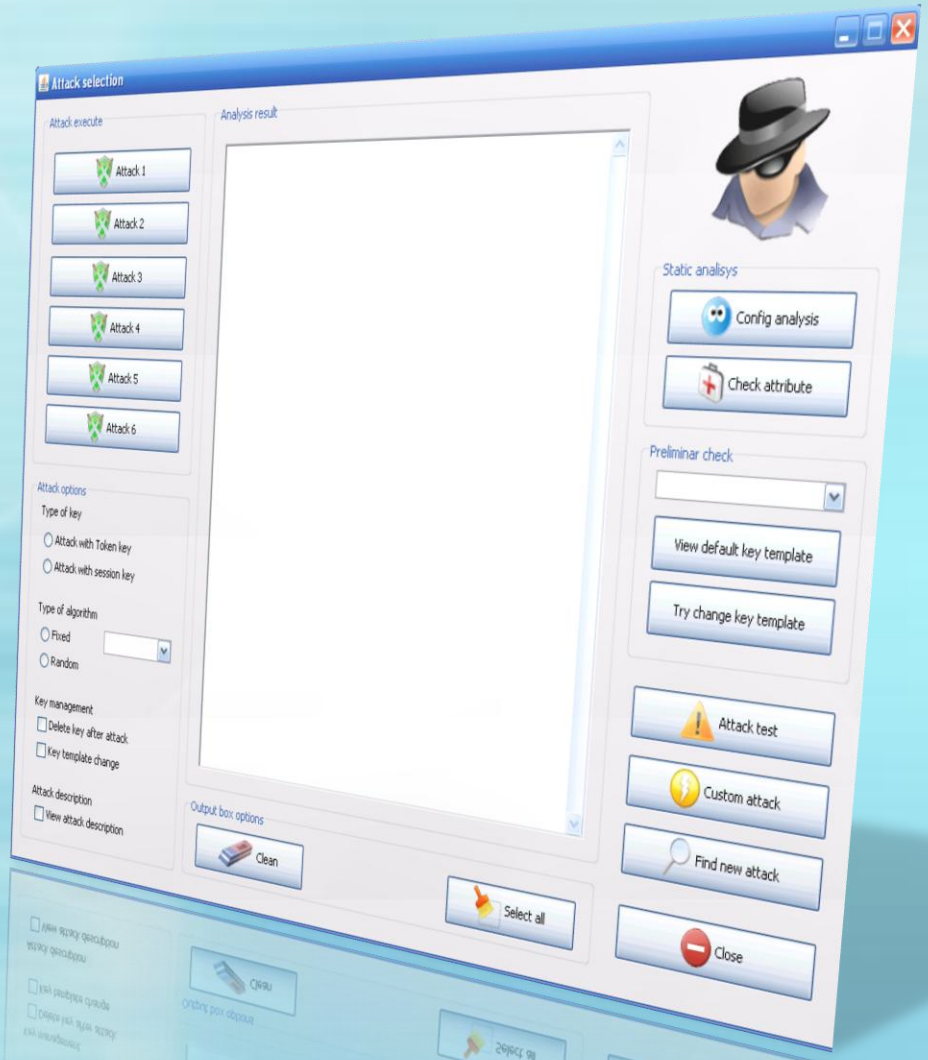
Key management:

- ✓ Asymmetric key generation
- ✓ Symmetric key generation
- ✓ Key attribute inspection
- ✓ Key template change
- ✓ key and object deletion

THE API ATTACKs!

Attacks management

- ✓ Known attacks execution
- ✓ Custom attacks execution
- ✓ Static keys analysis
- ✓ New attack discovery

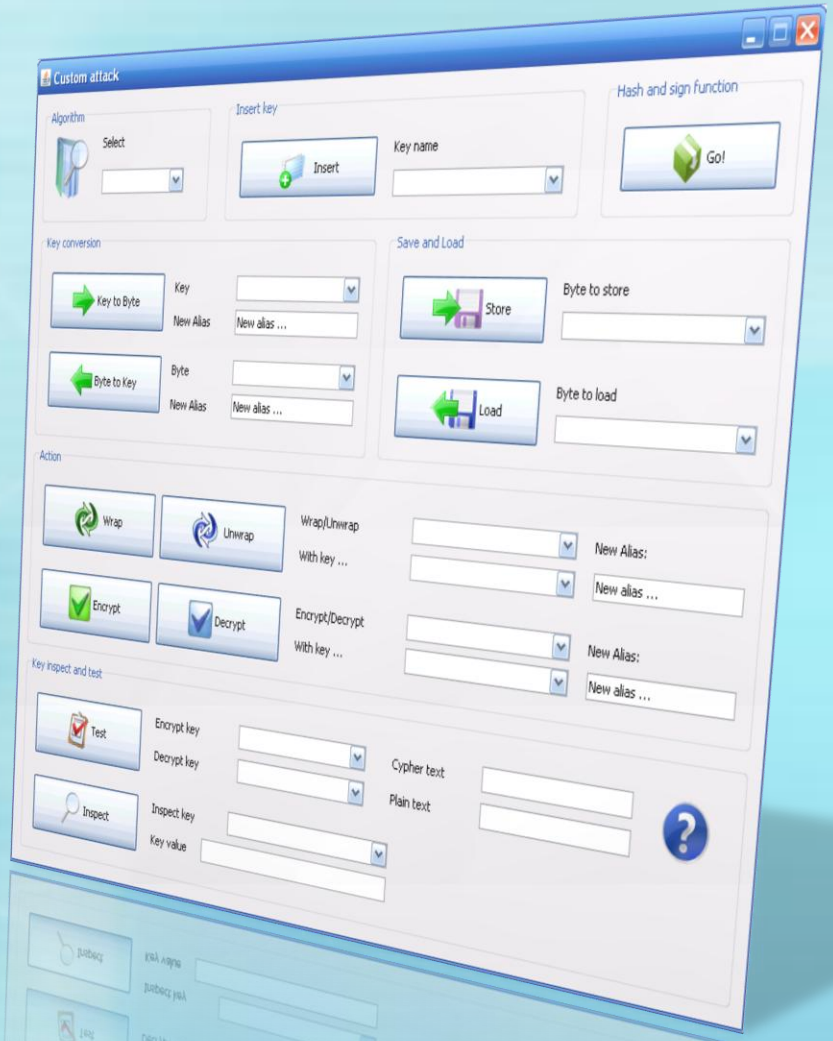


THE API ATTACKs!

Custom attacks window

Execute operations:

- ✓ **Wrap**
- ✓ **Unwrap**
- ✓ **Encrypt**
- ✓ **Decrypt**

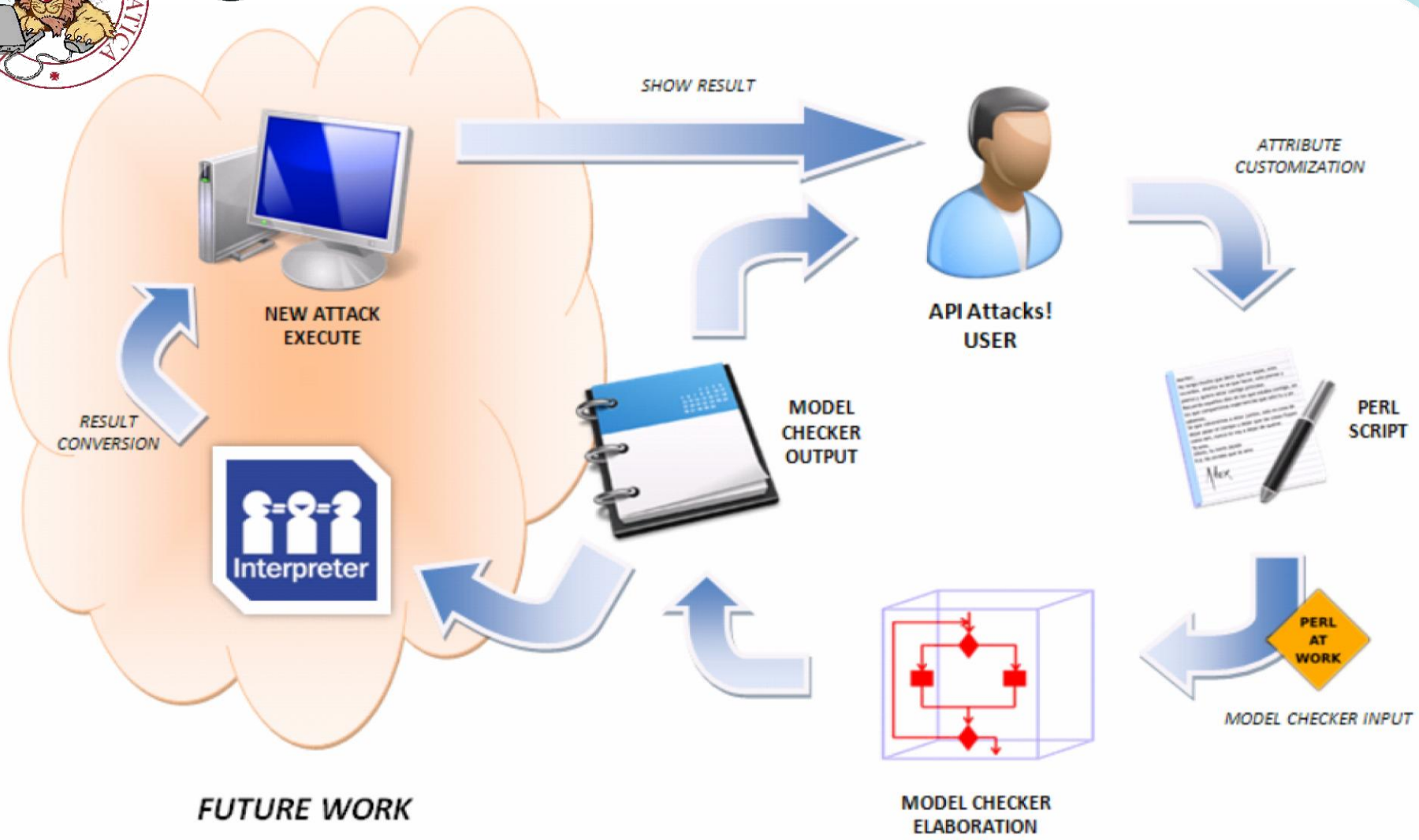


THE API ATTACKs!

… show API Attaks!

in action …

THE MODEL CHECKER



CURRENT AND FUTURE WORK

The current work:

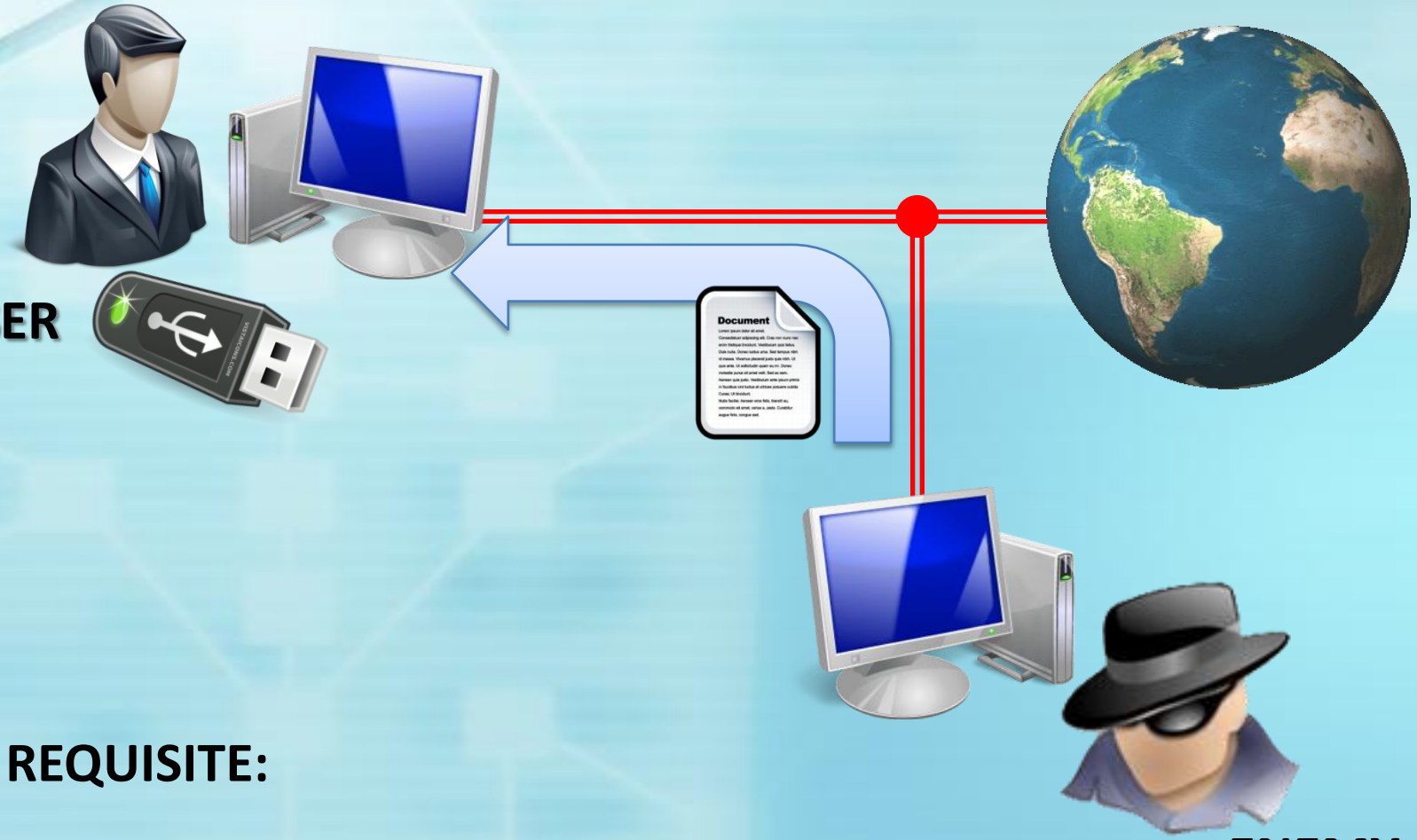
- ✓ **Improve the tool flexibility**
- ✓ **Model checker optimization**
- ✓ **Study different type of attacks**

Different type of attacks ??

- ➢ **e.g., launch PKCS#11 command from remote workstation, replace key, sign enemy document, …**

View an example …

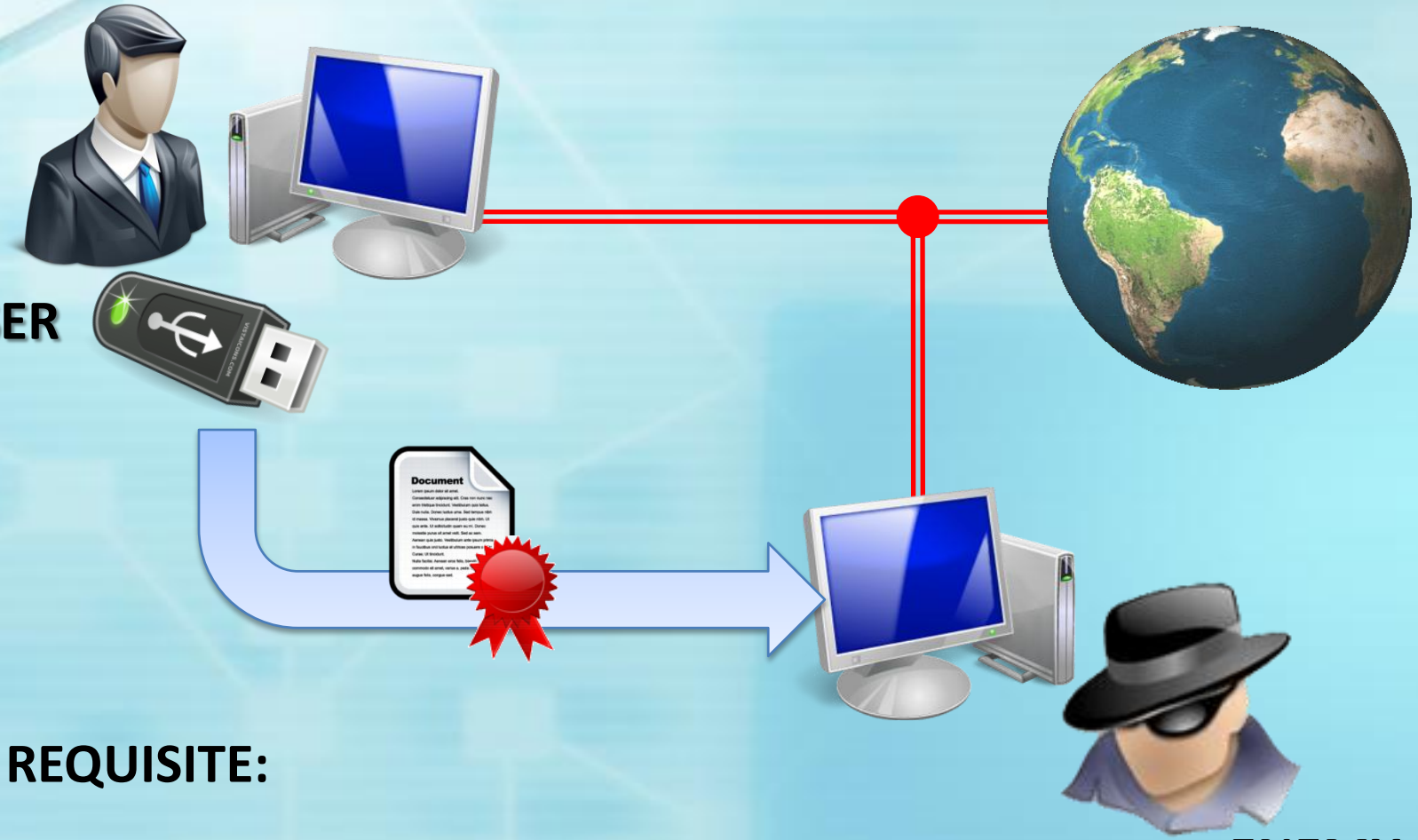
CURRENT AND FUTURE WORK



ATTACK REQUISITE:

- ✓ The enemy has got a control of user workstation
- ✓ The enemy can intercept the user PIN

CURRENT AND FUTURE WORK



ATTACK REQUISITE:

- ✓ The enemy has got a control of user workstation
- ✓ The enemy can intercept the user PIN

QUESTION

Secure your PKCS#11 token against API attacks!