Synthesising Secure APIs

(to appear at ESORICS'09)

Véronique Cortier and Graham Steel

LORIA, Projet Cassis, CNRS & INRIA

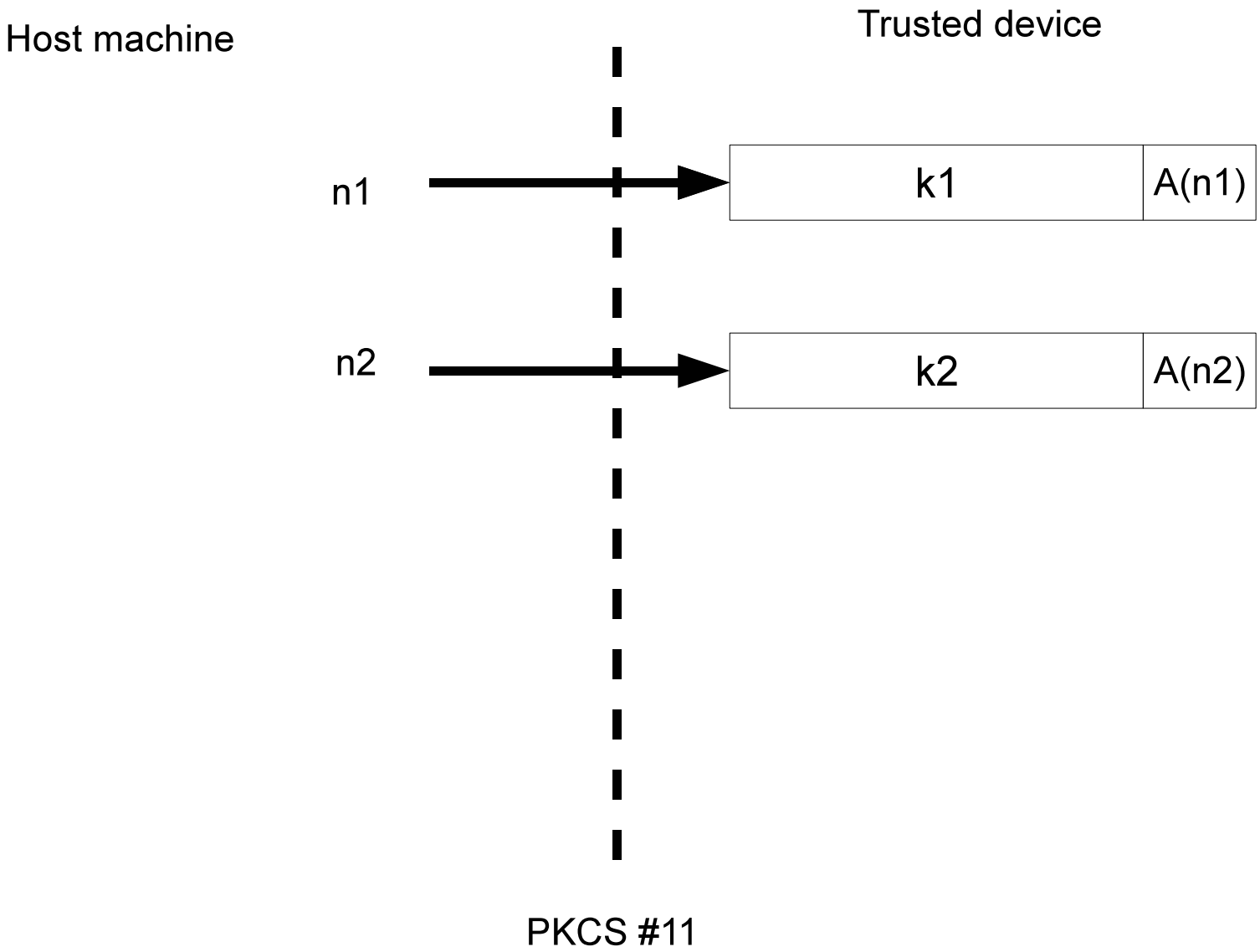
and

LSV, INRIA & CNRS & ENS-Cachan

Host machine

Trusted device

Security API


Host machine
Trusted device
n1
k1
A(n1)
n2
k2
A(n2)
PKCS #11

Security Policy

Security Policy

e.g. from PKCS#11 (1 page in 407 page standard)

Security Policy

e.g. from PKCS#11 (1 page in 407 page standard)

“the PIN may be passed through the operating system. This can make it easy for a rogue application on the operating system to obtain the PIN

Rogue applications and devices may also change the commands sent to the cryptographic device to obtain services other than what the application requested.

Security Policy

e.g. from PKCS#11 (1 page in 407 page standard)

“the PIN may be passed through the operating system. This can make it easy for a rogue application on the operating system to obtain the PIN

Rogue applications and devices may also change the commands sent to the cryptographic device to obtain services other than what the application requested.

We note that none of the attacks just described can compromise keys marked “sensitive,” since a key that is sensitive will always remain sensitive.”

Key Separation Attack (Clulow, 2003)

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

State: $\mathsf{wrap}(n_2)$, $\mathsf{decrypt}(n_2)$, $\mathsf{sensitive}(n_1)$, $\mathsf{extract}(n_1)$

Wrap: $h(n_2, k_2), h(n_1, k_1) \rightarrow \{k_1\}_{k_2}$

Decrypt: $h(n_2, k_2), \{k_1\}_{k_2} \rightarrow k_1$

Many variations of this attack exist, not easy to fix (see Delaune, Kremer & Steel CSF'08)

Protocols

Security APIs are used to implement endpoints of protocols

Protocols

Security APIs are used to implement endpoints of protocols

Idea:

Abduct security API policy from suite of protocols it is supposed to implement

Protocols

Security APIs are used to implement endpoints of protocols

Idea:

Abduct security API policy from suite of protocols it is supposed to implement

Design generic API that can be instantiated to any protocol

Prove properties for the generic API that hold no matter what protocol is implemented

Generic API: Concepts

Handles for objects stored on device

Static security types for objects on device:

0 Public data (exists outside device)

1 Secret on device, not usable as key

2 Short term (session) key

3 Long term (key transport) key

Agent identifiers

Generic API: Generate Commands

$$\overset{N,K}{\Rightarrow} K_a(h_a^g(N,K,i,S)) \quad i \geq 1 \qquad \textbf{(Secure Generate)}$$

$$\overset{N,K}{\Rightarrow} K_a(K), K_a(h_a^g(N,K,0,\emptyset)) \qquad \textbf{(Public Generate)}$$

where $N \in$ VarNonce, and $K \in$ VarNonce if $i = 1$, $K \in$ VarKey if $i = 2$.

Generic API: Encrypt

$$K_a(h_a^{\alpha}(X,K,i_0,S_0)),K_a(x_1),\ldots,K_a(x_k),$$
$$K_a(h_a^{\alpha_1}(X_{n_1},y_1,i_1,S_1)),\ldots,K_a(h_a^{\alpha_l}(X_{n_l},y_l,i_l,S_l))$$
$$\Rightarrow K_a(\{x_1,0,\ldots,x_k,0,y_1,i_1,S_1,\ldots,y_l,i_l,S_l\}_K) \quad \textbf{(Encrypt)}$$

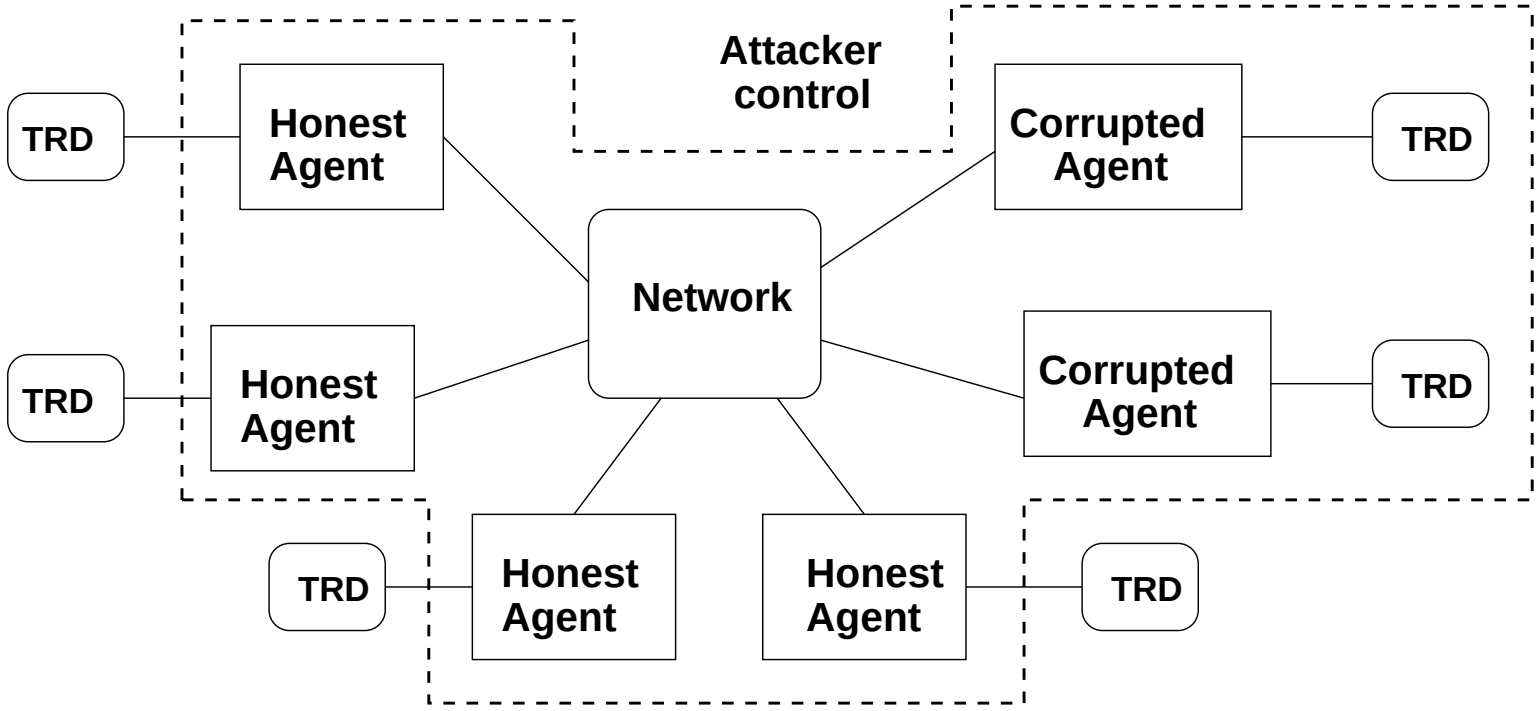
Require $i_0 > i_j$ (keys only encrypt data of strictly lower security level) and $S_0 \subseteq S_j$

Generic API: Decrypt

$$K_a(h_a^{\alpha}(X,K,i_0,S_0)), K_a(\{x_1,0,\ldots,x_k,0,y_1,i_1,S_1,\ldots,y_l,i_l,S_l\}_K),$$
$$K_a(h_a^{g}(X_1,x_1,0,\emptyset)),\ldots,K_a(h_a^{g}(X_s,x_s,0,\emptyset)),$$
$$K_a(h_a^{g}(Y_1,y_1,i_1,S_1)),\ldots,K_a(h_a^{g}(Y_l,y_l,i_l,S_l))$$
$$\stackrel{N_{r+1},\ldots,N_l}{\Rightarrow} K_a(x_{s+1})\ldots,K_a(x_k),$$
$$K_a(h^r(N_{r+1},y_{r+1},i_{r+1},S_{r+1}))\ldots,K_a(N_l,y_l,i_l,S_l) \quad \textbf{(Decrypt/Test)}$$

provided $i_0 > i_j$ (keys only encrypt data of strictly lower security level) and that $S_0 \subseteq S_j$

Properties: Scenario



Property 1 : Secrecy Invariant

"Secret data of honest users should not be known to the intruder."

i.e. data values k for which there are handles of the form $h_a^{\alpha}(n,k,i,S)$, where S is a subset of honest users, are unknown to the intruder.

$$\forall a \in \mathsf{Agent}, \forall x,y \in \mathsf{msg}, \forall i \in \{1,2,3\}, \forall \alpha \in \{r,g\}, \forall S \subseteq H$$
$$\mathcal{S} \vdash h_a^{\alpha}(x,y,i,S) \Rightarrow \mathcal{S} \nvdash y \text{ and } y \in \mathsf{Key} \cup \mathsf{Nonce} \quad \textbf{(Sec)}$$

Property 2 : Secrecy After Compromise

Intruder learns some session keys

Honest users erase old handles

API enforces tests - no session keys accepted without a test

Sec is still preserved

Implementing Protocols

Protocol (section in Clark-Jacob)	API	API^r
Needham-Schroeder SK (6.3.1)	+	-
NSSK amended version (6.3.4)	+	+
Otway-Rees (6.3.3)	+	+
Yahalom (6.3.6)	+	-
Carlsen (6.3.7)	+	+
Woo-Lam Mutual Auth (6.3.11)	+	+

http://www.lsv.ens-cachan.fr/~steel/genericapi/

Related Work

Cachin and Tandran, to appear at CSF ’09

- Cryptographic model
- Only one token
- Multiple user profiles on token
- Highly restrictive (un)wrapping commands

Further Work

- Cryptographic soundness
- Long-term key update
- PKI (asymmetric crypto, certificates,. . .)
- Timestamps

PUB

3rd Analysis of Security APIs workshop (ASA-3)

Long Island, New York, July 10-11 2009

(satellite of IEEE CSF)

http://www.lsv.ens-cachan.fr/~steel/asa3