Monotonicity for Multiple Key Management Tokens

Johannes Borgström

ASA'09

PKCS #11

- This slide left blank on purpose

Single Token Assumption

- “By storing a history of all operations in the token, we are able to keep track of all dependencies and avoid unexpected consequences of an operation.”
- What happens if there is a second token?

Usage Scenarios

- "We attempt to prevent the attack [...] by adding wrap and unwrap to our list of conflicting attribute pairs."

- Global interpretation:
  - Invalidates real-world usage scenarios.

- Local interpretation:
  - Does not prevent attack.

Non-Monotonic Operations

- Constrains future adversary actions.

- Examples:
  - Creating a key dependency (cf. Slide 3)
  - Setting a sticky attribute that has conflicts (cf. slide 4)

- If there are multiple copies of a key,
  - all bets are off.

Attacks

- Key assumption:

  Given access to decryption oracle, the adversary can unwrap a key.

- => Model wrapping as encryption.

$$
\begin{array}{llll}
\text{Handles} & g, h \in \mathbf{H} & & \\
\text{Keys} & k, l \in \mathbf{K} & & \\
\text{Permissions} & p, q \in \mathbf{P} & := & \{\mathsf{enc}, \mathsf{dec}, \mathsf{wrap}, \mathsf{unwrap}\} \\
\text{Tokens} & M \in \mathbf{M} & := & \mathbf{H} \rightarrow^{\mathsf{fin}} \mathbf{K} \times 2^{\mathbf{P}}
\end{array}
$$

Microsoft Research

Operational Model

$\texttt{enc}\ t\ \texttt{with}\ h:$

$$M; h, t \;\rightarrow\; M; \mathrm{enc}(t,k) \qquad \text{if } M(h) = (k, p \cup \mathsf{enc})$$

$\texttt{dec}\ \mathrm{enc}(t,k)\ \texttt{with}\ h:$

$$M; h, \mathrm{enc}(t,k) \rightarrow M; t \qquad \text{if } M(h) = (k, p \cup \mathsf{dec})$$

$\texttt{wrap}\ g\ \texttt{with}\ h:$

$$M; g, h \rightarrow M; \mathrm{enc}(l,k) \qquad \text{if } M(h) = (k, p \cup \mathsf{wrap}) \text{ and } M(g) = (l, q)$$

$\texttt{unwrap}\ \mathrm{enc}(l,k)\ \texttt{with}\ h\ \texttt{for}\ q:$

$$M; h, \mathrm{enc}(l,k) \xrightarrow{g} M'; g \qquad \text{if } M(h) = (k, p \cup \mathsf{unwrap}) \text{ and } M' = M \cup \{g \mapsto (l,q)\}$$

$$\frac{M; S \xrightarrow{H} M'; S' \text{ and } H \cap \mathrm{consts}(S \cup \mathrm{dom}(M)) = \emptyset}{\mathcal{M} \uplus M; S \cup S_0 \xrightarrow{H} \mathcal{M} \uplus M'; S \cup S_0 \cup S'}$$

Scenario

$$
\begin{array}{rcl}
M_1 & = & \{h_1 \mapsto (k, \mathsf{wrap}), g_1 \mapsto (l, \mathsf{dec})\} \\
M_2 & = & \{h_2 \mapsto (k, \mathsf{unwrap})\} \\
S & = & \{h_1, h_2, g_1, d\}
\end{array}
$$

$$M_1, M_2; S$$

$1 : \texttt{wrap } g_1 \texttt{ with } h_1$

$$\to M_1, M_2; S \cup \{\mathrm{enc}(l,k)\}$$

$2 : \texttt{unwrap } \mathrm{enc}(l,k) \texttt{ with } h_2 \texttt{ for enc}$

$$\xrightarrow{g_2} M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{enc})); S \cup \{\mathrm{enc}(l,k), g_2\}$$

$2 : \texttt{enc } d \texttt{ with } g_2$

$$\to M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{enc})); S \cup \{\mathrm{enc}(l,k), g_2, \mathrm{enc}(d,l)\}$$

$1 : \texttt{dec } \mathrm{enc}(d,l) \texttt{ with } g_1$

$$\to M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{enc})); S \cup \{\mathrm{enc}(l,k), g_2, \mathrm{enc}(d,l)\}$$

Attack 1

$$
\begin{aligned}
M_1 &= \{h_1 \mapsto (k, \mathsf{wrap}), g_1 \mapsto (l, \mathsf{dec})\} \\
M_2 &= \{h_2 \mapsto (k, \mathsf{unwrap})\} \\
S &= \{h_1, h_2, g_1, d\}
\end{aligned}
$$

$$M_1, M_2; S$$

$1 : \texttt{wrap}\ g_1\ \texttt{with}\ h_1$

$$\to M_1, M_2; S \cup \{\mathrm{enc}(l, k)\}$$

$2 : \texttt{unwrap}\ \mathrm{enc}(l, k)\ \texttt{with}\ h_2\ \texttt{for}\ \mathsf{wrap}$

$$\xrightarrow{g_2} M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{wrap})); S \cup \{\mathrm{enc}(l, k), g_2\}$$

$2 : \texttt{wrap}\ h_2\ \texttt{with}\ g_2$

$$\to M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{wrap})); S \cup \{\mathrm{enc}(l, k), g_2, \mathrm{enc}(k, l)\}$$

$1 : \texttt{dec}\ \mathrm{enc}(k, l)\ \texttt{with}\ g_1$

$$\to M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{wrap})); S \cup \{\mathrm{enc}(l, k), g_2, \mathrm{enc}(k, l), k\}$$

Administration

- Change the permissions associated with a handle
- Subject to a permission policy
- Conflicts + stickiness = non-monotonicity?

$$\begin{aligned}\phi_d(p,q) := p \subseteq q &\wedge (\{\mathsf{enc},\mathsf{unwrap}\} \not\subseteq q)\\ &\wedge (\{\mathsf{dec},\mathsf{wrap}\} \not\subseteq q)\\ &\wedge (\{\mathsf{wrap},\mathsf{unwrap}\} \not\subseteq q)\end{aligned}$$

$\mathtt{adm}\ h\ \mathtt{for}\ q:$

$$M \cup \{h \mapsto (k,p)\}; h \rightarrow M \cup \{h \mapsto (k,q)\}; \emptyset \qquad \text{if } \phi_d(p,q)$$

$\mathtt{gen}\ \mathtt{for}\ p:$

$$M; \emptyset \xrightarrow{h,k} M \cup \{h \mapsto (k,p)\}; h \qquad \text{if } \phi_d(p,p)$$

$$
\begin{aligned}
M_1 &= \{h_1 \mapsto (k, \mathsf{wrap}), g_1 \mapsto (l, \boxed{\emptyset})\} \\
M_2 &= \{h_2 \mapsto (k, \mathsf{unwrap})\} \\
S &= \{h_1, h_2, g_1, d\}
\end{aligned}
$$

Attack 2

$$M_1, M_2; S$$

$1 : \texttt{wrap}\ g_1\ \texttt{with}\ h_1$

$$\to M_1, M_2; S \cup \{\mathrm{enc}(l,k)\}$$

$2 : \texttt{unwrap}\ \ \mathrm{enc}(l,k)\ \texttt{with}\ h_2\ \texttt{for}\ \boxed{\emptyset}$

$$\xrightarrow{g_2} M_1, (M_2 \cup g_2 \mapsto (l, \boxed{\emptyset})); S \cup \{\mathrm{enc}(l,k), g_2\}$$

$2 : \boxed{\texttt{adm}\ g_2\ \texttt{for}\ \mathsf{wrap}}$

$$\to M_1, (M_2 \cup g_2 \mapsto (l, \boxed{\mathsf{wrap}})); S \cup \{\mathrm{enc}(l,k), g_2\}$$

$2 : \texttt{wrap}\ h_2\ \texttt{with}\ g_2$

$$\to M_1, (M_2 \cup g_2 \mapsto (l, \mathsf{wrap})); S \cup \{\mathrm{enc}(l,k), g_2, \mathrm{enc}(k,l)\}$$

$1 : \boxed{\texttt{adm}\ g_1\ \texttt{for}\ \mathsf{dec}}$

$$\to \{h_1 \mapsto (k, \mathsf{wrap}), g_1 \mapsto (l, \boxed{\mathsf{dec})}\}, M_2'; S \cup \{\mathrm{enc}(l,k), g_2, \mathrm{enc}(k,l)\}$$

$1 : \texttt{dec}\ \ \mathrm{enc}(k,l)\ \texttt{with}\ g_1$

$$\to \{h_1 \mapsto (k, \mathsf{wrap}), g_1 \mapsto (l, \mathsf{dec})\}, M_2'; S \cup \{\mathrm{enc}(l,k), g_2, \mathrm{enc}(k,l), \boxed{k}\}$$

Conclusions

- Non-monotonicity with multiple tokens:
  - Don’t rely on it!

- Consider multiple token scenarios:
  - Scenario-based verification
  - State invariants

Ongoing Work

- Towards a verified reference implementation
- Secrecy by typing
  - Abstract cryptography
  - Symmetric keys
  - Dolev-Yao attacker