ArmoredSoftware Semantics 0.0

ArmoredSoftware Crew
Information and Telecommunication Technology Center
The University of Kansas
palexand@ku.edu

November 24, 2014

Contents

List of Figures

Abstract

This document describes evolving ArmoredSoftware semantic definitions.

1 Introduction

2 SPI Calculus

Examples motivated by **?**.

2.1 Wide Mouth Frog

2.2 Needham Schroeder

$$\begin{aligned}
A &\rightarrow B : \{A^+, N_A\}_{B+} \text{ on } c \\
B &\rightarrow A : \{N_A, N_B\}_{A+} \text{ on } c \\
A &\rightarrow B : \{N_B\}_{B+} \text{ on } c
\end{aligned}$$

$$\begin{aligned}
A \;&\triangleq \overline{c}\langle\{(A, N_A)\}_{B+}\rangle. \\
&\quad c(M). \\
&\quad \mathsf{case}\ \{M\}_{A-}\ \mathsf{of}\ (N_A, nb)\ \mathsf{in} \\
&\quad \overline{c}\langle\{nb\}_{B+}\rangle. \\
&\quad A \\
B \;&\triangleq c(M). \\
&\quad \mathsf{case}\ \{M\}_{B-}\ \mathsf{of}\ (x, n)\ \mathsf{in} \\
&\quad \overline{c}\langle\{(n, N_B)\}_{x+}\rangle. \\
&\quad c(M). \\
&\quad \mathsf{case}\ \{M\}_{B-}\ \mathsf{of}\ N_B\ \mathsf{in}\ B \\
sys \;&\triangleq (\nu c)A \mid B
\end{aligned}$$

$$\text{React Inter}\ \frac{}{\overline{m}\langle N\rangle.P \mid m(x).Q \rightarrow P \mid [x \rightarrow N]Q}$$

$$\text{Red Replace}\ \frac{}{!P > P\ |!P}$$

$$\text{Red Match}\ \frac{}{[\![M\ \mathsf{is}\ M]\!]P > P}$$

$$\text{Red Let}\ \frac{}{\mathsf{let}\ (x, y) = (M, N)\ \mathsf{in}\ P > [x \rightarrow M][y \rightarrow N]P}$$

Note that we may want a more general let that matches more than pairs here. We'll see what the other inference rules give us.

$$\text{Red Zero}\ \frac{}{\mathsf{case}\ 0\ \mathsf{of}\ \ 0 : P\ suc(x) : Q > P}$$

$$\text{Red Suc}\ \frac{}{\mathsf{case}\ suc(M)\ \mathsf{of}\ \ 0 : P\ suc(x) : Q > [x \rightarrow M]Q}$$

I find the case rules over naturals quite crude.

$$\text{Red Sym Decrypt}\ \frac{}{\mathsf{case}\ \{M\}_k\ \mathsf{of}\ \{x\}_k\ \mathsf{in}\ P > [x \rightarrow M]P}$$

Additional proposed semantic rules for public/private key encryption and signature checking

$$\text{Red Asym Decrypt 1}\ \frac{}{\mathsf{case}\ \{M\}_{k^+}\ \mathsf{of}\ \{x\}_{k^-}\ \mathsf{in}\ P > [x \to M]P}$$

$$\text{Red Asym Decrypt 2}\ \frac{}{\mathsf{case}\ \{M\}_{k^-}\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ P > [x \to M]P}$$

The previous two rules capture the essence of asymmetric key pairs. Specifically, encrypt with one and decrypt with the other.

Assume $|M|$ is the hash and not the message itself and $\{|M|\}_{k^-}$ is the hash encrypted with private key, k^-. Thus, a signed message is the pair $(M, \{|M|\}_{k^-})$ consisting of the message and the signed hash. Given this, a successful signature check looks something like this:

$$\begin{array}{l}
\mathsf{let}\ (m,s) = (M, \{|M|\}_{k^-})\ \mathsf{in\ case}\ s\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [\![x\ \mathsf{is}\ |m|]\!]P \\
> [m \to M][s \to \{|M|\}_{k^-}]\mathsf{case}\ s\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [\![x\ \mathsf{is}\ |m|]\!]P \\
> \mathsf{case}\ \{|M|\}_{k^-}\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [m \to M][s \to \{|M|\}_{k^-}][\![x\ \mathsf{is}\ |m|]\!]P \\
> [x \to |M]\!][m \to M][s \to \{|M|\}_{k^-}][x\ \mathsf{is}\ |m]\!]P \\
> [\![|M|\ \mathsf{is}\ |M|]\!][x \to |M|][m \to M][s \to \{|M|\}_{k^-}]P \\
> [x \to |M|][m \to M][s \to \{|M|\}_{k^-}]P
\end{array}$$

This is precisely what we want. Specifically, P with m replaced by the message M and s replaced by the decrypted signature, $|M|$, produced by the signature check. It is unlikely that $|M|$ will be used in P, but it is available.

If the signature does not match, the process hangs. Assume the hash is incorrect. Specifically, $M \neq N$:

$$\begin{array}{l}
\mathsf{let}\ (m,s) = (M, \{|N|\}_{k^-})\ \mathsf{in\ case}\ s\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [\![x\ \mathsf{is}\ |m|]\!]P \\
> [m \to M][s \to \{|N|\}_{k^-}]\mathsf{case}\ s\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [\![x\ \mathsf{is}\ |m|]\!]P \\
> \mathsf{case}\ \{|N|\}_{k^-}\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [m \to M][s \to \{|N|\}_{k^-}][\![x\ \mathsf{is}\ |m|]\!]P \\
> [x \to |N|][m \to M][s \to \{|M|\}_{k^-}][\![x\ \mathsf{is}\ |m|]\!]P \\
> [\![|N|\ \mathsf{is}\ |M|]\!][x \to |N|][m \to M][s \to \{|M|\}_{k^-}]P
\end{array}$$

The process is stuck when $|N|\ \mathsf{is}\ |M|$ fails because $N \neq M$.

Now assume the wrong private key was used to sign the message hash. Specifically, $j \neq k$:

$$\begin{array}{l}
\mathsf{let}\ (m,s) = (M, \{|M|\}_{j^-})\ \mathsf{in\ case}\ s\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [\![x\ \mathsf{is}\ |m|]\!]P \\
> [m \to M][s \to \{|M|\}_{j^-}]\mathsf{case}\ s\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [\![x\ \mathsf{is}\ |m|]\!]P \\
> \mathsf{case}\ \{|M|\}_{j^-}\ \mathsf{of}\ \{x\}_{k^+}\ \mathsf{in}\ [m \to M][s \to \{|M|\}_{k^-}][\![x\ \mathsf{is}\ |m|]\!]P
\end{array}$$

The process is stuck when $\{|M|\}_{j^-}$ does not unify with $\{x\}_{k^+}$.

Do we really want a signature check that fails to get stuck? I think so. M is available, but the signature check is stuck. A signed message is best represented as a pair $(M, \{|M|\}_{k^-})$ allowing the message to be explicitly available.

$$\text{Struct Nil}\;\frac{}{P \mid \mathbf{0} \equiv P}$$

$$\text{Struct Comm}\;\frac{}{P \mid Q \equiv Q \mid P}$$

$$\text{Struct Assoc}\;\frac{}{P \mid (Q \mid R) \equiv (P \mid Q) \mid R}$$

$$\text{Struct Switch}\;\frac{}{(\nu m)(\nu n)P \equiv (\nu n)(\nu m)P}$$

$$\text{Struct Drop}\;\frac{}{(\nu n)\mathbf{0} \equiv \mathbf{0}}$$

$$\text{Struct Extrusion}\;\frac{n \notin fv(P)}{(\nu n)(P \mid Q) \equiv P \mid (\nu n)Q}$$

$$\text{Struct Red}\;\frac{P > Q}{P \equiv Q}$$

$$\text{Struct Refl}\;\frac{}{P \equiv P}$$

$$\text{Struct Symm}\;\frac{P \equiv Q}{Q \equiv P}$$

$$\text{Struct Trans}\;\frac{P \equiv Q \qquad Q \equiv R}{P \equiv R}$$

$$\text{Struct Par}\;\frac{P \equiv P'}{P \mid Q \equiv P' \mid Q}$$

$$\text{Struct Res}\;\frac{P \equiv P'}{(\nu n)P \equiv (\nu n)P'}$$

$$\text{React Struct}\;\frac{P \equiv P' \qquad P' \to Q' \qquad Q' \equiv Q}{P \to Q}$$

$$\text{React Par}\;\frac{P' \to P'}{P \mid Q \to P' \mid Q}$$

$$\text{React Res}\;\frac{P' \to P'}{(\nu n)P \to (\nu n)P}$$

$$A \quad \triangleq \quad \bar{c}\langle\{(A, N\_A)\}_{B+}\rangle.$$

2.3 Privacy CA Protocol

A Glossary

- **0** - null process

- $|M|$ - hash of M
- K^+ - public half of asymmetric key K
- K^- - private half of asymmetric key K
- $\{M\}_K$ - encrypt M with symmetric key K
- $\{M\}_{K^+}$ - encrypt M with the public key from K
- $\{M\}_{K^-}$ - decrypt M with the public key from K
- $\{|M|\}_{K^-}$ - sign M with the private key from K
- $\{|M|\}_{K^+}$ - check signature on M with the public key from K
- $(\nu x)P$ - new variable x defined in scope of P
- $\overline{c}\langle M\rangle$ - send M on channel c
- $c(M)$ - receive M on channel c
- $!P$ - infinite replication of P
- $P\ +\ Q$ - P or Q
- $P \mid Q$ - P in parallel with Q
- $\mathsf{case}\ \{M\}_k\ \mathsf{of}\ x\ \mathsf{in}\ P$ - attempt to decrypt $\{M\}_k$ and bind to x in P if successful. Stuck if unsuccessful
- $\mathsf{case}\ \{M\}_{k^-}\ \mathsf{of}\ x\ \mathsf{in}\ P$ - attempt to decrypt $\{M\}_{k^+}$ and bind to x in P if successful. Stuck if unsuccessful
- $\mathsf{case}\ \{|M|\}_{k^+}\ \mathsf{of}\ x\ \mathsf{in}\ P$ - attempt to check signature $\{|M|\}_{k^-}$ and bind to x in P if successful. Stuck if unsuccessful
- $\mathsf{case}\ x\ \mathsf{of}\ y\ 0 : P\ suc(x) : Q$ - case splitting over integers. x is bound in Q.
- $\mathsf{let}\ (x, y) = M\ \mathsf{in}\ y$ - match M to (x, y) binding x and y to pair elements in M
- $A\ \triangleq\ B$ - define an equivalence
- $A \rightarrow B : M\ \mathsf{on}\ c$ - A sends B message M on channel c

$$
\begin{aligned}
A\ \triangleq\ & (\nu c)\ \overline{c}\langle M\rangle.\mathbf{0}\ \mid \\
& c(M).A
\end{aligned}
$$