ArmoredSoftware Semantics 0.0

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

November 24, 2014

Contents

1 Introduction **1**

2 SPI Calculus **1**

2.1 Wide Mouth Frog 2

2.2 Needham Schroeder 2

2.3 Privacy CA Protocol 4

A Glossary **4**

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

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

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

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

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

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

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

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

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

Red Sig Check Broken $\dfrac{}{\mathsf{case}\ \{|M|\}_{k^-}\ \mathsf{of}\ \{|x|\}_{k^+}\ \mathsf{in}\ P > [x \to M]P}$

This rule has a more serious problem as it allows us to reproduce a message from it's signature. Specifically, if we have $\{|M|\}_{k^-}$ and signature match is successful, then x is bound to M. That can't happen. Possibly the rule should look like this:

Red Sig Check $\dfrac{}{\mathsf{case}\ \{|M|\}_{k^-}\ \mathsf{of}\ \{|x|\}_{k^+}\ \mathsf{in}\ P > [x \to |M]]P}$

where $|M|$ is the hash and not the message itself. A signature check should look something like this:

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

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 $M \neq N$:

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

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.

I think this is pretty much what we want.

Struct Nil $\dfrac{}{P \mid \mathbf{0} \equiv P}$

Struct Comm $\dfrac{}{P \mid Q \equiv Q \mid P}$

Struct Assoc $\dfrac{}{P \mid (Q \mid R) \equiv (P \mid Q) \mid R}$

Struct Switch $\dfrac{}{(\nu m)(\nu n)P \equiv (\nu n)(\nu m)P}$

Struct Drop $\dfrac{}{(\nu n)\mathbf{0} \equiv \mathbf{0}}$

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

Struct Red $\dfrac{P > Q}{P \equiv Q}$

Struct Refl $\dfrac{}{P \equiv P}$

Struct Symm $\dfrac{P \equiv Q}{Q \equiv P}$

Struct Trans $\dfrac{P \equiv Q \qquad Q \equiv R}{P \equiv R}$

Struct Par $\dfrac{P \equiv P'}{P \mid Q \equiv P' \mid Q}$

Struct Res $\dfrac{P \equiv P'}{(\nu n)P \equiv (\nu n)P'}$

React Struct $\dfrac{P \equiv P' \qquad P' \rightarrow Q' \qquad Q' \equiv Q}{P \rightarrow Q}$

React Par $\dfrac{P' \rightarrow P'}{P \mid Q \rightarrow P' \mid Q}$

React Res $\dfrac{P' \rightarrow P'}{(\nu n)P \rightarrow (\nu n)P}$

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

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
- $\textsf{case}\ \{M\}_k\ \textsf{of}\ x\ \textsf{in}\ P$ - attempt to decrypt $\{M\}_k$ and bind to x in P if successful. Stuck if unsuccessful
- $\textsf{case}\ \{M\}_{k^-}\ \textsf{of}\ x\ \textsf{in}\ P$ - attempt to decrypt $\{M\}_{k^+}$ and bind to x in P if successful. Stuck if unsuccessful
- $\textsf{case}\ \{|M|\}_{k^+}\ \textsf{of}\ x\ \textsf{in}\ P$ - attempt to check signature $\{|M|\}_{k^-}$ and bind to x in P if successful. Stuck if unsuccessful
- $\textsf{case}\ x\ \textsf{of}\ y\ 0 : P\ suc(x) : Q$ - case splitting over integers. x is bound in Q.
- $\textsf{let}\ (x, y) = M\ \textsf{in}\ y$ - match M to (x, y) binding x and y to pair elements in M
- $A \triangleq B$ - define an equivalence
- $A \rightarrow B : M\ \textsf{on}\ c$ - A sends B message M on channel c

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