

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

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

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

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

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

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

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

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.

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

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

I find the **case** rules over naturals quite crude.

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

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

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

$$\text{Red Sig Check Broken} \frac{}{\text{case } \{|M|\}_{k^-} \text{ of } \{|x|\}_{k^+} \text{ in } P > [x \rightarrow M]P}$$

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:

$$\text{Red Sig Check} \frac{}{\text{case } \{|M|\}_{k^-} \text{ of } \{|x|\}_{k^+} \text{ in } P > [x \rightarrow |M|]P}$$

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

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

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 $M \neq N$:

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

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.

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

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

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

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

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

$$\begin{array}{c}
\text{Struct Extrusion} \frac{n \notin fv(P)}{(\nu n)(P \mid Q) \equiv P \mid (\nu n)Q} \\
\text{Struct Red} \frac{P > Q}{P \equiv Q} \\
\text{Struct Refl} \frac{}{P \equiv P} \\
\text{Struct Symm} \frac{P \equiv Q}{Q \equiv P} \\
\text{Struct Trans} \frac{P \equiv Q \quad Q \equiv R}{P \equiv R} \\
\text{Struct Par} \frac{P \equiv P'}{P \mid Q \equiv P' \mid Q} \\
\text{Struct Res} \frac{P \equiv P'}{(\nu n)P \equiv (\nu n)P'} \\
\text{React Struct} \frac{P \equiv P' \quad P' \rightarrow Q' \quad Q' \equiv Q}{P \rightarrow Q} \\
\text{React Par} \frac{P' \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \\
\text{React Res} \frac{P' \rightarrow P'}{(\nu n)P \rightarrow (\nu n)P}
\end{array}$$

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

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

$$A \triangleq (\nu c) \ \bar{c}\langle M \rangle. \mathbf{0} \mid c(M).A$$