

Private Authentication

Martín Abadi and Cédric Fournet

Deepak Garg

Languages and Logics for Security

Goals and Motivation

- Construct an authentication protocol with privacy guarantees
 - Honest principals A and B should be able to establish a secret
 - Adversary should not learn the identities of the honest parties involved
 - Most authentication protocols (think simple challenge response, Needham-Schroeder) send principal names in the clear
 - Primarily intended for use in broadcast settings (for e.g., wireless access points)
- Develop bisimulation techniques to prove protocol correct
- Formalized and analyzed in the applied pi-calculus (pi-calculus + equational theory on terms)

Assumptions

- On Participating Principals
 - Pre-established private-public keys
 - No trusted third parties
- Cryptographic
 - Encrypted messages don't reveal public key unless private key is known (which-key concealing)
 - Use of incorrect decryption key should be evident
- Network and Adversary
 - Network packets don't reveal sending/receiving principals
 - Dolev-Yao adversary (can read and inject messages; cannot break cryptography)

Protocol

- Notation

- A – initiator
- B – responder
- S_B – set of principals responder will accept connections from

- Protocol

- $A \rightarrow B: \{N_A, K_A\}_{K_B}$
- $B \rightarrow A: \begin{cases} \{N_A, N_B, K_B\}_{K_A} & \text{if } A \in S_B \\ \{N\}_K & \text{if } A \notin S_B \end{cases}$

- $h(N_A, N_B)$ is the shared secret (h is a one-way hash function)

Properties Established

$$\begin{array}{l}
 A \rightarrow B : \{N_A, K_A\}_{K_B} \\
 B \rightarrow A : \begin{cases} \{N_A, N_B, K_B\}_{K_A} & \text{if } A \in S_B \\ \{N\}_K & \text{if } A \notin S_B \end{cases}
 \end{array}$$

- [Feasibility] Protocol can always be completed
- [Privacy] To an adversary, protocol is same as running (for random values N_1, N_2)
 - $? \rightarrow ? : N_1$
 - $? \rightarrow ? : N_2$
- [Key freshness] End configuration cannot distinguish $h(N_A, N_B)$ from a random value
- [Responder Authentication] If in some reachable state P , initiator A believes that the protocol was completed, then
 - 1 All steps of the protocol occurred
 - 2 P could be reached by first running the protocol and then running the remaining steps
- A generic method for establishing other privacy properties

Applied pi-calculus

Term	$U, V, W ::= a, n$ x, y $f(U_1, \dots, U_n)$	name variable functions
Process	$P, Q, R ::= 0$ $P \mid Q$ $!P$ $\nu n.P$ $\text{if } U = V \text{ then } P \text{ else } Q$ $u(x).P$ $\bar{u}\langle V \rangle.P$	nil parallel composition replication name restriction conditional input output

Extended Processes

Extended Process $A, B, C ::= P$

- $A \mid B$
- $\nu n.A$
- $\nu x.A$
- $\{x = V\}$ active substitution

- Substitutions assumed to be acyclic
- $A[M/x] \equiv \nu x.(A \mid \{x = M\})$ if $x \notin fv(M)$
- For each bound variable x , assume exactly one active substitution.

Equational Theory

$$\begin{array}{l}
 A \rightarrow B : \{N_A, K_A\}_{K_B} \\
 B \rightarrow A : \begin{cases} \{N_A, N_B, K_B\}_{K_A} & \text{if } A \in S_B \\ \{N\}_K & \text{if } A \notin S_B \end{cases}
 \end{array}$$

- $\Sigma \vdash U = V$
- Encryption: $\{V\}_K$
- Decryption: $\text{decrypt}(V, K)$
- $\Sigma \vdash \text{decrypt}(\{V\}_K, K^{-1}) = V$
- Special functions: $\text{hello}(N_A, K_A)$, hello.0 , hello.1 , $\text{ack}(N_A, N_B, K_B)$, ack.0 , ack.1 , ack.2
- Often use pattern matching (derived construct)

Operational Semantics and Bisimulation

- Labeled Transitions: $A \xrightarrow{\alpha} B$

$$\alpha = a(V) \qquad a(x).P \xrightarrow{a(V)} \nu x.(P \mid \{x = V\})$$

$$\alpha = \nu \tilde{u}.\bar{a}\langle V \rangle \qquad \bar{a}\langle V \rangle.P \xrightarrow{\bar{a}\langle V \rangle} P$$
- Full definition in the paper
- [Labeled Bisimulation] $A \mathcal{R} B$ if:
 - 1 A and B have the same top-level active substitutions
 - 2 if $A \rightarrow A'$, then $B \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B'
 - 3 if $A \xrightarrow{\alpha} A'$, $fv(\alpha) \subseteq dom(A)$ and $bn(\alpha) \cap fn(B) = \phi$, then $B \rightarrow^* \xrightarrow{\alpha} \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B' .
- \approx_l is the largest labeled bisimulation

Modeling the Protocol

Honest principal, Q_A : two parallel components U_A, P_A

U_A : User process (arbitrary)

$P_A = I_A | R_A$: Initiator protocol | Responder protocol

I_A : Initiator protocol (fixed)

R_A : Responder protocol (fixed)

Three channels: $connect_A, init_A, accept_A$ private to P_A

U_A as initiator:

1. $U_A \rightarrow I_A: \overline{init_A}\langle B \rangle$
2. I_A runs protocol with R_B , obtaining secret K
3. $I_A \rightarrow U_A: \overline{connect_A}\langle B, K \rangle$

R_B as responder:

1. $R_B \rightarrow U_B: \overline{accept_B}\langle A, K \rangle$

Initiator and Responder

$$P_A = I_A \mid R_A$$

$$I_A = !\text{init}_A(B). \nu N_A. (\overline{c_1} \langle x_1 \sigma_1 \rangle \mid I'_A)$$

$$I'_A = c_2(x_2).$$

if $x_2 = \{\text{ack}(N_A, \nu N_B, B)\}_A$ using K_A^{-1} then $\overline{\text{connect}_A} \langle B, K \sigma_K \rangle$

$$R_B = !c_1(x_1 \setminus \phi).$$

if x_1 fresh and $x_1 = \{\text{hello}(\nu N_A, \nu A)\}_B$ using K_B^{-1} and $A \in S_B$

then $\nu N_B. (\overline{c_2} \langle x_2 \sigma_2 \rangle \mid \overline{\text{accept}_B} \langle A, K \sigma_K \rangle)$ else $\nu N_B. \overline{c_2} \langle x_2 \sigma_2^\circ \rangle$

$$\sigma_1 = \{x_1 = \{\text{hello}(N_A, A)\}_B\}$$

$$\sigma_2 = \{x_2 = \{\text{ack}(N_A, N_B, B)\}_A\}$$

$$\sigma_2^\circ = \{x_2 = N_B\}$$

$$\sigma_K = \{K = h(N_A, N_B)\}$$

Configurations

- Any number of honest principals in initiator/responder roles
- Adversary modeled by evaluation context

$$\begin{aligned} \mathcal{C} &= \text{set of honest principals} \\ PK_A[-] &= \nu K_A^{-1} . (\{A = \text{pk}(K_A^{-1})\} \mid [-]) \\ \mathcal{P} &= \prod_{A \in \mathcal{C}} PK_A[P_A] \end{aligned}$$

Feasibility

- Let $A, B \in \mathcal{C}$
- (Success) If $\mathcal{P} \xrightarrow{\eta} P'$ and $A \in S_B$, then
 $P' \xrightarrow{\omega} \approx_I P' \mid \nu N_1.\{x_1 = N_1\} \mid \nu N_2.\{x_2 = N_2\} \mid \nu N.\{K = N\}$
- (Failure) If $\mathcal{P} \xrightarrow{\eta} P'$ and $A \notin S_B$, then
 $P' \xrightarrow{\omega^-} \approx_I P' \mid \nu N_1.\{x_1 = N_1\} \mid \nu N_2.\{x_2 = N_2\}$

$$\begin{array}{l}
 \xrightarrow{\omega} = \xrightarrow{\text{init}_A(B)} \xrightarrow{\nu x_1.\overline{c_1}\langle x_1 \rangle} \xrightarrow{c_1(x_1)} \xrightarrow{*} \xrightarrow{\nu x_2.\overline{c_2}\langle x_2 \rangle} \xrightarrow{c_2(x_2)} \xrightarrow{\nu K.\overline{\text{accept}_B}\langle A, K \rangle} \xrightarrow{\overline{\text{connect}_A}\langle B, K \rangle} \\
 \\
 \xrightarrow{\omega^-} = \xrightarrow{\text{init}_A(B)} \xrightarrow{\nu x_1.\overline{c_1}\langle x_1 \rangle} \xrightarrow{c_1(x_1)} \xrightarrow{*} \xrightarrow{\nu x_2.\overline{c_2}\langle x_2 \rangle} \xrightarrow{c_2(x_2)} \xrightarrow{}
 \end{array}$$

Privacy and Key Freshness

- Let $A, B \in \mathcal{C}$
- If $\mathcal{P} \xrightarrow{\omega} \mathcal{P}'$, then $A \in \mathcal{S}_B$ and
 $\mathcal{P}' \approx_I \mathcal{P} \mid \nu N_1. \{x_1 = N_1\} \mid \nu N_2. \{x_2 = N_2\} \mid \nu N. \{K = N\}$

$$\begin{array}{l}
 \xrightarrow{\omega} = \frac{\text{init}_A(B) \xrightarrow{\nu x_1. \overline{c_1} \langle x_1 \rangle} c_1(x_1) \xrightarrow{*} \nu x_2. \overline{c_2} \langle x_2 \rangle c_2(x_2) \xrightarrow{\quad}}{\nu K. \overline{\text{accept}_B} \langle A, K \rangle \quad \overline{\text{connect}_A} \langle B, K \rangle} \\
 \\
 \xrightarrow{\omega^-} = \frac{\text{init}_A(B) \xrightarrow{\nu x_1. \overline{c_1} \langle x_1 \rangle} c_1(x_1) \xrightarrow{*} \nu x_2. \overline{c_2} \langle x_2 \rangle c_2(x_2) \xrightarrow{\quad}}{\quad}
 \end{array}$$

Responder Authentication

- Let $A, B \in \mathcal{C}$
- If $\mathcal{P} \xrightarrow{\eta} \mathcal{P}'$ and $\overline{\text{connect}}_A \langle B, K \rangle \in \eta$ for some K , then
 $\mathcal{P} \xrightarrow{\omega} \xrightarrow{\eta'} \mathcal{P}'$ for some permutation $\omega\eta'$ of η

$$\begin{array}{l}
 \xrightarrow{\omega} = \frac{\text{init}_A(B) \xrightarrow{\nu x_1. \overline{c_1} \langle x_1 \rangle} c_1(x_1) \xrightarrow{*} \nu x_2. \overline{c_2} \langle x_2 \rangle} \xrightarrow{\nu K. \overline{\text{accept}}_B \langle A, K \rangle} \xrightarrow{\overline{\text{connect}}_A \langle B, K \rangle} \\
 \xrightarrow{\omega^-} = \frac{\text{init}_A(B) \xrightarrow{\nu x_1. \overline{c_1} \langle x_1 \rangle} c_1(x_1) \xrightarrow{*} \nu x_2. \overline{c_2} \langle x_2 \rangle} \xrightarrow{c_2(x_2)} \xrightarrow{}
 \end{array}$$

More Privacy Properties

- Paper describes a general way to prove privacy properties involving user process U_A .

- An example:

- Let

$$\begin{aligned} \mathcal{V} &= \bigcup_{D \in C} \{init_D, accept_D, connect_D\} \\ U_{A_1} &= \overline{init_{A_1}} \langle B_1 \rangle \mid connect_{A_1}(B_1, K).V_1 \\ U_{A_2} &= \overline{init_{A_2}} \langle B_2 \rangle \mid connect_{A_2}(B_2, K).V_2 \end{aligned}$$

- If $\nu K.V_1 \approx_I \nu K.V_2$, and V_1, V_2 do not use channels in \mathcal{V} , then $\nu \mathcal{V}.(U_{A_1} \mid \mathcal{P}) \approx_I \nu \mathcal{V}.(U_{A_2} \mid \mathcal{P})$