

Secrecy-Oriented First-Order Logical Analysis of Cryptographic Protocols

Gergei Bana (ENS Cachan)
with Koji Hasebe, Mitsuhiro Okada

Computational Soundness of Formal Protocol Analysis

- Active adversaries in two groups:

- Two-world view**

- Symbolic and computational executions are formalized separately as well as security properties
 - Soundness: Try to prove that no successful symbolic (Dolev-Yao) attacker implies no successful computational attacker.
 - Such are
 - Reactive Simulatability of M. Backes, B. Pfitzmann, M. Waidner
 - D. Micciancio, B. Warinschi, Cortier (mapping lemma)
 - V. Cortier, H. Comon-Lundh (soundness of observational equivalence)

- Logical view**

- Only computational execution, symbolic formulas have direct computational meaning
 - Logical theory axiomatizes the relevant properties cryptographic primitives.
 - Security properties are directly proven from the axioms and derivation rules
 - Computational Protocol Compositional Logic of Stanford (John Mitchell's group)
 -

Parsing

- ⊗ Computational soundness and parsing
 - ⊗ Formal expressions that are different but have the same computational interpretation cause problems
 - ⊗ Especially in two-world view, but also in PCL
 - ⊗ E.g. $\{N\}_K^R = \{N'\}_{K'}^{R'}$ problems with key forging
 - ⊗ Or, type-flaw attacks depend on things like $\langle n, m \rangle = N$
 - ⊗

Aims

- **First-order system where (a fully probabilistic) soundness does not have to rely on parsing, fake keys are allowed.**
- **Takes care of key cycles**
- **Relatively simple**
- **We can still prove protocols (NSL[10pages], symmetric NS[20pages], Otway-Rees all for unbounded sessions)**
- **We do this putting a non-trace property, secrecy in the center, even trace properties**

Sorts, Function Symbols

Sorts

$$\left. \begin{array}{l} \text{hseed} \\ \text{hname} \subseteq \text{name} \\ \text{hnonce} \\ \text{hkey} \\ \text{sessionid} \end{array} \right\} \subseteq \text{bitstring} \subseteq \text{bittree}$$

timesection event

Infinitely many variables of each sort

Message terms

$$T ::= t \mid \overline{T} \mid \langle T, T \rangle \mid \{T\}_Q^s \mid \{T\}_{QQ'}^s \mid \{T\}_k^s$$

t: variable of sort **bittree**, **s**: **bitstring**, **Q, Q'**: names

T: in general sort **bittree**, but **T overline**: **bitstring**

Message terms represent parsed messages, but **T overline** is the corresponding bitstring

They are in fact infinite sequences of random variables

Computational Model

The usual probabilistic polynomial time execution, the adversary controlling the network

For each fixed security parameter, the execution is a stochastic process (Markov) with an underlying probability space

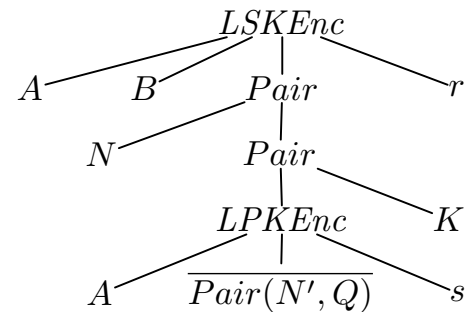
Interpretation of elements of sort bitstring: sequences (in the security par) of random variables

Interpretation of elements of sort bittree: Ordered trees with sequences of random var on the leaves and encryption or paring on the internal nodes.

Interpretation of elements of sort timesection: Infinite sequence of stopping times

Interpretation of elements of sort event: Non-negligible sequence of subsets of the probability spaces

Tree structure of
 $\{N, \{\overline{\{N', Q\}}_A^s, K\}_{AB}^r\}$



Atomic Formulas

$$\varphi_0 ::= Q \text{ acts}_\tau^i t \mid \text{Sec}_\tau(\mathbf{A}, t, \nu) \mid \text{KeySec}_\tau(\mathbf{A}, t, K) \mid t = t' \mid t \sqsubseteq t' \mid \tau < \tau'$$

acts

is either generates or receives or sends

Q does the corresponding action on section τ in session i doing as much parsing as indicated by \dagger

Sec, KeySec

A is a list of honest names $\langle A, B, C \dots \rangle$ for Sec and KeySec to be not false.

ν is either honest nonce or honest key.

Sec means that agents other than those listed in A together with the adversary, based on their combined view until τ cannot distinguish ν from another nonce (or key) generated independently from the protocol, even if we give them the bit string corresponding to \dagger

KeySec means that agents other than those listed in A together with the adversary, based on their combined view until τ cannot break the security game (CCA-2) against key K even if we give them the bit string corresponding to \dagger

Formulas

$$\varphi_0 ::= Q \text{ acts}_\tau^i t \mid \text{Sec}_\tau(\mathbf{A}, t, \nu) \mid \text{KeySec}_\tau(\mathbf{A}, t, K) \mid t = t' \mid t \sqsubseteq t' \mid \tau < \tau'$$

=

on bitstrings: equality of sequences of random variables up to negligible sets.

on bittrees: the leaves are equal up to neglig, except the random seed.

⊆

$$\overline{\{t\}_{Q_1 Q_2}^s} = \overline{\{t\}_{Q_1 Q_2}^{s'}} \rightarrow \{t\}_{Q_1 Q_2}^s = \{t\}_{Q_1 Q_2}^{s'}$$

subterm

e.g.:

$$N \not\sqsubseteq \overline{\{\{N\}_B^r, N'\}_{QQ'}^s} \quad \overline{\{N\}_B^r} \sqsubseteq \overline{\{\{N\}_B^r, N'\}_{QQ'}^s}$$

<

τ is earlier than τ' on all traces

Formulas:

$$\varphi ::= \varphi_0 \mid \neg \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \forall v \varphi \mid \exists v \varphi$$

Roles

E.g. NSL

1. $A \rightarrow B: \{N_1, A\}_B$ 2. $B \rightarrow A: \{N_1, N_2, B\}_A$ 3. $A \rightarrow B: \{N_2\}_B$

With our syntax:

$Init_{\text{NSL}}^A[A, i, Q, N_1, n_2, r_1, s_2, r_3] \equiv A \text{ generates}^i N_1; A \text{ sends}^i \{N_1, A\}_Q^{r_1};$
 $A \text{ receives}^i \{N_1, n_2, Q\}_A^{s_2}; A \text{ sends}^i \{n_2\}_Q^{r_3}$

$Resp_{\text{NSL}}^B[B, i', Q', n_1, N_2, s_1, r_2, s_3] \equiv B \text{ receives}^{i'} \{n_1, Q'\}_B^{s_1}; B \text{ generates}^{i'} N_2;$
 $B \text{ sends}^{i'} \{n_1, N_2, B\}_{Q'}^{r_2}; B \text{ receives}^{i'} \{N_2\}_B^{s_3}$

$Q_1 \text{ acts}_{1}^{i_1} t_1; \dots; Q_k \text{ acts}_{k}^{i_k} t_k \equiv$

$\exists \tau_1 \dots \tau_k (\mathbf{0} < \tau_1 < \dots < \tau_k \wedge Q_1 \text{ acts}_{1, \tau_1}^{i_1} t_1 \wedge \dots \wedge Q_k \text{ acts}_{k, \tau_k}^{i_k} t_k)$

Agreement and Authentication

E.g. NSL from the responder's view:

This is what we want to prove

$$\text{Resp}_{NSL}^B[i', A, n_1, N_2, s_1, r_2, s_3] \wedge \text{FOLL}(\text{Init}_{NSL}^A) \wedge \text{FOLL}(\text{Resp}_{NSL}^B) \\ \vdash \exists i r_1 s_2 r_3 \text{Init}_{NSL}^A[i, B, n_1, N_2, r_1, s_2, r_3]$$

Where

$$\text{FOLL}(\text{Init}_{NSL}^A) \equiv \forall i \exists Q N_1 n_2 r_1 s_2 r_3 \text{Foll}(\text{Init}_{NSL}^A[i, Q, N_1, n_2, r_1, s_2, r_3])$$

Foll means that an initial section of the trace in question was executed (maybe to the end) with the given values.

Proof through secrecy 1

First we show that nonces (or keys) are not corrupted throughout the protocol.

Let's try to prove:

$$\forall Ait\nu\tau \left(A \sqsubseteq \mathbf{A} \wedge C[\nu] \wedge A \text{ sends}_\tau^i t \wedge [Key]Sec_\tau(\mathbf{A}, \nu) \longrightarrow [Key]Sec_\tau(\mathbf{A}, t, \nu) \right)$$

C expresses that ν was generated by the agents in \mathbf{A} and intended for communication among them

The formula expresses that ν remains a secret of the agents in \mathbf{A}

Does not work.

Proof through secrecy 2

We need:

$$\begin{aligned} [Key]SecSend(\mathbf{A}, C, C') &\equiv \forall Ait\nu u\tau \left((A \sqsubseteq \mathbf{A} \wedge C[\nu] \wedge C'[u] \wedge \nu \not\sqsubseteq u \wedge A \text{ sends}_\tau^i t) \right. \\ &\quad \left. \wedge \forall u' (C'[u'] \wedge \nu \not\sqsubseteq u' \rightarrow [Key]Sec_\tau(\mathbf{A}, u', \nu)) \right) \\ &\rightarrow [Key]Sec_\tau(\mathbf{A}, \langle t, u \rangle, \nu) \end{aligned}$$

In NSL: $\mathbf{A} = \langle \mathbf{A}, \mathbf{B} \rangle$

$$C[N] \equiv \exists irn (A \text{ generates}^i N; A \text{ sends}^i \{N, A\}_B^r \vee B \text{ generates}^i N; B \text{ sends}^i \{n, N, B\}_A^r)$$

$$\begin{aligned} C'[u] &\equiv \forall t (t \sqsubseteq u \rightarrow \exists m (t = m) \vee \exists t_1 t_2 (t = \langle t_1, t_2 \rangle)) \\ &\quad \wedge \forall m (m \sqsubseteq u \rightarrow \exists i (A \text{ generates}^i m \vee B \text{ generates}^i m)) \end{aligned}$$

C expresses that **N** was generated by **A** or **B**

C' expresses that **u** is a list of nonces generated by **A** or **B**

Proof steps

First show

$$FOLL(Init_{NSL}^A) \wedge FOLL(Resp_{NSL}^B) \vdash SecSend(\langle A, B \rangle, C, C').$$

For each A' sends $_{\tau}^i t$ send actions of A and B, we have to show that $Sec_{\tau}(\langle A, B \rangle, u', N)$ for all u' implies $Sec_{\tau}(\langle A, B \rangle, \langle u, t \rangle, N)$ for all u .

Send actions of Init: $t = \{N_1, A\}_Q^{r_1} \vee t = \{n_2\}_Q^{r_3}$

Send actions of Resp: $t = \{n_1, N_2, B\}_{Q'}$

Once SecSend is proven, agreement and auth.

Axioms 1

Axioms about the ordering of τ

Term axioms resulting e.g. $N \not\equiv \{\overline{\{N\}_B^r}, N'\}_{QQ'}^s$

Axioms about secrecy $\neg[Key]Sec_\tau(\mathbf{A}, \nu, \nu)$

$$[Key]Sec_\tau(\mathbf{A}, \langle t, t' \rangle \longrightarrow \langle t', t \rangle, \nu)$$

$$[Key]Sec_\tau(\mathbf{A}, \langle t, t' \rangle \longrightarrow t, \nu)$$

$$[Key]Sec_\tau(\mathbf{A}, t \xrightarrow{A \sqsubseteq \mathbf{A}} \langle t, \{t'\}_A^r \rangle, \nu)$$

$$(Q \text{ sends}_\tau^i t \vee Q \text{ receives}_\tau^i t) \wedge \tau < \tau' \\ \wedge [Key]Sec_{\tau'}(\mathbf{A}, t', \nu) \rightarrow [Key]Sec_\tau(\mathbf{A}, \langle t, t' \rangle, \nu)$$

$$Sec_\tau(\mathbf{A}, t \xrightarrow{KeySec(\mathbf{A}, \langle t, t' \rangle, K) \wedge K \neq \nu} \langle t, \{t'\}_K^r \rangle, \nu)$$

Axioms 2

Encryption implying authentication e.g. for public key

$$\begin{aligned}
 & \left(\{t'\}_A^s \sqsubseteq t_2 \wedge A \text{ receives}_{\tau_2}^{i_2} t_2 \right. \\
 & \quad \left. \wedge [Key]Sec_{\tau_2}(\mathbf{A}, \langle t, \{t'\}_A^s \rangle, \nu) \wedge \neg [Key]Sec_{\tau_2}(\mathbf{A}, \langle t, t' \rangle, \nu) \right) \\
 & \rightarrow \exists A' t_1 t'' i_1 r \tau_1 \left(A' \sqsubseteq \mathbf{A} \wedge \{t''\}_A^r \sqsubseteq t_1 \right. \\
 & \quad \left. \wedge \overline{\{t'\}_A^s} = \overline{\{t''\}_A^r} \wedge \left(A' \text{ sends}_{\tau_1}^{i_1} t_1; A \text{ receives}_{\tau_2}^{i_2} t_2 \right) \right)
 \end{aligned}$$

Not sound.

Because of this, all predicates have to be redefined on non-negligible subsets

$$\begin{aligned}
 & \left(\{t'\}_A^s \sqsubseteq t_2 \wedge A \text{ receives}_{\tau_2}^{i_2} t_2 \right. \\
 & \quad \left. \wedge [Key]Sec_{\tau_2}(\mathbf{A}, \langle t, \{t'\}_A^s \rangle, \nu) \wedge \neg [Key]Sec_{\tau_2}(\mathbf{A}, \langle t, t' \rangle, \nu) \right) \Big|_{\Delta} \\
 & \rightarrow \exists A' t_1 t'' i_1 r \tau_1 \Delta' \left(A' \sqsubseteq \mathbf{A} \wedge \Delta' \subseteq \Delta \wedge \{t''\}_A^r \sqsubseteq t_1 \right. \\
 & \quad \left. \wedge \overline{\{t'\}_A^s} = \overline{\{t''\}_A^r} \wedge \left(A' \text{ sends}_{\tau_1}^{i_1} t_1; A \text{ receives}_{\tau_2}^{i_2} t_2 \right) \Big|_{\Delta'} \right)
 \end{aligned}$$

Good news

For properties that are preserved under restriction to subsets and unions, you don't have to consider sets in the proof

Soundness: Soundness holds if the encryption is CCA-2 (for sym key, we have unforgeable and non unforgeable versions) and if length of pairing and encryption depend only on the length of the inputs

Useful: We can actually prove protocols

So far: NSL, symmetric NS, Otway-Rees

No extra assumptions

We only include things in the axioms that we are absolutely sure of (with CCA-2).

E.g. the NSL proof needs the condition

$$\overline{\langle n, A \rangle} \neq N$$

If we want to allow such **type-flaw attack**, we don't include this in the axioms.

If we want to allow **assigned name attacks**, then we have to remove some axioms.

Further Work

Should be easy: allowing corrupted long-term keys

Dynamic corruption - should not be difficult

Composability conditions