

Tree Automata and Tree Transducers for Analyzing Recursive Cryptographic Protocols

Thomas Wilke—joint work with R. Küsters

June 9, 2006

Cryptographic Protocols ...

- ... run in a hostile environment,
- ... run concurrently (parallel sessions),
- ... interfere with each other.

Example: Authenticated Ping

Goal

A successful protocol run between Alice (as originator) and Bob (as responder) guarantees that Bob is still alive.

Assumptions

Every pair of principals $\{A, B\}$ shares a secret key $k_{\{A,B\}}$ for symmetric encryption and decryption.

Technique

Challenge-and-response protocol.

Example: Authenticated Ping (cont'd)

Protocol

Alice (as originator):

- 1a Alice chooses a random number n (nonce) and encrypts it with $k_{\text{Alice,Bob}}$; result: m (challenge).
- 1b Alice sends m to Bob over the network.
- 2a Alice waits for a reply n' from Bob.
- 2b Alice checks whether $n = n'$ and if so, she considers the run successful (i.e., she believes Bob is alive).

Bob (as responder):

- 1a Bob receives a message m' from Alice and decrypts it to n' .
- 1b Bob sends n' (response) to Alice.

Authenticated Ping: The Problem

- ▶ Alice sends a challenge m to Bob.
- ▶ The intruder removes Alice's message m from the network.
- ▶ The intruder starts a new run of the protocol: She pretends to be Bob and asks Alice for authentication, using the challenge m from the first run.
- ▶ Alice decrypts m and sends n to the intruder (as part of the second run).
- ▶ The intruder replies to Alice's request in the first run by sending n to her.
- ▶ Alice receives n and considers the first run to have been completed successful.

The Formal Model: Dolev & Yao

- ▶ messages are terms
- ▶ cryptographic primitives work perfectly
- ▶ intruder controls the entire network

Messages

K_s, K_p, \hat{K}_p : keys for (a)symmetric encryption/decryption

\mathcal{A} : atomic messages

$$\Sigma = K_s \cup K_p \cup \hat{K}_p \cup \mathcal{A} \\ \cup \{\text{enc}_k^s(\cdot) : k \in K_s\} \cup \{\text{enc}_k^a(\cdot) : k \in K_p\} \cup \{\langle \cdot, \cdot \rangle\},$$

message = ground term

Example

$\langle B, \text{enc}_{k_{\text{Alice, Bob}}}^s(N) \rangle, \text{enc}_{k_{\text{Alice}}}^a(\langle \text{Alice}, \langle \text{Bob}, N \rangle \rangle)$

Inference Capabilities of Intruder

W set of messages

$\text{der}(W)$, set of messages derivable from W :

- ▶ $W \subseteq \text{der}(W)$
- ▶ $\langle m, m' \rangle \in \text{der}(W)$ if $m, m' \in \text{der}(W)$
- ▶ $m, m' \in \text{der}(W)$ if $\langle m, m' \rangle \in \text{der}(W)$
- ▶ $\text{enc}_k^s(m) \in \text{der}(W)$ if $k, m \in \text{der}(W)$ and $k \in K_s$
- ▶ $\text{enc}_k^a(m) \in \text{der}(W)$ if $k, m \in \text{der}(W)$ and $k \in K_p$
- ▶ $m \in \text{der}(W)$ if $k \in \text{der}(W)$ and $\text{enc}_k^s(m) \in \text{der}(W)$
- ▶ $m \in \text{der}(W)$ if $\hat{k} \in \text{der}(W)$ and $\text{enc}_k^a(m) \in \text{der}(W)$

High-Level View of an Attack

W_I given initial knowledge of the intruder

1. The intruder sends a message $x_0 \in \text{der}(W_I)$ to some principal.
2. The principal receives the message, processes it, sends reply y_0 to the intruder.— Principal performs receive-send action.
3. The intruder sends a message $x_1 \in \text{der}(W_I \cup \{y_0\})$ to some principal (same or other).
4. The principal receives the message, processes it, sends reply y_1 to the intruder.— Principal performs receive-send action.
5. The intruder sends a message $x_2 \in \text{der}(W_I \cup \{y_0, y_1\})$ to some principal (same or other).

...

Successful Attack

Something bad happens, for instance, $\text{secret} \in W_I \cup \{y_0, \dots, y_{r-1}\}$ for some atom secret the intruder should not get hold of.

How are the Principals (receive-send actions) Specified?

(By now) Classical Approach

Model

- ▶ a finite number of principals
- ▶ principal = finite sequence of receive-send actions, processed one after the other
- ▶ receive-send action: a root rewrite rule such as

$$\text{enc}_{k_{\{Alice, Bob\}}}^s(x) \rightarrow x$$

Theorem [Amadio, Lugiez, Vanackere & Rusinowitch, Turuani]

Whether Eve gets hold of secret (Attack) is NP-complete.

Extensions and Restrictions of Basic Result

Extensions

- ▶ complex keys
- ▶ exclusive or
- ▶ exponentiation
- ▶ associativity for pairing
- ▶ ...

Restrictions

- ▶ parallel sessions have to be spelled out, hence only a fixed number of parallel sessions can be dealt with (undecidability otherwise)
- ▶ receive-send actions are very, very simple (no recursion)

Recursive Authentication Protocol by Bull and Otway

Goal

Distribute (session) keys among the members of a group using a key distribution server S .

Assumption

Every member P_i of the group shares a secret key k_i with the server S .

Server's receive-send action

S receives:

$\text{hash}_{k_{P_r}} \langle P_r, S, N_r, \text{hash}_{k_{r-1}} \langle P_{r-1}, P_r, N_{r-1} \dots \text{hash}_{k_0} \langle P_0, P_1, N_0, \text{init} \rangle \rangle \rangle$

S sends:

$\langle \text{enc}_{k_r}^s \langle \langle k_{r,S}, S, N_r \rangle \rangle, \text{enc}_{k_r}^s \langle \langle k_{r-1,r}, P_{r-1}, N_r \rangle \rangle, \text{enc}_{k_{r-1}}^s \langle \langle k_{r-1,r-2}, P_r, N_{r-1} \rangle \rangle, \dots, \text{enc}_{k_0}^s \langle \langle k_{0,1}, P_1, N_0 \rangle \rangle \rangle$

We need ...

- ... iteration (arbitrarily long requests to server)
- ... a mechanism for generating new alphabet symbols (new keys)
- ... memory for storing these symbols

We wish ...

- ... to be able to model a receive-send action of the intruder.

Receive-Send Actions by Tree Transducers (TTAC's)

non-deterministic top-down tree transducers with

- ▶ linear left-hand sides (decidability)
- ▶ anonymous constants
- ▶ a register for anonymous constants
- ▶ ϵ -steps
- ▶ regular look ahead

Formal Description of TTAC's

- ▶ C infinite set of anonymous constants, $C \cap \Sigma = \emptyset$
- ▶ \mathcal{A} bottom-up tree automaton with state set S
- ▶ Q finite set of states of tree transducer, q_I initial state
- ▶ finite number of transitions:

$$q(t) \Rightarrow_s t'([q_0, z_0](t_0), \dots, [q_{r-1}, z_{r-1}](t_{r-1}))$$

where

- ▶ $q, q_0, \dots, q_{r-1} \in Q$
- ▶ t is a linear term
- ▶ $s \in S$
- ▶ $z_i = v_R$ or $z_i = v_N$
- ▶ the t_i 's are subterms of t
- ▶ t' is an arbitrary term which may also have occurrences of v_R and v_N

Example

$$\text{start}(\text{hash}_{k_i} \langle P_i, S, x_0, x_1 \rangle) \Rightarrow [\text{read}, v_N](\text{hash}_{k_i} \langle P_i, S, x_0, x_1 \rangle)$$

$$\begin{aligned} \text{read}(\text{hash}_{k_i} (\langle P_i, P_j, x_0, \text{hash}_{k_{i'}} (\langle P_{i'}, P_i, x_1, x_2 \rangle) \rangle)) \Rightarrow \\ \langle \text{enc}_{k_i}^s (\langle v_R, P_j, x_0 \rangle), \text{enc}_{k_i}^s (\langle v_N, P_{i'}, x_0 \rangle), \\ [\text{read}, v_N](\text{hash}_{k_{i'}} (\langle P_{i'}, P_i, x_1, x_2 \rangle)) \rangle \end{aligned}$$

$$\text{read}(\text{hash}_{k_i} \langle P_i, P_j, x_0, \text{init} \rangle) \Rightarrow \text{enc}_{k_i}^s \langle v_R, P_j, x_0 \rangle$$

Results

Theorem

Attack is decidable for a finite number of principals each of which is modelled as a finite sequence of TTAC's.

Lemma 1

There exists a TTAC which realizes the relation

$$\{(m, m') \mid m \text{ message, } m' \in \text{der}(\{m\})\} .$$

Lemma 2

The inverse image of a “regular language of trees with anonymous constants” is a “regular language of anonymous constants”.

Lemma 3

The word problem is decidable for “regular languages of trees with anonymous constants”.

Tree Automata with Anonymous Constants (TAAC's)

Non-deterministic bottom-up tree automaton

$$\mathcal{A} = (Q, q_I, \delta, q^d, q^s, \Delta, F)$$

where

- ▶ $q^d \in Q$ is a **default state**
- ▶ $q^s \in Q$ is a **selecting state**

Rule for anonymous constants

- ▶ all occurrences of at most one anonymous constant get assigned q^s and
- ▶ all other occurrences of anonymous constants get assigned q^d .

$$L(\mathcal{A}) = \{t \mid \text{there exists a run } t \text{ which yields } q \in F\}$$

regular language of trees with anonymous constants: $L(\mathcal{A})$ for some TAAC \mathcal{A}

Examples

$$L = \{f(c, c) \mid c \in C\}$$

for each i ,

$$L_i = \{t \mid t \text{ has at least } i \text{ occurrences of anonymous constants}\}$$

for each i ,

$$L'_i = \{t \mid t \text{ has at least } i \text{ distinct anonymous constants}\}$$

for each i ,

$$L''_m = \{f(t, t') \mid \exists c(c \in C \wedge \#_c(t) \neq \#_c(t') \bmod m)\}$$

Open Problems

- ▶ What is the complexity of the iterated preimage problem for TTAC's and TAAC's?
- ▶ Is there a good data structure for representing TAAC's (which are obtained by computing inverse images with respect to the "intruder automaton")?