

Proof theory for security protocols

A Baskar Sergiu Bursuc R Ramanujam S P Suresh

LSV, ENS Cachan, France
bursuc@lsv.ens-cachan.fr

Institute of Mathematical Sciences, Chennai
jam@imsc.res.in

Chennai Mathematical Institute
{spsuresh,abaskar}@cmi.ac.in

Formal Methods Update Meeting 2008
July 19, 2008

- 1 *Introduction*
- 2 *The passive intruder*
- 3 *The active intruder*

- Structural proof theory
 - Structure of proofs
 - Normal forms for proofs
 - Normalization
 - Strategies for proof search
- What has all this got to do with security protocols?

- An example scenario ...

Akshay \rightarrow Suresh: Hey, you there?

Suresh \rightarrow Akshay: Very much!

- An example protocol ...

$A \rightarrow B: \{n\}_B$

$B \rightarrow A: \{n\}_A$

- Another protocol

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$
$$A?B:\{n\}_A$$

- Another protocol

 $A!B:\{A, \{n\}_B\}_B$ $A?B:\{n\}_A$ $B?A:\{A, \{n\}_B\}_B$ $B!A:\{n\}_A$

- Another protocol

$A!B:\{A, \{n\}_B\}_B$

$A?B:\{n\}_A$

$B?A:\{A, \{n\}_B\}_B$

$B!A:\{n\}_A$

- ... and an attack!

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$
$$A?B:\{n\}_A$$
$$B?A:\{A, \{n\}_B\}_B$$
$$B!A:\{n\}_A$$

- ... and an attack!

$$A!B:\{A, \{p\}_B\}_B$$

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$
$$A?B:\{n\}_A$$
$$B?A:\{A, \{n\}_B\}_B$$
$$B!A:\{n\}_A$$

- ... and an attack!

$$A!B:\{A, \{p\}_B\}_B$$
$$B?I:\{I, \{A, \{p\}_B\}_B\}_B$$

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$

$$A?B:\{n\}_A$$

$$B?A:\{A, \{n\}_B\}_B$$

$$B!A:\{n\}_A$$

- ... and an attack!

$$A!B:\{A, \{p\}_B\}_B$$

$$B?I:\{I, \{A, \{p\}_B\}_B\}_B$$

$$B!I:\{A, \{p\}_B\}_I$$

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$

$$A?B:\{n\}_A$$

$$B?A:\{A, \{n\}_B\}_B$$

$$B!A:\{n\}_A$$

- ... and an attack!

$$A!B:\{A, \{p\}_B\}_B$$

$$B?I:\{I, \{A, \{p\}_B\}_B\}_B$$

$$B!I:\{A, \{p\}_B\}_I$$

$$B?I:\{I, \{p\}_B\}_B$$

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$

$$A?B:\{n\}_A$$

$$B?A:\{A, \{n\}_B\}_B$$

$$B!A:\{n\}_A$$

- ... and an attack!

$$A!B:\{A, \{p\}_B\}_B$$

$$B?I:\{I, \{A, \{p\}_B\}_B\}_B$$

$$B!I:\{A, \{p\}_B\}_I$$

$$B?I:\{I, \{p\}_B\}_B$$

$$B!I:\{p\}_I$$

- Another protocol

$$A!B:\{A, \{n\}_B\}_B$$

$$A?B:\{n\}_A$$

$$B?A:\{A, \{n\}_B\}_B$$

$$B!A:\{n\}_A$$

- ... and an attack!

$$A!B:\{A, \{p\}_B\}_B$$

$$B?I:\{I, \{A, \{p\}_B\}_B\}_B$$

$$B!I:\{A, \{p\}_B\}_I$$

$$B?I:\{I, \{p\}_B\}_B$$

$$B!I:\{p\}_I$$

$$A?B:\{p\}_A$$

- The basis for **symbolic analysis** of protocols
- Messages are symbolic terms
- A **malicious active intruder**
 - copies messages sent over the network
 - can generate new messages from old according to some rules
 - can initiate new sessions by sending/re-routing messages in its own/others' name

- The basic verification problem
- Given a protocol, and a specific secret m , decide whether there exists an execution of the protocol in which the intruder can learn m .
- Search for attacks!
 - a sequence of message exchanges
 - action of honest agents dictated by the protocol
 - every message sent by the intruder should be constructible from messages learnt earlier
 - check to see if $T \vdash t$, for appropriate T and t !

- 1 *Introduction*
- 2 *The passive intruder*
- 3 *The active intruder*

$\frac{}{T \cup \{t\} \vdash t} \text{Ax}$	
$\frac{T \vdash (t_1, t_2)}{T \vdash t_i} \text{split}_i (i = 1, 2)$	$\frac{T \vdash t_1 \quad T \vdash t_2}{T \vdash (t_1, t_2)} \text{pair}$
$\frac{T \vdash \{t\}_k \quad T \vdash \text{inv}(k)}{T \vdash t} \text{decrypt}$ <p style="text-align: center;"><i>analz</i>-rules</p>	$\frac{T \vdash t \quad T \vdash k}{T \vdash \{t\}_k} \text{encrypt}$ <p style="text-align: center;"><i>synth</i>-rules</p>

- The deducibility problem: given T and t , check if there is proof of $T \vdash t$
- This problem is decidable.
 - A notion of normal proofs.
 - If $T \vdash t$ is provable, there is a normal proof of $T \vdash t$.
 - Every term r occurring in a normal proof of $T \vdash t$ is a subterm of $T \cup \{t\}$.
 - Derive bounds on the size of normal proofs from this.

- An example:

$$\frac{\frac{\frac{}{\{t\} \vdash t} Ax}{\{t\} \vdash t} \text{ pair}}{\{t\} \vdash (t, t)} \text{ split}}{\{t\} \vdash t}$$

- An example:

$$\frac{\frac{\frac{}{\{t\} \vdash t} Ax \quad \frac{}{\{t\} \vdash t} Ax}{\{t\} \vdash (t, t)} pair}{\{t\} \vdash t} split$$

- Another one:

$$\frac{\frac{\frac{}{\{t, k\} \vdash t} Ax \quad \frac{}{\{t, k\} \vdash k} Ax}{\{t, k\} \vdash \{t\}_k} encrypt \quad \frac{}{\{t, k\} \vdash inv(k)} Ax}{\{t, k\} \vdash t} decrypt$$

Normalization rules

$$\begin{array}{c}
 \pi_1 \qquad \pi_2 \\
 \vdots \qquad \vdots \\
 \frac{T \vdash t \quad T \vdash t}{T \vdash (t, t)} \textit{pair} \\
 \frac{T \vdash (t, t)}{T \vdash t} \textit{split}
 \end{array}
 \rightsquigarrow
 \begin{array}{c}
 \pi_1 \\
 \vdots \\
 T \vdash t
 \end{array}$$

$$\begin{array}{c}
 \pi_1 \qquad \pi_2 \\
 \vdots \qquad \vdots \\
 \frac{T \vdash t \quad T \vdash k}{T \vdash \{t\}_k} \textit{encrypt} \\
 \frac{T \vdash \{t\}_k \quad T \vdash \textit{inv}(k)}{T \vdash t} \textit{decrypt}
 \end{array}
 \rightsquigarrow
 \begin{array}{c}
 \pi_1 \\
 \vdots \\
 T \vdash t
 \end{array}$$

Lemma

If π is a normal proof of $T \vdash t$ and r occurs in π :

- $r \in st(T \cup \{t\})$
- if π ends in an *analz*-rule, then $r \in st(T)$.

Lemma

If π is a normal proof of $T \vdash t$ and r occurs in π :

- $r \in st(T \cup \{t\})$
- if π ends in an *analz*-rule, then $r \in st(T)$.

$$\frac{\begin{array}{c} \pi_1 \\ \vdots \\ T \vdash t \end{array} \quad \begin{array}{c} \pi_2 \\ \vdots \\ T \vdash k \end{array}}{T \vdash \{t\}_k} \text{encrypt}$$

- if r occurs in π_1 ,
 $r \in st(T \cup \{t\})$
- if r occurs in π_2 ,
 $r \in st(T \cup \{k\})$
- therefore, if r occurs in π ,
 $r \in st(T \cup \{\{t\}_k\})$

Lemma

If π is a normal proof of $T \vdash t$ and r occurs in π :

- $r \in st(T \cup \{t\})$
- if π ends in an *analz*-rule, then $r \in st(T)$.

$$\frac{\begin{array}{c} \pi_1 \\ \vdots \\ T \vdash \{t\}_k \end{array} \quad \begin{array}{c} \pi_2 \\ \vdots \\ T \vdash inv(k) \end{array}}{T \vdash t} \text{decrypt}$$

- if r occurs in π_1 or π_2 ,
 $r \in st(T \cup \{\{t\}_k\})$
- since π is normal, π_1 does not end with the *encrypt* rule
- so it ends with an *analz* rule, and $\{t\}_k \in st(T)$
- so any r occurring in π is in $st(T)$.

A polynomial-time algorithm

- The height of a normal proof of $T \vdash t$ is bounded by $n = |st(T \cup \{t\})|$.
- Let $T_0 = T$
- Compute $T_i = \text{one-step-derivable}(T_{i-1}) \cap st(T \cup \{t\})$, for $i \leq n$
- Check if $t \in T_n$!

- What about other cryptographic primitives?
- Diffie-Hellman encryption, exclusive or, homomorphic encryption, blind signatures, ...
- A large body of results: Rusinowitch & Turuani 2003, Millen & Shmatikov 2001, Comon & Shmatikov 2003, Chevalier, Küsters, Rusinowitch & Turuani 2005, Delaune & Jacquemard 2006, Bursuc, Comon & Delaune 2007, Lafourcade, Lugiez & Treinen 2007

Cancellations: the xor case

- One new *synth* rule:

$$\frac{T \vdash t_1 \quad \dots \quad T \vdash t_n}{T \vdash (t_1 \oplus \dots \oplus t_n) \downarrow} \text{ xor}$$

- Normalization rules: no more than one occurrence of any term as a premise of an *xor* rule
- Simplify

$$\frac{\frac{\frac{\pi'_1}{T \vdash t'_1} \quad \dots \quad \frac{\pi'_m}{T \vdash t'_m}}{T \vdash t_1} \text{ xor} \quad \dots \quad \frac{\pi_n}{T \vdash t_n} \text{ xor}}{T \vdash t} \text{ xor}$$

- The cases other than *xor* go through smoothly
- *xor* brings cancellations to the party!

$$\frac{\begin{array}{c} \pi_1 \\ \vdots \\ T \vdash t_1 \oplus t_2 \end{array} \quad \begin{array}{c} \pi_2 \\ \vdots \\ T \vdash t_2 \oplus t_3 \end{array}}{T \vdash t_1 \oplus t_3} \text{ xor}$$

- t_2 is not a subterm of the conclusion. Is it a subterm of the premises?

- The cases other than *xor* go through smoothly
- *xor* brings cancellations to the party!

$$\frac{\begin{array}{c} \pi_1 \\ \vdots \\ T \vdash t_1 \oplus t_2 \end{array} \quad \begin{array}{c} \pi_2 \\ \vdots \\ T \vdash t_2 \oplus t_3 \end{array}}{T \vdash t_1 \oplus t_3} \text{ xor}$$

- t_2 is not a subterm of the conclusion. Is it a subterm of the premises? **Yes!**

Nontrivial interaction: homomorphic encryption

- Add the rewriting rule: $\{t_1 \oplus \dots \oplus t_n\}_k \longrightarrow \{t_1\}_k \oplus \dots \oplus \{t_n\}_k$
- The subterm property no longer holds!

Nontrivial interaction: homomorphic encryption

- Add the rewriting rule: $\{t_1 \oplus \dots \oplus t_n\}_k \longrightarrow \{t_1\}_k \oplus \dots \oplus \{t_n\}_k$
- The subterm property no longer holds!

$$\frac{\frac{\overline{T \vdash t \oplus t'} \quad Ax} \quad \overline{T \vdash k} \quad Ax}{T \vdash \{t\}_k \oplus \{t'\}_k} \text{encrypt} \quad \frac{\overline{T \vdash \{t\}_k} \quad Ax}{\{t'\}_k} \text{xor}$$

Nontrivial interaction: homomorphic encryption

- Add the rewriting rule: $\{t_1 \oplus \dots \oplus t_n\}_k \longrightarrow \{t_1\}_k \oplus \dots \oplus \{t_n\}_k$
- The subterm property no longer holds!

$$\frac{\frac{\overline{T \vdash t \oplus t'} \quad Ax}}{T \vdash \{t\}_k \oplus \{t'\}_k} \quad \frac{\overline{T \vdash k} \quad Ax}{\text{encrypt}}}{\frac{\overline{T \vdash \{t\}_k} \quad Ax}{T \vdash \{t\}_k} \quad \text{xor}}{\{t'\}_k}$$

- Considerably more complex normalization rules
- Every atom of every r occurring in a normal proof of $T \vdash t$ belongs to $st(T \cup \{t\})$!
- Use this to derive an EXPTIME decision procedure.

Nontrivial interaction: homomorphic encryption

- Add the rewriting rule: $\{t_1 \oplus \dots \oplus t_n\}_k \longrightarrow \{t_1\}_k \oplus \dots \oplus \{t_n\}_k$
- The subterm property no longer holds!

$$\frac{\frac{\overline{T \vdash t \oplus t'} \quad Ax}{T \vdash \{t\}_k \oplus \{t'\}_k} \quad \frac{\overline{T \vdash k} \quad Ax}{\text{encrypt}}}{\frac{\overline{T \vdash \{t\}_k} \quad Ax}{xor} \quad \{t'\}_k}$$

- Considerably more complex normalization rules
- Every atom of every r occurring in a normal proof of $T \vdash t$ belongs to $st(T \cup \{t\})$!
- Use this to derive an EXPTIME decision procedure.
- The hardest case: encryption homomorphic over an abelian group operator.
- Non-elementary decision procedure!

- 1 *Introduction*
- 2 *The passive intruder*
- 3 *The active intruder*

- Consider the protocol:

$$A \rightarrow B: \{n\}_{\text{public}(B)}$$

$$B \rightarrow A: \{n\}_{\text{private}(B)}$$

- I can build $\{n\}_{\text{private}(B)}$ from n and $\text{private}(B)$. Problem is, it is highly unlikely to have $\text{private}(B)$!

- Consider the protocol:

$$\begin{aligned} A &\rightarrow B: \{n\}_{\text{public}(B)} \\ B &\rightarrow A: \{n\}_{\text{private}(B)} \end{aligned}$$

- I can build $\{n\}_{\text{private}(B)}$ from n and $\text{private}(B)$. Problem is, it is highly unlikely to have $\text{private}(B)$!
- But it can also build $\{n\}_{\text{private}(B)}$ from $\{n\}_{\text{public}(B)}$!
- One can add the following rule in the intruder system:

$$\frac{T \vdash \{n\}_{\text{public}(B)}}{T \vdash \{n\}_{\text{private}(B)}}$$

- Active attacks are just proofs in this system!

$$\frac{}{T \cup \{t:(\sigma, u)\} \vdash_a t:(\sigma, u)} \text{Ax}'$$

$$\frac{}{T \cup \{t:(\sigma, u)\} \vdash_a t:(\sigma, u)} \text{Ax}'$$

$$\frac{T \vdash_a (t_1, t_2):(\sigma, (u_1, u_2))}{T \vdash_a t_i:(\sigma, u_i)} \text{split}'_i (i = 1, 2)$$

$$\frac{}{T \cup \{t:(\sigma, u)\} \vdash_a t:(\sigma, u)} \text{Ax}'$$

$$\frac{T \vdash_a (t_1, t_2):(\sigma, (u_1, u_2))}{T \vdash_a t_i:(\sigma, u_i)} \text{split}'_i (i = 1, 2)$$

$$\frac{T \vdash_a \{t_1\}t_2:(\sigma, \{u_1\}u_2) \quad T \vdash_s \text{inv}(t_2):(\sigma, \text{inv}(u_2))}{T \vdash_a t:(\sigma, u_1)} \text{decrypt}'$$

$$\frac{T \vdash_a t : (\sigma_1, u_1) \quad t \text{ is not a pair}}{T \vdash_s t : (\sigma_2, u_2)} \textit{unify}$$

$$\frac{T \vdash_a t : (\sigma_1, u_1) \quad t \text{ is not a pair}}{T \vdash_s t : (\sigma_2, u_2)} \textit{unify}$$

$$\frac{T \vdash_s t : (\sigma_1, u_1) \quad u_1 \text{ is not a nonce} \quad n \text{ is a nonce}}{T \vdash_s t : (\sigma_2, n)} \textit{simplify}$$

$$\frac{T \vdash_a t : (\sigma_1, u_1) \quad t \text{ is not a pair}}{T \vdash_s t : (\sigma_2, u_2)} \textit{unify}$$

$$\frac{T \vdash_s t : (\sigma_1, u_1) \quad u_1 \text{ is not a nonce} \quad n \text{ is a nonce}}{T \vdash_s t : (\sigma_2, n)} \textit{simplify}$$

$$\frac{T \vdash_s t_1 : (\sigma, u_1) \quad T \vdash_s t_2 : (\sigma, u_2)}{T \vdash_s (t_1, t_2) : (\sigma, (u_1, u_2))} \textit{pair'}$$

$$\frac{T \vdash_a t : (\sigma_1, u_1) \quad t \text{ is not a pair}}{T \vdash_s t : (\sigma_2, u_2)} \textit{unify}$$

$$\frac{T \vdash_s t : (\sigma_1, u_1) \quad u_1 \text{ is not a nonce} \quad n \text{ is a nonce}}{T \vdash_s t : (\sigma_2, n)} \textit{simplify}$$

$$\frac{T \vdash_s t_1 : (\sigma, u_1) \quad T \vdash_s t_2 : (\sigma, u_2)}{T \vdash_s (t_1, t_2) : (\sigma, (u_1, u_2))} \textit{pair'}$$

$$\frac{T \vdash_s t_1 : (\sigma, u_1) \quad T \vdash_s t_2 : (\sigma, u_2)}{T \vdash_s \{t_1\}_{t_2} : (\sigma, \{u_1\}_{u_2})} \textit{encrypt'}$$

- Normal proofs in this system guide us in performing simplifications to runs of a protocol to get more well-behaved runs.
- Such simplifications are the basis of the NP decision procedure for insecurity presented in [Rusinowitch, Turuani 2003](#).
- Also used essentially in [Ramanujam, Suresh 2006](#) in decidability proofs for tagged protocols.
- These arguments can be cast purely as a proof-search problem.

- Basic problems related to security protocol verification can be easily cast in terms of proof search.
- Normalization techniques are essential to the solution of many of these problems.
- Can we apply other tools from proof theory? For example, can we use advanced techniques for proof search to improve tools for cryptographic protocol verification?
- More fundamental questions:
 - what is the logical content of the key cryptographic operations?
 - can we come up with new cryptographic operators based on new logical structures?

Thank you!