

A Computationally Sound Mechanized Prover for Security Protocols

P. Cagnée, D. Kolokosso, F. Méjean, L. Pillard, J. Tharaud

National School of Applied Mathematics and Computer Science, ENSIMAG

27 November 2009

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

CryptoVerif

A Computationally Sound Mechanized Prover
for Security Protocols

Bruno Blanchet (CNRS, ENS, Paris)

2 approaches for proving secrecy properties of security protocols :

- Symbolic : $\{ \langle a, x \rangle \}_k$, a deduction system (e.g. Dolev-Yao model), proofs based on constraint solving, ...
- Computational : $10101001010\dots$, a PPTT machine, proofs based on cryptographic assumption (\rightarrow [CryptoVerif](#))

CryptoVerif is a sequence of games transformations :
first game = real protocol represented in process calculus
final game = no variables, only arrays of booleans

Two consecutive games cannot be distinguished.

- Process calculus = pi-calculus + cryptographic primitives
- Pi-calculus : probabilistic semantic over bistrings
 - input process, output process
 - arrays of booleans, replication
 - parallel composition, channel restriction
- Cryptographic primitives : functions over bistrings (blackboxes)

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

Observational equivalence

Definition, more important result

Adversary represented by Context $C[.] \rightarrow a$

context C : process with an hole, having access to V , set of Variables

Processes Q, Q' , verifying invariant-rules

if $|Pr[C[Q] \rightarrow 1] - Pr[C[Q'] \rightarrow 1]|$ is negligible then $Q \approx^V Q'$

The adversary cannot distinguish which process have been used.

Observational equivalence

Definition, more important result

Adversary represented by Context $C[.] \rightarrow$ a

context C : process with an hole, having access to V , set of Variables

Processes Q, Q' , verifying invariant-rules

if $|Pr[C[Q] \rightarrow 1] - Pr[C[Q'] \rightarrow 1]|$ is negligible then $Q \approx^V Q'$

The adversary cannot distinguish which process have been used.

Which

purpose ?

if $Q \approx^V Q'$ then

$$\text{GAME1}[Q] \rightarrow_{\approx} \text{GAME2}[Q']$$

using syntactic and primitives transformations

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

Goal : transform the process that represents the initial protocol into a **process** on which security property can be proved directly.

It consists in :

- syntactic transformations (*RemoveAssign*(x), *SArename*(x), *Simplify*())
- applying the definition of security of primitives : axioms used by the prover to transform a game into another equivalent game

What means the security primitives ?

- Cryptographic fonctions like enc, mac, keygen ...
- Designed like black-boxes here
- e.g : **MAC (Message Authentication Code)** linked with check relation : $\text{check}(m,k,\text{mac}(m,k)) = \text{true}$
Guaranties Authenticity and integrity of a message

Predefined transformation for security primitives:

check

Because, mac is UF-CMA (difficult to forge), then we can replace

`check(m,k,t)`

with:

find $j < N$ such that defined $(x[j]) \wedge (m = x[j]) \wedge \text{check}'(m,k,t)$ then true ,
else false

It means that the adversary can compute check only if he has already computed mac(m,k);

enc

Because enc is IND-CPA we can replace :

$enc(x, keygen(r))$

with :

$enc'(Z(x), keygen'(r))$

where $Z(x)$ returns a bitstring of the same length than x

Intuitively, it means that adversary cannot distinguish the cyphering of 2 same-size messages

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

Secrecy Criterias:

- one-session secrecy
- secrecy

Lemma

If $Q \approx^x Q'$ and Q preserves the one-session secrecy of x then Q' preserves the one-session secrecy of x . The same result holds for secrecy.

We can then apply the following mechanism, to prove that one protocol preserves the one-session secrecy of x :

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

How do we organize transformations in order to prove protocols:

```
// if we can apply crypto transformations
while(Is_transformable() == 1)
{
    apply_crypto_transform()
    //the game is modified
    Simplify()
    if(Is_transformable() == 0) then
        //we apply, if necessary, syntactic transformations
        RemoveAssign()
        SARename()
    if(IsSecret()) then return "SUCCESS"
}
return "FAILED"
```

- 1 CryptoVerif and Semantic
- 2 Equivalences
- 3 Game Transformations
- 4 Proof for Security
 - Security Primitives
 - Criteria for proving Secrecy Properties
 - Proof Strategy
- 5 Results and Conclusion

Tests of the prover on a number of protocols :

- configuration in which participants run sessions with the adversary
- prove secrecy of keys for sessions between participants.

But some cases of failure :

- Needham-Schroeder public-key : limitation for N_A
- Needham-Schroeder shared-key : does not prove the secrecy of the exchanged key (only in the corrected version)
- Denning-Sacco public-key, Yahalom : the same but for the one-session secrecy

Results are conclusive : using CryptoVerif prover to prove protocols in the computational model, without relying on the Dolev-Yao model, is a great progress. Briefly :

- limitations in some cryptographic primitives (Diffie-Hellman key agreements)
- best suited for proving security protocols such as encryption and signatures.

Questions ?