Reducing memory consumption of ProVerif with hash consing techniques

Margot Catinaud

PROSECCO team, INRIA Paris
under supervision of Vincent Cheval

07 September 2022

Verification of protocols

(a) The Needham-Shroeder protocol

$$
\begin{aligned}
A \longrightarrow B &: \mathtt{enc}((A, n_A), \mathtt{pk}(B)) \\
B \longrightarrow A &: \mathtt{enc}((n_A, n_B), \mathtt{pk}(A)) \\
A \longrightarrow B &: \mathtt{enc}(n_B, \mathtt{pk}(B))
\end{aligned}
$$

Main difficulties encountered

Main difficulties encountered

1. Cryptographic weaknesses

Main difficulties encountered

1. Cryptographic weaknesses
2. Protocol design flaws

Main difficulties encountered

1. Cryptographic weaknesses
2. Protocol design flaws
3. Implementation bugs

Cryptography

Protocol

Implementation

Main difficulties encountered

1. Cryptographic weaknesses
2. **Protocol design flaws**
3. Implementation bugs

Cryptography

Protocol

Implementation

General overview of ProVerif

- Symbolic approach: Dolev-Yao model

General overview of ProVerif

- Symbolic approach: Dolev-Yao model

- Cryptographic primitives are black-boxes

$$\frac{\mathtt{enc}(x, \mathtt{pk}(A)) \qquad \mathtt{sk}(A)}{x}\ \mathtt{dec}$$

General overview of ProVerif

- Symbolic approach: Dolev-Yao model

- Cryptographic primitives are black-boxes

$$\frac{\mathtt{enc}(x, \mathtt{pk}(A)) \qquad \mathtt{sk}(A)}{x}\ \mathtt{dec}$$

- Three main types of security properties handled: *secrecy*, *authenticity* and *equivalence*

Horn clauses

Example of Horn clause

$$(\texttt{enc}/\texttt{dec}\ \text{scheme}) \quad \texttt{att}(\texttt{enc}(x, k)) \wedge \texttt{att}(k) \Rightarrow \texttt{att}(x)$$

Horn clauses

Example of Horn clause

$$(\texttt{enc}/\texttt{dec}\text{ scheme}) \quad \texttt{att}(\texttt{enc}(x, k)) \wedge \texttt{att}(k) \Rightarrow \texttt{att}(x)$$

Figure 4: Syntax of Horn clauses

$M, N ::=$	terms
x	variable
$f(M_1, \ldots, M_n)$	function application

Horn clauses

Example of Horn clause

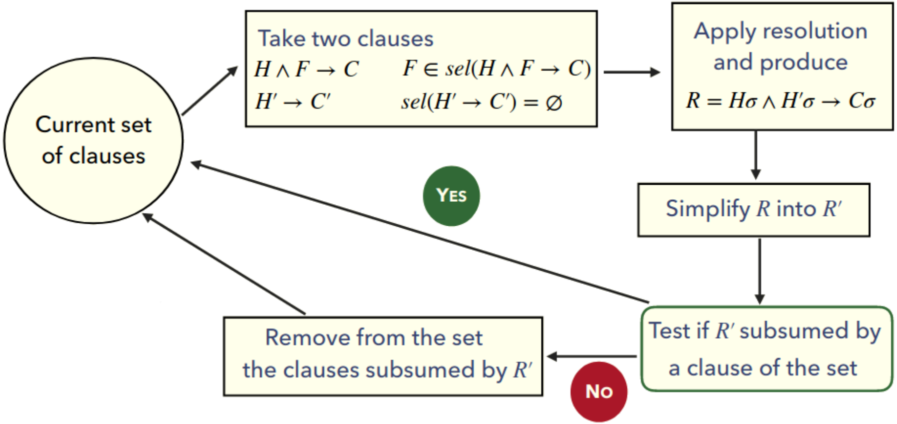
$$(\texttt{enc}/\texttt{dec}\text{ scheme}) \quad \texttt{att}(\texttt{enc}(x,k)) \wedge \texttt{att}(k) \Rightarrow \texttt{att}(x)$$

Figure 4: Syntax of Horn clauses

$M, N ::=$	terms
x	variable
$f(M_1, \ldots, M_n)$	function application
$F ::= p(M_1, \ldots, M_n)$	fact
$R ::= F_1 \wedge \ldots \wedge F_n \Rightarrow F$	Horn clause

Saturation process

Figure 5: Summary of the saturation procedure



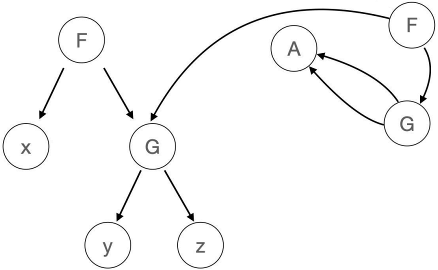
New representation of terms

Let $t_1 = f(x, g(y, z))$ and $t_2 = f(g(y, z), g(a, a))$ be two terms.

New representation of terms

Let $t_1 = f(x, g(y, z))$ and $t_2 = f(g(y, z), g(a, a))$ be two terms.

Figure 6: Example of term-graph representing t_1 and t_2



New representation of terms - OCaml

```
type hcterm = {

    hc_desc: hcterm_desc;
```

New representation of terms - OCaml

```
type hcterm = {
    hc_tag: int;
    hc_desc: hcterm_desc;
```

New representation of terms - OCaml

```
type hcterm = {
    hc_tag: int;
    hc_desc: hcterm_desc;
    mutable hc_link: hcterm_link
}
```

New representation of terms - OCaml

```
type hcterm = {
    hc_tag: int;
    hc_desc: hcterm_desc;
    mutable hc_link: hcterm_link
}

type hcterm_link =
    | HCNoLink
    | HCVisited of bool
    | HCTerm of hcterm
    | (* ... *)
```

Databases of Horn clauses

- Horn clauses are invariant by renaming variables

Databases of Horn clauses

- Horn clauses are invariant by renaming variables

- Several databases of Horn clauses:

  Solved clauses: $\texttt{sel}(H \Rightarrow C) = \varnothing$
  Unsolved clauses: $F \in \texttt{sel}(H \Rightarrow C)$

Syntactic unification

- No **occur-check**

Syntactic unification

- No **occur-check**

- First step: unification whithout care of cycles

Syntactic unification

- No **occur-check**
- First step: unification whithout care of cycles
- Second step: cycle detection

Syntactic unification

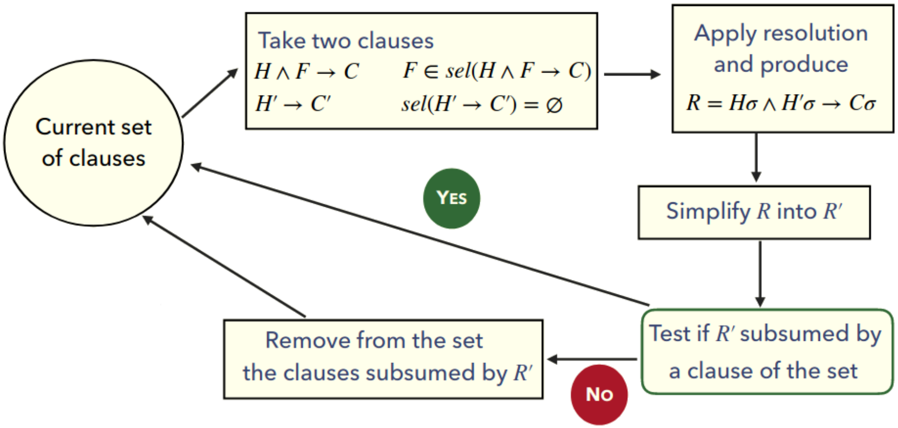
- No **occur-check**
- First step: unification whithout care of cycles
- Second step: cycle detection
- At most one function call on each node

Benchmarks

Type of query	**Version of** ProVerif				**Gain**	
	2.04		2.04_h1			
	Speed	Memory	Speed	Memory	Speed	Memory
Key secrecy & Uniqueness	2 h 48 min	162 GB	2 h 59 min	5.6 GB	0.9	29
Authentication	2 h 51 min	141 GB	3 h 16 min	22 GB	0.9	6.4
Secrecy & Authenticity	1 h 21 min	162 GB	1 h 12 min	2.4 GB	1.1	67.5

Conclusion and future work

Figure 7: Summary of the saturation procedure



Unsolved problem

Problem

Let $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{X}, \mathcal{F})$ be n terms. Find the set of bijective renaming functions $\{\rho_i\}_{i=1}^{n}$ that minimises the quantity

$$\texttt{Card}\left(\bigcup_{i=1}^{n} \texttt{Subterms}(t_i \rho_i)\right).$$

Thank you for your attention!