Reducing memory consumption of ProVerif

Vincent Cheval

Inria Paris
vincent.cheval@inria.fr

Master 2 internship of Margot Catinaud

GT FMS - Rump Session

28/03/2023

TLS with Encrypted Client Hello extension

Many features

HelloRetryRequest

Certificate-based Client Authentication

Pre-Shared Keys and Tickets

0RTT

Post Handshake Authentication

Other TLS extensions (e.g. SNI)

Status	Reachability		Equivalence		Total	
Verified	358	60 %	208	69 %	566	63 %
Stopped mostly due to OM (200-300GB)	230	39 %	87	29 %	317	36 %
Total	**592**		**300**		**892**	

Many security properties

Server Authentication

Client Authentication

Key and Transcript Agreement

Data Stream Integrity

Key Uniqueness

Downgrade Resilience

Key Secrecy

Key Indistinguishability

1RTT Data Forward Secrecy

0RTT Data Secrecy

Client Identity Privacy

Client Unlinkability

Server Extension Privacy

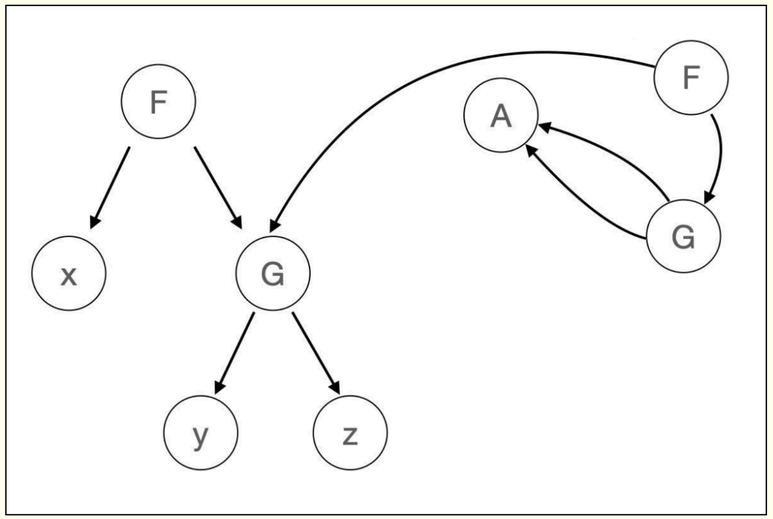
Client Extension Privacy

Server Identity Privacy

Main idea to reduce memory

Process → Translation into Horn clauses → Saturation of Horn clauses → Verification of query

Our solution: Hash consing and representation of terms in DAG



Required to modify several internal algorithms of ProVerif

Unification
Unification modulo equational theory
Matching
Subsumption of Horn Clauses
…

Preliminary results

Query	ProVerif 2.04	Prototype	Memory ratio
Key secrecy & Uniqueness	162 GB	6 GB	28,9
Authentication	141 GB	22 GB	6,4
Secrecy & Authenticity	162 GB	2 GB	67,5
Forward secrecy & Stream integrity	46 GB	11 GB	4,2
Post-handshake authentication	61 GB	39 GB	1,6
Key indistinguishability	34 GB	2 GB	18,9

Next steps?

What remains to be done?

Generalise to all types of queries

Adapt all the saturation transformation rules

Lot of implementation required

Release end of 2023?

Theoretical questions

$$F_1^1 \wedge \ldots \wedge F_{n_1}^1 \rightarrow C^1$$
$$F_1^2 \wedge \ldots \wedge F_{n_2}^2 \rightarrow C^2$$
$$\vdots$$
$$F_1^k \wedge \ldots \wedge F_{n_k}^k \rightarrow C^k$$

Set S of Horn Clauses

How to find the best variable renaming that minimise the size (in DAG) of S?

How to find a variable renaming that approximate the minimal size (in DAG) of S?