Machine-Checking Unforgeability Proofs for Signature Schemes with Tight Reductions to the Computational Diffie-Hellman Problem

Authors: Dr Francois Dupressoir (University of Bristol, f.dupressoir@bristol.ac.uk) and Sara Zain (University of Bristol, lu20465@bristol.ac.uk)
Presenter: Sara Zain

34th IEEE Computer Security Foundations Symposium
June 21-24, 2021 - Virtual Conference
EDL(EUROCRYPT 2003) and Chevallier-Mames(CM)(CRYPTO 2005) signature schemes

- The first machine-checked proofs for signature schemes based on Discrete Logarithm (DL) problem.
The EDL Scheme

\[ \mathcal{H} \in M \times N \rightarrow G, \]
\[ \mathcal{G} \in G^6 \rightarrow F_q \]

**EDL** \[ \mathcal{H}, \mathcal{G} \]

- **KGen()**
  - \( sk \leftarrow \mathbb{F}_q \)
  - return \((sk, g^{sk})\)

- **Sign_{sk}(m)**
  - \( r \leftarrow d_Y \)
  - \( h \leftarrow \mathcal{H}(m, r) \)
  - \( z \leftarrow h^{sk} \)

- **Ver_{pk}(m, (z, r, s, c))**
  - \( h \leftarrow \mathcal{H}(m, r) \)
  - \( u \leftarrow g^spk^{-c} \)
  - \( v \leftarrow h^z^{-c} \)
  - \( c' \leftarrow \mathcal{G}(g, h, pk, z, u, v) \)
  - return \( c = c' \)

NIZK proof of DL equality

**M**: set of msgs,
**N**: set of nonces,
**G**: cyclic group,
**F_q**: prime field
Definitions and the advantages

\[ \text{Exp}_{\mathcal{H}, \mathcal{G}, S, \mathcal{F}}^{\text{euf-cma}}(\cdot) \]

\[(pk, sk) \leftarrow S.KGen() \]

\[(\tilde{m}, \tilde{\sigma}) \leftarrow \mathcal{F}^{\mathcal{H}, \mathcal{G}, S, \text{Sign}_{sk}(\cdot)}(pk) \]

\[b \leftarrow S.\text{Ver}_{pk}(\tilde{m}, \tilde{\sigma}) \]

\[ \text{Adv}_{\mathcal{H}, \mathcal{G}, S}^{\text{euf-cma}}(A) := \Pr \left[ \text{Exp}_{\mathcal{H}, \mathcal{G}, S, \mathcal{A}}^{\text{euf-cma}}(\cdot) : b \land \tilde{m} \notin Q_S \right] \]

\[ \text{Adv}_{G, g_G, n}^{\text{cdh}}(A) := \Pr \left[ \text{Exp}_{\mathcal{A}}^{\text{cdh}}(G, g_G, n) : r = g_G^{ab} \right] \]

\[ \text{Adv}_{\text{EDL}}^{\text{euf-cma}}(A) \leq \text{Adv}_{G, g_G, n}^{\text{cdh}} + \epsilon \]
Intuition

Formal proof in 4 steps
1) Refactoring
2) Embedding
3) Simulation
4) Reduction
Formalisation

• Machine-checked proofs
  – Emerging approach, ensures the correctness of reasoning steps (smt solvers & automated theorem provers).

• EasyCrypt
  – Follows the code-based, game-based approach to reductionist argument
  – Security goals & assumptions are modelled as probabilistic programs (called experiments/games)
Overview of the sequence of games & the Shim

The EDL proof Shim

Sequence of games
3) Simulation (pRHL judgment)

$$\text{Pr} \left[ \text{Game}^{EDL}_{H', G, S_1, \mathcal{F}}() : \text{bad}_G \right] = \text{Pr} \left[ \text{Game}^{EDL}_{H', G, S_2, \mathcal{F}}() : \text{bad}_G \right]$$

$$\text{Pr} \left[ \text{Game}^{EDL}_{H', G, S_1, \mathcal{F}}() : \text{win} \right] \leq \text{Pr} \left[ \text{Game}^{EDL}_{H', G, S_2, \mathcal{F}}() : \text{win} \right] + \text{Pr} \left[ \text{Game}^{EDL}_{H', G, S_1, \mathcal{F}}() : \text{bad}_G \right]$$
4) Reduction

For any forger $\mathcal{F}$, the forger’s success probability either
- the forger solves its given CDH instance ($z = h^{sk}$) or
- the forger exploited the unsoundness in the proof of discrete logarithm equality ($z \neq h^{sk}$)

\[
\Pr \left[ \text{Game}^{\text{EDL}}_{\mathcal{H}', \mathcal{G}, S_2, \mathcal{F}}() : \text{win} \right] = \Pr \left[ \text{Game}^{\text{EDL}}_{\mathcal{H}', \mathcal{G}, S_2, \mathcal{F}}() : \text{win} \land \tilde{z} = h^{sk} \right] + \Pr \left[ \text{Game}^{\text{EDL}}_{\mathcal{H}', \mathcal{G}, S_2, \mathcal{F}}() : \text{win} \land \tilde{z} \neq h^{sk} \right]
\]
B. Chevallier-Mames (CM) - Crypto 2005

- Proposed a new signature scheme that also has a tight security reduction to CDH but whose resulting signatures are smaller than EDL signatures.
- Message is not included in the random oracle query to $\mathcal{H}$ whose output serves as the second base for the proof of discrete logarithm equality.
Summary

- First machine-checked proof for signature scheme based on discrete logarithm.
- We identify a proof schema that we believe applies more broadly.
- We refine some EasyCrypt techniques to reduce the proof burden and support better proof reuse. (Shim)