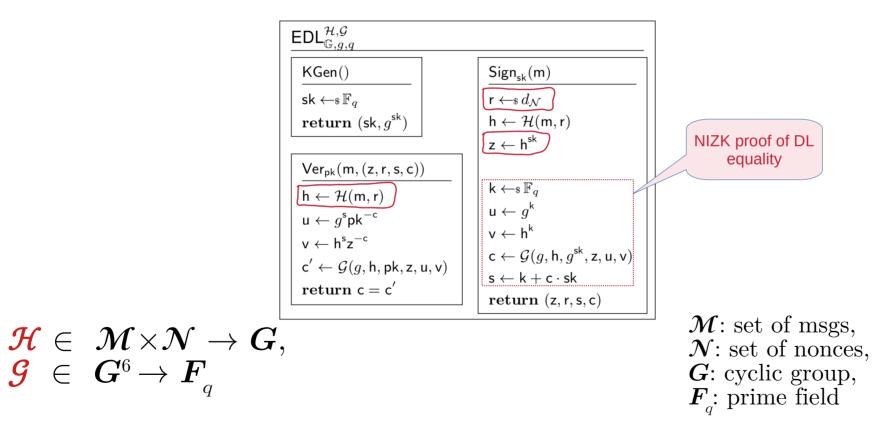
Machine-Checking Unforgeability Proofs for Signature Schemes with Tight Reductions to the Computational Diffie-Hellman Problem

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



Definitions and the advantages

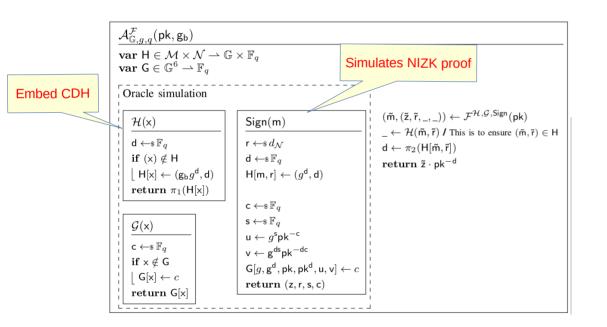
$$\begin{split} & \frac{\mathsf{Exp}^{\mathsf{euf-cma}}_{\mathcal{H},\mathcal{G},\mathcal{S},\mathcal{F}}()}{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{S}.\mathsf{KGen}()} \\ & (\tilde{\mathsf{m}},\tilde{\sigma}) \leftarrow \mathcal{F}^{\mathcal{H},\mathcal{G},\mathcal{S}.\mathsf{Sign}_{\mathsf{sk}}(\cdot)}(\mathsf{pk}) \\ & \mathsf{b} \leftarrow \mathcal{S}.\mathsf{Ver}_{\mathsf{pk}}(\tilde{\mathsf{m}},\tilde{\sigma}) \end{split}$$

$$\begin{aligned} & \mathsf{Exp}_{\mathcal{A}}^{\mathsf{cdh}}(G, g_G, n) \\ & \mathsf{a} \leftarrow \ensuremath{\mathbb{F}_q} \\ & \mathsf{b} \leftarrow \ensuremath{\mathbb{F}_q} \\ & \mathsf{r} \leftarrow \mathcal{A}_{G, g_G, n}(g_G^{\mathsf{a}}, g_G^{\mathsf{b}}) \end{aligned}$$

$$\mathsf{Adv}^{\mathsf{euf-cma}}_{\mathcal{H},\mathcal{G},\mathcal{S}}(\mathcal{A}) := \Pr\left[\mathsf{Exp}^{\mathsf{euf-cma}}_{\mathcal{H},\mathcal{G},\mathcal{S},\mathcal{A}}() : \mathsf{b} \land \tilde{\mathsf{m}} \notin \mathcal{Q}_{\mathcal{S}}\right] \qquad \mathsf{Adv}^{\mathsf{cdh}}_{G,g_G,n}(\mathcal{A}) := \Pr\left[\mathsf{Exp}^{\mathsf{cdh}}_{\mathcal{A}}(G,g_G,n) : \mathsf{r} = g^{\mathsf{ab}}_{G}\right]$$

$$Adv \frac{euf - cma}{EDL}(A) \le Adv^{cdh} + \mathcal{E}$$

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

$Game_{\mathcal{H},\mathcal{G},\mathcal{S},\mathcal{F}}^{EDL}()$		
	$sk \gets_{\!\!\!\$} \mathbb{F}_q$	
2:	$b \gets \mathbb{F}_q$	
3:	$pk \gets g^{sk}$	
4:	$g_b \gets g^b$	
5:	$(\tilde{m}, (\tilde{z}, \tilde{r}, \tilde{s}, \tilde{c})) \leftarrow \mathcal{F}^{\mathcal{H}, \mathcal{G}, \mathcal{S}}(pk)$	
6:	$h \leftarrow \mathcal{H}(\tilde{m},\tilde{r})$	
7:	$u \gets g^{\tilde{s}} p k^{-\tilde{c}}$	
8:	$v \gets h^{\tilde{s}} p k^{-\tilde{c}}$	
9:	$c \gets \mathcal{G}(g,h,pk,\tilde{z},u,v)$	
10:	$win \gets \tilde{c} = c \land \tilde{m} \notin \mathcal{Q}_\mathcal{S}$	

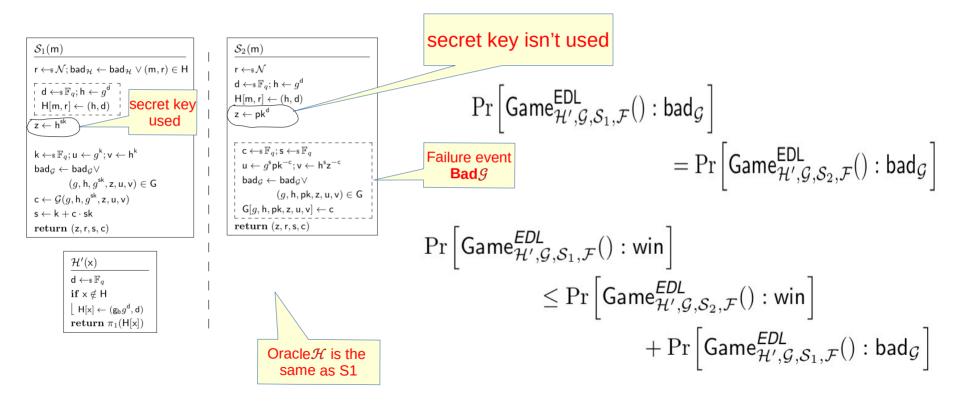
$\label{eq:states} \begin{split} & \frac{\mathcal{S}_0(m)}{r \leftarrow s \mathcal{N}} \\ & h \leftarrow \mathcal{H}(m,r) \\ & z \leftarrow h^{sk} \end{split}$	$eq:started_st$	$ \begin{vmatrix} \mathcal{S}_2(m) \\ r \leftarrow s \mathcal{N} \\ d \leftarrow s \mathbb{F}_q; h \leftarrow g^d \\ H[m, r] \leftarrow (h, d) \\ z \leftarrow pk^d \end{vmatrix} $
$\begin{split} k & \leftarrow s \mathbb{F}_q; u \leftarrow g^k; v \leftarrow h^k \\ c & \leftarrow \mathcal{G}(g, h, g^sk, z, u, v) \\ s & \leftarrow k + c \cdot sk \\ \mathbf{return} \ (z, r, s, c) \end{split}$	$\begin{array}{c c c c c c c } k \leftarrow & & \mathbb{F}_q; \mathbf{u} \leftarrow g^k; \mathbf{v} \leftarrow \mathbf{h}^k \\ & & bad_{\mathcal{G}} \leftarrow bad_{\mathcal{G}} \lor \\ & & & (g, \mathbf{h}, g^{sk}, z, u, v) \in G \\ & & c \leftarrow \mathcal{G}(g, \mathbf{h}, g^{sk}, z, u, v) \\ & & s \leftarrow k + c \cdot sk \\ & & \mathbf{return} \ (z, r, s, c) \end{array}$	$ \begin{bmatrix} c \leftarrow s \mathbb{F}_q; s \leftarrow s \mathbb{F}_q \\ u \leftarrow g^s p k^{-c}; v \leftarrow h^s z^{-c} \\ b a d_{\mathcal{G}} \leftarrow b a d_{\mathcal{G}} \lor \\ (g, h, p k, z, u, v) \in G \\ \\ G[g, h, p k, z, u, v] \leftarrow c \\ \mathbf{return} (z, r, s, c) $
$ \begin{array}{c c} \displaystyle \frac{\mathcal{H}(x)}{h \leftarrow s \ \mathbb{G}} & & \displaystyle \frac{\mathcal{G}(x)}{c \leftarrow s \ \mathbb{F}_q} \\ \displaystyle \mathbf{if} \ x \notin H & & \displaystyle \mathbf{if} \ x \notin G \\ \displaystyle \lfloor \ H[x] \leftarrow h & & \displaystyle \lfloor \ G[x] \leftarrow c \\ \displaystyle \mathbf{return} \ H[x] \end{array} \right. $	$ \begin{array}{c} \displaystyle \frac{\mathcal{H}'(x)}{d \leftarrow \mathfrak{s} \mathbb{F}_q} \\ \displaystyle \mathbf{if} x \notin H \\ \displaystyle \left\lfloor H[x] \leftarrow (g_{b} g^{d}, d) \\ \mathbf{return} \pi_1(H[x]) \end{array} \right. \end{array} $	

The EDL proof Shim

Sequence of games

7

3) Simulation (pRHL judgment)



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 $(z \neq h^{sk})$

$$\begin{split} \Pr \begin{bmatrix} \mathsf{Game}_{\mathcal{H}',\mathcal{G},\mathcal{S}_{2},\mathcal{F}}^{\mathsf{EDL}}():\mathsf{win} \end{bmatrix} \\ &= \Pr \begin{bmatrix} \mathsf{Game}_{\mathcal{H}',\mathcal{G},\mathcal{S}_{2},\mathcal{F}}^{\mathsf{EDL}}():\mathsf{win} \wedge \tilde{\mathsf{z}} = \mathsf{h}^{\mathsf{sk}} \end{bmatrix} \\ &+ \Pr \begin{bmatrix} \mathsf{Game}_{\mathcal{H}',\mathcal{G},\mathcal{S}_{2},\mathcal{F}}^{\mathsf{EDL}}():\mathsf{win} \wedge \tilde{\mathsf{z}} \neq \mathsf{h}^{\mathsf{sk}} \end{bmatrix} \end{split}$$

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)