

Computer-aided cryptographic proofs and designs

Gilles Barthe

IMDEA Software Institute, Madrid, Spain

Based on joint work with:

Benjamin Grégoire (INRIA Sophia Antipolis, France)

César Kunz (U. Politecnica Madrid/IMDEA, Spain)

Yassine Lakhnech (U. de Grenoble/CNRS, France)

Santiago Zanella Béguelin (MSR Cambridge, UK)

Historical perspective

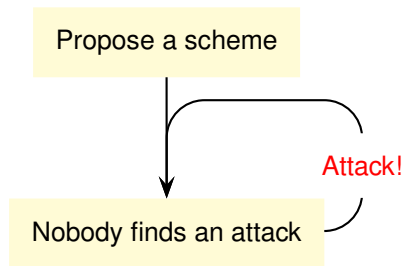
Propose a scheme

```
graph TD; A[Propose a scheme] --> B[Nobody finds an attack]
```

Nobody finds an attack

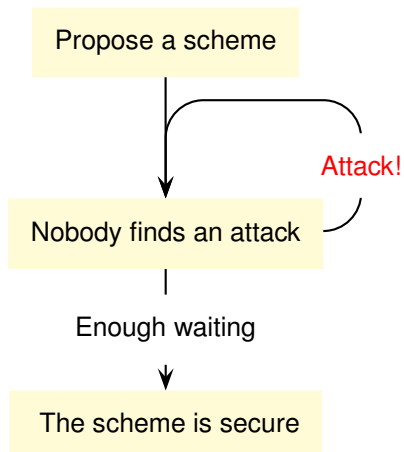
Before 1980

Historical perspective



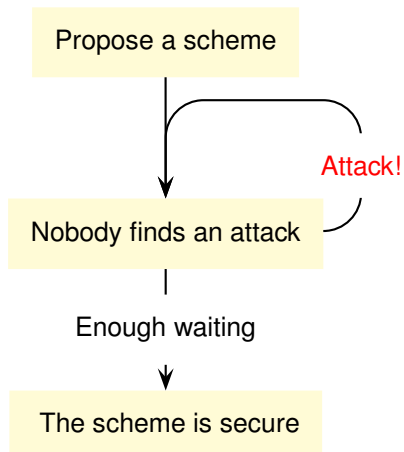
Before 1980

Historical perspective



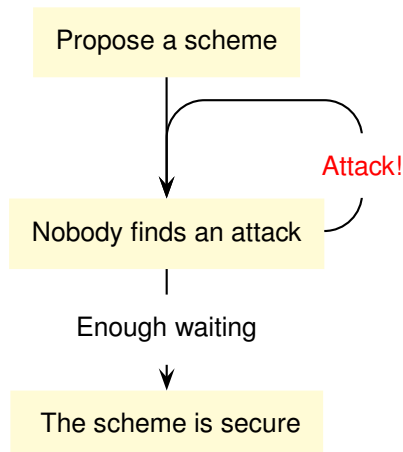
Before 1980

Historical perspective



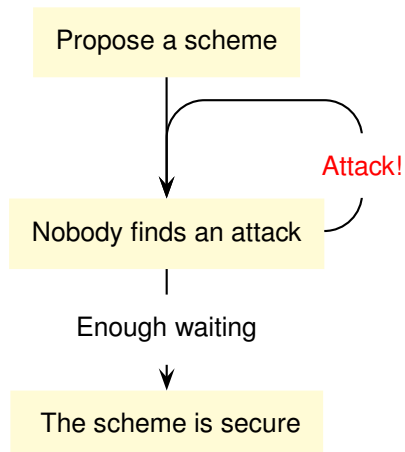
How much time is *enough*?

Historical perspective



1 year? 5 years? 10 years?

Historical perspective

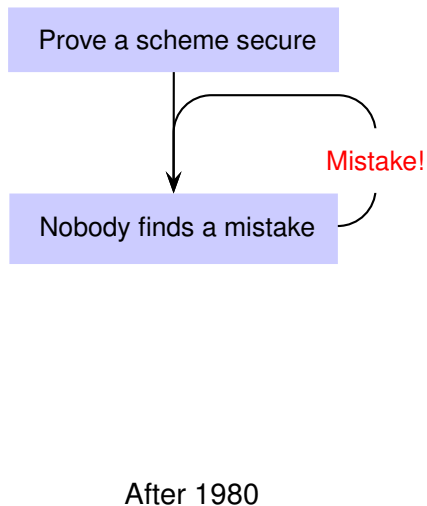
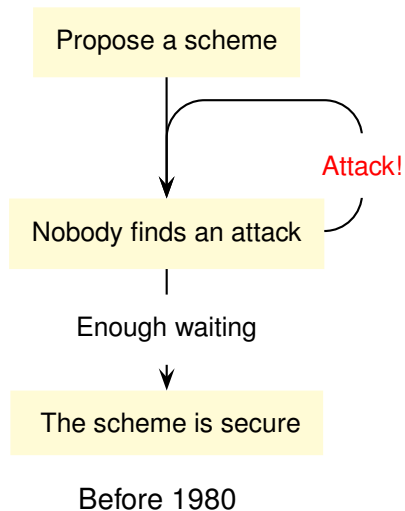


Before 1980

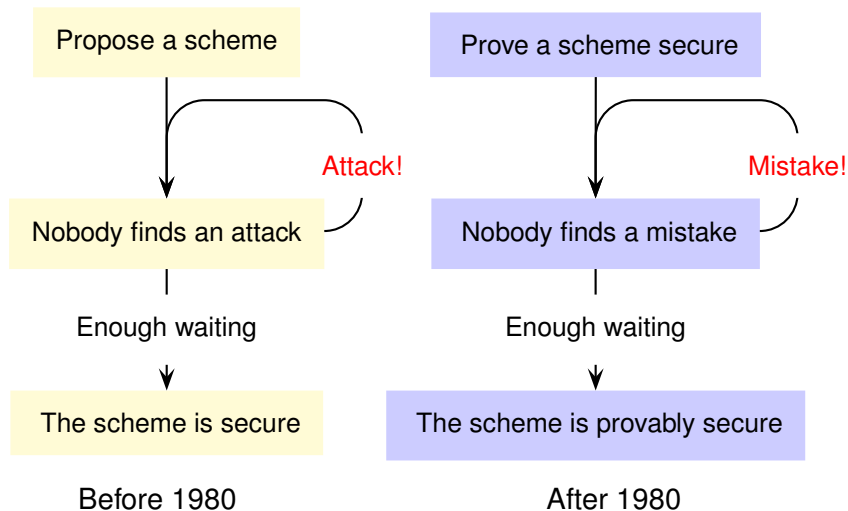
Prove a scheme secure

After 1980

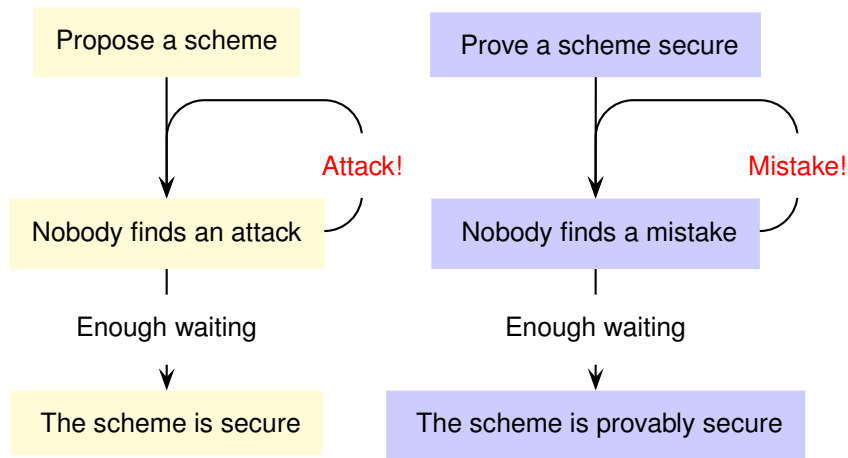
Historical perspective



Historical perspective

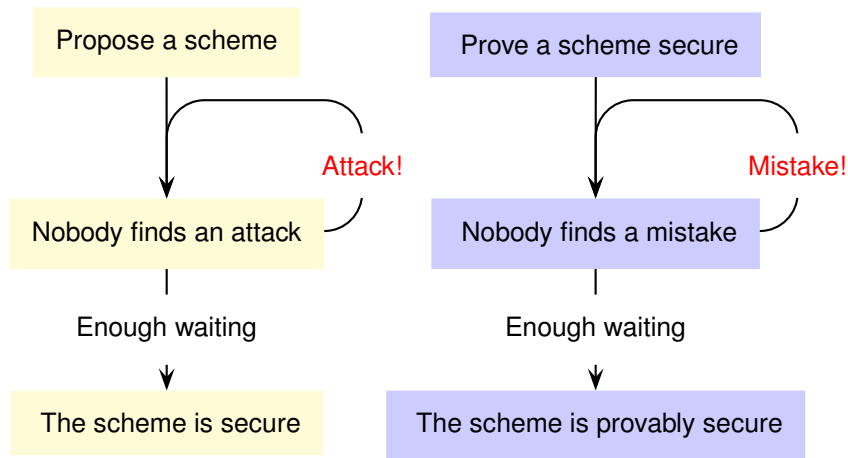


Historical perspective



How much time is *enough*?

Historical perspective



1 week? 1 year? 5 years? 10 years? 20 years?

A famous example: RSA-OAEP

Oracle $\text{Enc}_{pk}(m)$:

$r \xleftarrow{\$} \{0, 1\}^{k_0}$;

$s \leftarrow G(r) \oplus (m \parallel 0^{k_1})$;

$t \leftarrow H(s) \oplus r$;

return $f_{pk}(s \parallel t)$

Oracle $\text{Dec}_{sk}(c)$:

$(s, t) \leftarrow f_{sk}^{-1}(c)$;

$r \leftarrow t \oplus H(s)$;

if $[s \oplus G(r)]_{k_1} = 0^{k_1}$ then return $[s \oplus G(r)]^n$
else return \perp

A famous example: RSA-OAEP

Oracle $\text{Enc}_{pk}(m)$:

$r \xleftarrow{\$} \{0, 1\}^{k_0}$;
 $s \leftarrow G(r) \oplus (m \parallel 0^{k_1})$;
 $t \leftarrow H(s) \oplus r$;
return $f_{pk}(s \parallel t)$

Oracle $\text{Dec}_{sk}(c)$:

$(s, t) \leftarrow f_{sk}^{-1}(c)$;
 $r \leftarrow t \oplus H(s)$;
if $[s \oplus G(r)]_{k_1} = 0^{k_1}$ then return $[s \oplus G(r)]^n$
else return \perp

Game IND-CCA2 :

$(sk, pk) \leftarrow \mathcal{KG}(\cdot)$;
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk)$;
 $b \xleftarrow{\$} \{0, 1\}$;
 $c^* \leftarrow \text{Enc}(pk, m_b)$;
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma)$;
return $b = b'$

A famous example: RSA-OAEP

Oracle $\text{Enc}_{pk}(m)$:

$r \xleftarrow{\$} \{0, 1\}^{k_0}$;
 $s \leftarrow G(r) \oplus (m \parallel 0^{k_1})$;
 $t \leftarrow H(s) \oplus r$;
return $f_{pk}(s \parallel t)$

Oracle $\text{Dec}_{sk}(c)$:

$(s, t) \leftarrow f_{sk}^{-1}(c)$;
 $r \leftarrow t \oplus H(s)$;
if $[s \oplus G(r)]_{k_1} = 0^{k_1}$ then return $[s \oplus G(r)]^n$
else return \perp

Oracle $G(x)$:

if $x \notin \text{dom}(L_G)$ then $L_G[x] \xleftarrow{\$} \{0, 1\}^{n+k_1}$;
return $L_G[x]$

Oracle $H(x)$:

if $x \notin \text{dom}(L_H)$ then $L_H[x] \xleftarrow{\$} \{0, 1\}^{k_0}$;
return $L_H[x]$

Game IND-CCA2 :

$(sk, pk) \leftarrow \mathcal{KG}()$;
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk)$;
 $b \xleftarrow{\$} \{0, 1\}$;
 $c^* \leftarrow \text{Enc}(pk, m_b)$;
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma)$;
return $b = b'$

Game POW :

$(sk, pk) \leftarrow \mathcal{KG}()$;
 $y \xleftarrow{\$} \{0, 1\}^{n+k_1}$;
 $z \xleftarrow{\$} \{0, 1\}^{k_0}$;
 $y' \leftarrow \mathcal{I}(f_{pk}(y \parallel z))$;
return $y = y'$

A famous example: RSA-OAEP

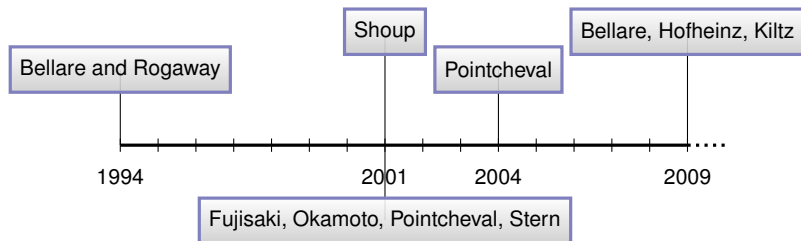
For every IND-CCA2 adversary \mathcal{A} there exists an inverter \mathcal{I} s.t.

$$\begin{aligned}\text{Adv}_{\text{IND-CCA2}}(\mathcal{A}) &= \left| \Pr_{\text{IND-CCA2}}[b = b'] - \frac{1}{2} \right| \\ &\leq \mathbf{Succ}_f^{\text{POW}}(\mathcal{I}) + \frac{3q_D q_G + q_D^2 + 4q_D + q_G}{2^{k_0}} + \frac{2q_D}{2^{k_1}}\end{aligned}$$

where

$$\mathbf{Succ}_f^{\text{POW}} = \Pr_{\text{POW}}[y = y']$$

A famous example: RSA-OAEP



1994 Purported proof of chosen-ciphertext security

2001 Proof establishes a weaker security notion, but desired security can be achieved

- ① ...for a modified scheme, or
- ② ...under stronger assumptions

2004 Filled gaps in Fujisaki et al. 2001 proof

2009 Security definition needs to be clarified

2010 Filled gaps in 2004 proof

What's wrong with provable security?

- ▶ *In our opinion, many proofs in cryptography have become essentially unverifiable. Our field may be approaching a crisis of rigor.* M. Bellare and P. Rogaway, 2004-2006
- ▶ *Do we have a problem with cryptographic proofs? Yes, we do [...] We generate more proofs than we carefully verify (and as a consequence some of our published proofs are incorrect).* S. Halevi, 2005

Provable security as program verification!

CertiCrypt/EasyCrypt project, 2006-

High-assurance cryptographic proofs

- ▶ based on rigorous methods from programming languages
 - program verification
 - compiler verification
- ▶ with machine support building upon off-the-shelf tools

Provable security as program verification!

CertiCrypt/EasyCrypt project, 2006-

High-assurance cryptographic proofs

- ▶ based on rigorous methods from programming languages
 - program verification
 - compiler verification
- ▶ with machine support building upon off-the-shelf tools

Sources of inspiration

- ▶ *I advocate creating an automated tool to help [...] writing and checking [...] our proofs.* Halevi, 2005
- ▶ *The possibility for tools [to help write and verify proofs] has always been [...] one of the reasons why we focused on code-based games.* Bellare and Rogaway, 2004-2006

Code-based game-playing proofs

(Bellare & Rogaway 2004, Halevi 2005)

Games as probabilistic programs

$\mathcal{C} ::=$	$\mathcal{V} \leftarrow \mathcal{E}$	assignment
	$\mathcal{V} \xleftarrow{\$} \mathcal{D}$	random sampling
	$\mathcal{C}; \mathcal{C}$	sequence
	if \mathcal{E} then \mathcal{C} else \mathcal{C}	conditional
	while \mathcal{E} do \mathcal{C}	while loop
	$\mathcal{V} \leftarrow \mathcal{P}(\mathcal{E}, \dots, \mathcal{E})$	procedure call

- ▶ For cryptographers: rigorous notation for games
- ▶ In our work: rigorous justification of game-based proofs

The game-playing approach

(Shoup 2004, Bellare & Rogaway 2004, Halevi, 2005)

For every feasible adversary \mathcal{A} against scheme \mathbf{S} (wrt goal \mathbf{G})
there exists a feasible adversary \mathcal{B} against assumption \mathbf{H} st

$$\Pr_{G_a}[\mathcal{A} \text{ breaks } \mathbf{S}] \leq h(\Pr_{G_h}[\mathcal{B} \text{ breaks } \mathbf{H}])$$

The game-playing approach

(Shoup 2004, Bellare & Rogaway 2004, Halevi, 2005)

For every feasible adversary \mathcal{A} against scheme \mathbf{S} (wrt goal \mathbf{G})
there exists a feasible adversary \mathcal{B} against assumption \mathbf{H} st

$$\Pr_{G_a}[\mathcal{A} \text{ breaks } \mathbf{S}] \leq h(\Pr_{G_h}[\mathcal{B} \text{ breaks } \mathbf{H}])$$

Game G_a :

...
... $\leftarrow \mathcal{A}(\dots)$;
...

Game G_1 :

...
...
...

...

Game G_h :

...
... $\leftarrow \mathcal{B}(\dots)$;
...

$$\Pr_{G_a}[\mathcal{A} \text{ breaks } \mathbf{S}] \leq h_1(\Pr_{G_1}[E_1]) \leq \dots \leq h(\Pr_{G_h}[\mathcal{B} \text{ breaks } \mathbf{H}])$$

Example: IND-CPA security of BR93

Game IND-CPA :

$(sk, pk) \leftarrow \mathcal{KG}(\);$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $c^* \leftarrow \text{Enc}(pk, m_b);$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$
return $b = b'$

$\text{Enc}_{pk}(m) :$

$r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \leftarrow G(r) \oplus m;$
 $y \leftarrow f_{pk}(r) \parallel s;$
return y

Game OW :

$(sk, pk) \leftarrow \mathcal{KG}();$
 $y \xleftarrow{\$} \{0, 1\}^\ell;$
 $y' \leftarrow \mathcal{I}(f_{pk}(y));$
return $y = y'$

$G(x) :$

if $x \notin \text{dom}(L_G)$ then $L_G[x] \xleftarrow{\$} \{0, 1\}^k;$
return $L_G[x]$

For every IND-CPA adversary \mathcal{A} making at most q_G queries to G , there exists an inverter \mathcal{I} against OW such that

$$\left| \Pr_{\text{IND-CPA}} [b = b'] - \frac{1}{2} \right| \leq q_G \text{ Succ}_f^{\text{OW}}(\mathcal{I})$$

Step 1: failure event

Game G_0 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $g \leftarrow G(r);$
 $s \leftarrow g \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Game G_1 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $g \xleftarrow{\$} \{0, 1\}^k;$
 $s \leftarrow g \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Step 1: failure event

Game G_0 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $g \leftarrow G(r);$
 $s \leftarrow g \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Game G_1 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $g \xleftarrow{\$} \{0, 1\}^k;$
 $s \leftarrow g \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

The games are equivalent until the adversary queries G with r

$$|\Pr_{\text{IND-CPA}}[b = b'] - \Pr_{G_1}[b = b']| \leq \Pr_{G_1}[r \in Q_G]$$

Step 2: optimistic sampling

Game G_1 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $g \leftarrow \{0, 1\}^k;$
 $s \leftarrow g \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Game G_2 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $g \leftarrow s \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Step 2: optimistic sampling

Game G_1 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $g \leftarrow \{0, 1\}^k;$
 $s \leftarrow g \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Game G_2 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $g \leftarrow s \oplus m_b;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Games are equivalent and c^* is independent from b , hence

$$\left| \Pr_{\text{IND-CPA}}[b = b'] - \frac{1}{2} \right| \leq \Pr_{G_2}[r \in Q_G]$$

Step 3: reduction

Game G_2 :

```
 $L_G \leftarrow \emptyset; Q_G \leftarrow [];$   
 $(sk, pk) \leftarrow \mathcal{KG}();$   
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$   
 $r \xleftarrow{\$} \{0, 1\}^\ell;$   
 $s \xleftarrow{\$} \{0, 1\}^k;$   
 $c^* \leftarrow f_{pk}(r) \parallel s;$   
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$ 
```

Game OW :

```
 $(sk, pk) \leftarrow \mathcal{KG}();$   
 $y \xleftarrow{\$} \{0, 1\}^\ell;$   
 $y' \leftarrow \mathcal{I}(f_{pk}(y));$   
return  $y = y'$ 
```

Adversary $\mathcal{I}(x)$:

```
 $L_G \leftarrow \emptyset; Q_G \leftarrow [];$   
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$   
 $s \xleftarrow{\$} \{0, 1\}^k; y \leftarrow x \parallel s;$   
 $b' \leftarrow \mathcal{A}_2(pk, y, \sigma);$   
 $i \xleftarrow{\$} [1, |Q_G|];$   
return  $Q_G[i];$ 
```

Step 3: reduction

Game G_2 :

$L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(sk, pk) \leftarrow \mathcal{KG}();$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $c^* \leftarrow f_{pk}(r) \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, c^*, \sigma);$

Game OW :

$(sk, pk) \leftarrow \mathcal{KG}();$
 $y \xleftarrow{\$} \{0, 1\}^\ell;$
 $y' \leftarrow \mathcal{I}(f_{pk}(y));$
return $y = y'$
Adversary $\mathcal{I}(x)$:
 $L_G \leftarrow \emptyset; Q_G \leftarrow [];$
 $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1(pk);$
 $s \xleftarrow{\$} \{0, 1\}^k; y \leftarrow x \parallel s;$
 $b' \leftarrow \mathcal{A}_2(pk, y, \sigma);$
 $i \xleftarrow{\$} [1, |Q_G|];$
return $Q_G[i];$

Inverter wins with probability $\frac{1}{q_G}$ if $r \in Q_G$, and 0 otherwise

$$\left| \Pr_{\text{IND-CPA}} [b = b'] - \frac{1}{2} \right| \leq q_G \text{ Succ}_f^{\text{OW}}(\mathcal{I})$$

pRHL: a Relational Hoare Logic for pWHILE

(after Benton 2004)

- ▶ Judgment:

$$c_1 \sim c_2 : P \Rightarrow Q$$

where P and Q are relations on memories

- ▶ Validity:

$$\models c_1 \sim c_2 : P \Rightarrow Q$$

iff for all memories m_1 and m_2

$$(m_1, m_2) \models P \rightarrow (\llbracket c_1 \rrbracket_{m_1}, \llbracket c_2 \rrbracket_{m_2}) \models Q^\sharp$$

- ▶ Lifting Q^\sharp asserts *existence* of maximal flow in flow network (beware of existential quantification)

Conditionals

$$\frac{\vDash c_1 \sim c : P \wedge e\langle 1 \rangle \Rightarrow Q \quad \vDash c_2 \sim c : P \wedge \neg e\langle 1 \rangle \Rightarrow Q}{\vDash \text{if } e \text{ then } c_1 \text{ else } c_2 \sim c : P \Rightarrow Q}$$

Assignment

$$\frac{}{\vDash x \leftarrow e \sim \text{nil} : Q\{x\langle 1 \rangle := e\langle 1 \rangle\} \Rightarrow Q}$$

Random assignment

$$\frac{f \text{ is 1-1 and } Q' \stackrel{\text{def}}{=} \forall v, Q\{x\langle 1 \rangle := f v, x\langle 2 \rangle := v\}}{\vDash x \xleftarrow{\$} A \sim x \xleftarrow{\$} A : Q' \Rightarrow Q}$$

Adversary calls

$$\frac{\forall \mathcal{O}. \vDash z \leftarrow \mathcal{O}(\vec{w}) \sim z \leftarrow \mathcal{O}(\vec{w}) : Q \wedge =_W \Rightarrow Q \wedge =_{\{z\}}}{\vDash x \leftarrow \mathcal{A}(\vec{y}) \sim x \leftarrow \mathcal{A}(\vec{y}) : Q \wedge =_Y \Rightarrow Q \wedge =_{\{x\}}}$$

Cryptographic reasoning with pRHL

pRHL captures common patterns in cryptographic proofs

- ▶ Failure events: if $\models c_1 \sim c_2 : P \Rightarrow \neg F\langle 2 \rangle \rightarrow Q_1\langle 1 \rangle \leftrightarrow Q_2\langle 2 \rangle$ then

$$(m_1, m_2) \models P \implies |\Pr_{c_1, m_1}[Q_1] - \Pr_{c_2, m_2}[Q_2]| \leq \Pr_{c_2, m_2}[F]$$

- ▶ Bridging steps: if $\models c_1 \sim c_2 : P \Rightarrow =_Y$ and Q depends on Y , then

$$(m_1, m_2) \models P \implies \Pr_{c_1, m_1}[Q] = \Pr_{c_2, m_2}[Q]$$

- ▶ Reductions: if $\models c_1 \sim c_2 : P \Rightarrow Q_1\langle 1 \rangle \rightarrow Q_2\langle 2 \rangle$, then

$$(m_1, m_2) \models P \implies \Pr_{c_1, m_1}[Q_1] \leq \Pr_{c_2, m_2}[Q_2]$$

- ▶ Eager/lazy sampling

apRHL: quantitative extension for

▶ Statistical distance

▶ Differential privacy

Cryptographic reasoning with pRHL

pRHL captures common patterns in cryptographic proofs

- ▶ Failure events: if $\models c_1 \sim c_2 : P \Rightarrow \neg F\langle 2 \rangle \rightarrow Q_1\langle 1 \rangle \leftrightarrow Q_2\langle 2 \rangle$ then

$$(m_1, m_2) \models P \implies |\Pr_{c_1, m_1}[Q_1] - \Pr_{c_2, m_2}[Q_2]| \leq \Pr_{c_2, m_2}[F]$$

- ▶ Bridging steps: if $\models c_1 \sim c_2 : P \Rightarrow =_Y$ and Q depends on Y , then

$$(m_1, m_2) \models P \implies \Pr_{c_1, m_1}[Q] = \Pr_{c_2, m_2}[Q]$$

- ▶ Reductions: if $\models c_1 \sim c_2 : P \Rightarrow Q_1\langle 1 \rangle \rightarrow Q_2\langle 2 \rangle$, then

$$(m_1, m_2) \models P \implies \Pr_{c_1, m_1}[Q_1] \leq \Pr_{c_2, m_2}[Q_2]$$

- ▶ Eager/lazy sampling

apRHL: quantitative extension for

- ▶ Statistical distance
- ▶ Differential privacy

Tool support and examples

CertiCrypt: formally verified Coq libraries

- ▶ Optimizations and probabilistic relational Hoare logic
- ▶ Verified against operational semantics based on ALEA

EasyCrypt: SMT-based verification tool

- ▶ Probabilistic relational Hoare logic
- ▶ Verification condition generation + why3 back-end
- ▶ Accessible to cryptographers

Examples

- ▶ Crypto: public-key encryption, block ciphers, signatures, hash designs, zero-knowledge proofs of knowledge, authenticated key exchange protocols
- ▶ Differential privacy: continuous statistics, approximation algorithms, synthetic databases, 2-party computation

At Crypto 2011 the Program Chair, Phil Rogaway, explained that the paper [2] was chosen for the Best Paper Award “overwhelmingly” by the Program Committee, which “praised the work for its broad appeal. . . and its potential impact.”

Cousot?

At Crypto 2011 the Program Chair, Phil Rogaway, explained that the paper [2] was chosen for the Best Paper Award “overwhelmingly” by the Program Committee, which “praised the work for its broad appeal. . . and its potential impact.”



Cousot? No, Foucault!

At Crypto 2011 the Program Chair, Phil Rogaway, explained that the paper [2] was chosen for the Best Paper Award “overwhelmingly” by the Program Committee, which “praised the work for its broad appeal. . . and its potential impact.”

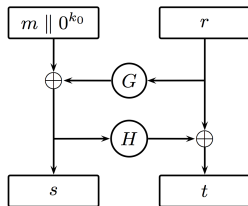
And, indeed, [2] is in some sense a remarkable achievement. Usually one has to leave the sciences entirely if one wishes to find works of scholarship — for example, Michel Foucault’s turgid 760-page three-volume philosophical treatise on sex [14] — that have been so successful in turning something that should be interesting and accessible to everyone into something lengthy, unreadable, and boring.



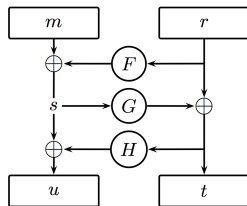
Synthesis of encryption schemes

Over 100 variants of OAEP in the literature

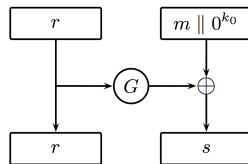
- ▶ Are they all secure?
- ▶ Are there better ones?
- ▶ Are there common patterns in proofs?
- ▶ Can proofs be automated?



(a) OAEP



(b) 3-round OAEP



(c) SAEP

Generation

\mathcal{E}	::=	m	input message
		0	zero bitstring
		\mathcal{R}	uniform random bitstring
		$\mathcal{E} \oplus \mathcal{E}$	xor
		$\mathcal{E} \parallel \mathcal{E}$	concatenation
		$H(\mathcal{E})$	hash
		$f(\mathcal{E})$	trapdoor permutation

Filtering

Eliminate schemes that are not :

- ▶ invertible $f(r)$
- ▶ IND-CPA
 - is decryption possible without a key? $m \parallel f(r)$
 - is encryption randomized? $f(m)$
 - is randomness extractable without a key? $r \parallel f(m \oplus r)$
- ▶ IND-CCA2
 - is encryption malleable? $f(r) \parallel m \oplus G(r)$

Deducibility relation

$$\frac{e \vdash e_1 \quad e \vdash e_2}{e \vdash e_1 \parallel e_2} [\text{Conc}] \quad \frac{e \vdash e_1 \quad e \vdash e_2}{e \vdash (e_1 \oplus e_2)} \downarrow [\text{Xor}] \quad \frac{e \vdash e'}{e \vdash H(e')} [\text{H}]$$
$$\frac{e \vdash e_1 \parallel e_2}{e \vdash e_j} [\text{Proj}_j] \quad \frac{e \vdash e'}{e \vdash f(e')} [f] \quad \boxed{\frac{e \vdash f(e')}{e \vdash e'} [\text{finv}]}$$

Chosen-plaintext security

Step 1: proof finding

Optimistic sampling Replace $e \oplus r$, where r is fresh, by r

Permutation Replace $f(r)$, where r is fresh, by r

Failure event Replace $H(e)$ by fresh r

Probability Compute probability of $b = b'$ or $e \in L$

Step 2: proof generation and proof checking

Generation Output EasyCrypt file

Checking Independent verification of EasyCrypt file (< 120 s)

Chosen-ciphertext security

If decryption oracle is of the form

```

$$u_0 \dots u_n, t_n \leftarrow \text{Extract}(c);$$

$$\text{for } i \leftarrow n \dots 1 \text{ do } t_{i-1} \leftarrow u_i \oplus H_i(t_i);$$

$$\text{if Test}(\vec{t}, \vec{u}, c) \text{ then GetMsg}(\vec{t}) \text{ else return } \perp$$

```

and IND-CPA and IND-CPA-like properties then IND-CCA2

Proof intuition

- ▶ Plaintext-awareness: infeasible to get valid ciphertext otherwise than by encrypting a known plaintext

IND-CPA + plaintext-awareness \implies IND-CCA2

- ▶ Successively modify decryption oracle to reject ciphertexts for which corresponding hash queries have not been made
- ▶ Yields plaintext extractor and reduction to IND-CPA
- ▶ IND-CPA-like properties provide bounds for failure events

Experiments

- ▶ Generated over 100,000 schemes
- ▶ Filters leave 4,500 schemes
- ▶ Proved IND-CPA security of 3,000 schemes
- ▶ Proved IND-CCA2 security of 2,000 schemes

Assessment

- ▶ Systematic exploration of design space
- ▶ Tens of papers automated
- ▶ IND-CCA2 checker fails on redundant-free schemes

ZAEP

(with David Pointcheval)

Two minimal schemes

$$\text{BR93} : f(r) \parallel (G(r) \oplus m) \quad \text{ZAEP} : f(r \parallel G(r) \oplus m)$$

ZAEP is redundant-free

$$\text{Dec}(c) : r \parallel t \leftarrow f_{sk}^{-1}(c); g \leftarrow G(r); \text{return } t \oplus g$$

INDCCA Security of ZAEP for RSA exponent 2 and 3

$$\left| \Pr_{\text{IND-CCA2}}[b = b'] - \frac{1}{2} \right| \leq \text{Succ}_f^{\text{OW}}(\mathcal{I}) + \frac{q_D}{2^n}$$

Based on existence of two efficient algorithms:

- ▶ CIE: given $f(r, s_1), f(r, s_2)$ with $s_1 \neq s_2$, returns s_1, s_2 and r
- ▶ SIE: given $f(r, s)$ and r returns s

Conclusion

High-assurance cryptographic proofs

- ▶ Rigorous proofs using PL techniques (pRHL)
- ▶ Independent verification and (in principle) certified proofs

Automated generation of schemes and proofs

- ▶ public-key encryption
- ▶ zero-knowledge compilers

Directions for future work

- ▶ Foundations: logics, decision procedures
- ▶ Techniques: probabilistic encapsulation, relational invariant inference, automated reasoning about probabilities. . .
- ▶ Towards implementations
- ▶ Synthesis: revisit classical cryptography