

Computer-aided cryptographic proofs

Gilles Barthe & Yassine Lakhnech

IMDEA Software Institute, Madrid, Spain
Université Joseph Fourier & CNRS, Grenoble, France

Based on joint work with J.M. Crespo, F. Dupressoir, B. Grégoire,
C. Kunz, B. Schmidt, P.-Y. Strub, S. Zanella, J.C.B. Almeida,
M. Barbosa

Modern cryptography

1949 C. Shannon. *Communication theory of secrecy systems*.

- ▶ No practical encryption system is perfectly secure

▶ Scheme \rightarrow Attack \rightarrow Scheme \rightarrow Attack $\rightarrow \dots$

- ▶ Scheme deemed secure if no attack found for long time

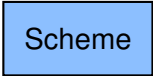
1984 S. Goldwasser and S. Micali. *Probabilistic encryption*.

- ▶ Complexity-theoretical approach
- ▶ Negligible probability to break a scheme in polynomial-time

1994 M. Bellare and P. Rogaway. *Optimal Asymmetric Encryption*.

- ▶ Upper bound the probability to break a scheme in time t

Reductionist proof



Scheme

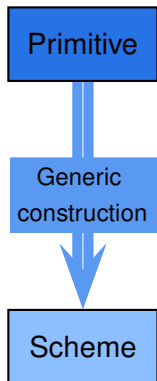
Reductionist proof



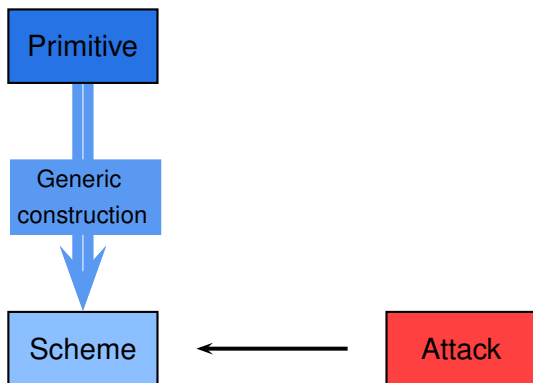
Primitive

Scheme

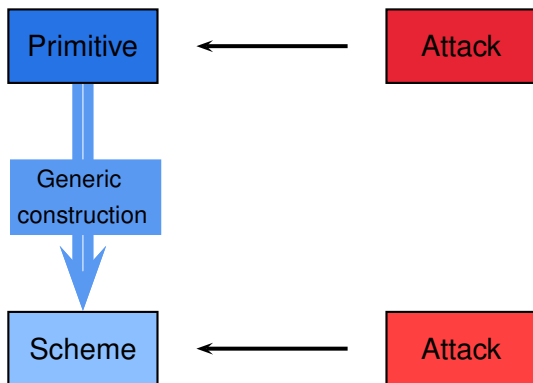
Reductionist proof



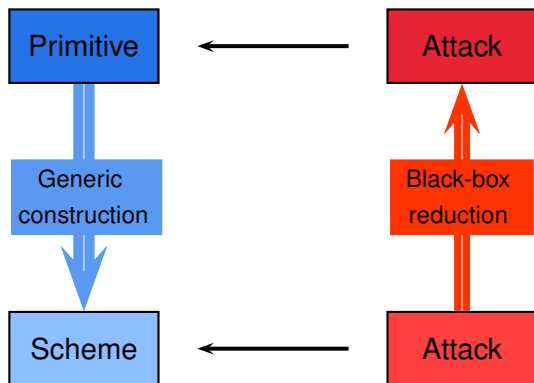
Reductionist proof



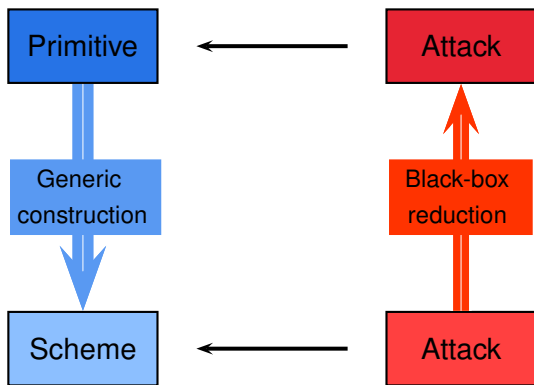
Reductionist proof



Reductionist proof



Reductionist proof



Ideally attacks have similar execution times

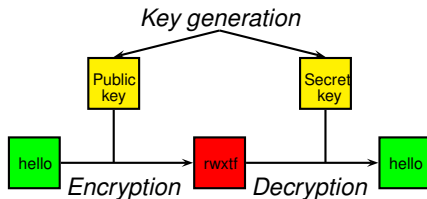
Public-key encryption

Algorithms $(\mathcal{K}, \mathcal{E}_{pk}, \mathcal{D}_{sk})$

- ▶ \mathcal{E} probabilistic
- ▶ \mathcal{D} deterministic and partial

If (sk, pk) is a valid key pair,

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)) = m$$

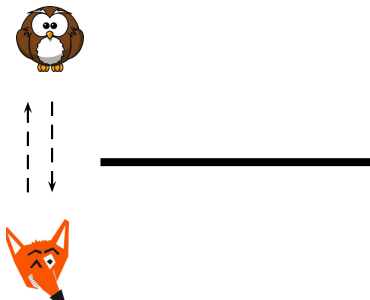


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

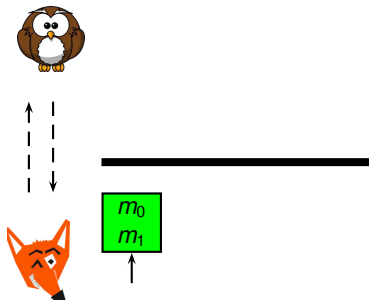


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

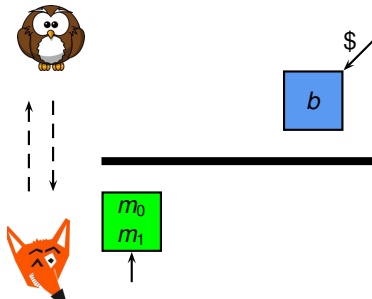


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

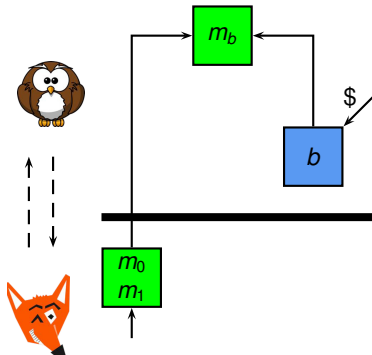


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

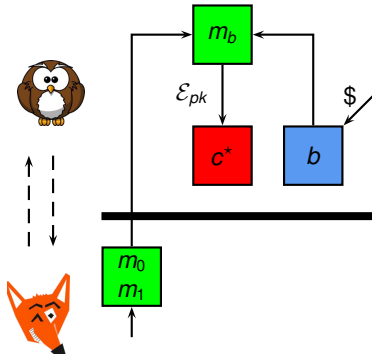


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

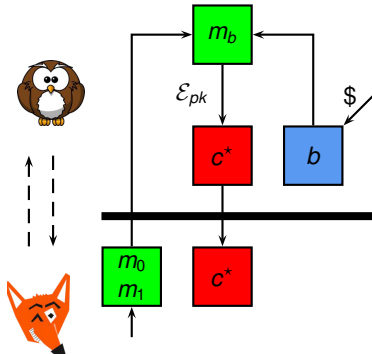


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

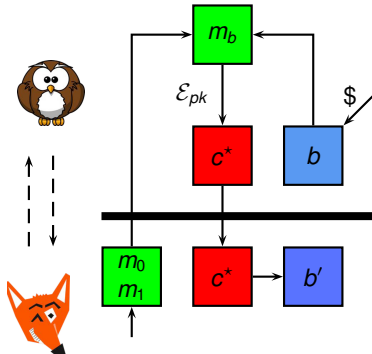


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

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

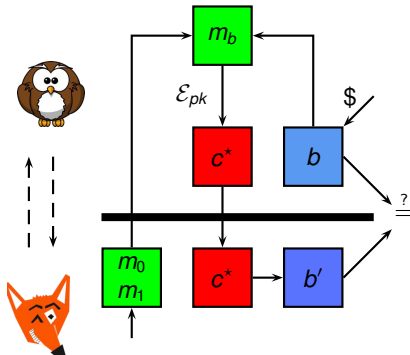


Indistinguishability against chosen-ciphertext attacks

Game IND-CCA(\mathcal{A})

$$(sk, pk) \leftarrow \mathcal{K}();$$
$$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$$
$$b \stackrel{\$}{\leftarrow} \{0, 1\};$$
$$c^* \leftarrow \mathcal{E}_{pk}(m_b);$$
$$b' \leftarrow \mathcal{A}_2(c^*);$$

```
return ( $b' = b$ )
```

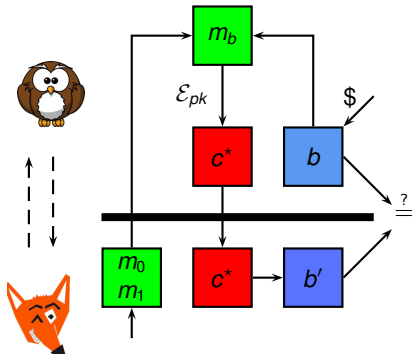


Public-key encryption

Indistinguishability against chosen-ciphertext attacks

Game $\text{IND-CCA}(\mathcal{A})$

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



$$\left| \Pr_{\text{IND-CCA}(\mathcal{A})}[b' = b] - \frac{1}{2} \right| \text{ small}$$

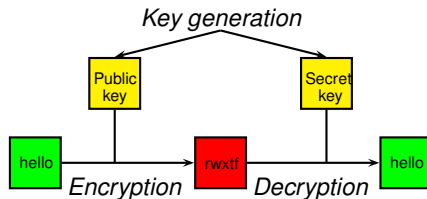
One-way trapdoor permutations

Algorithms $(\mathcal{K}, f_{pk}, f_{sk}^{-1})$

- ▶ f_{pk} and f_{sk}^{-1} deterministic

If (sk, pk) is a valid key pair,

$$f_{sk}^{-1}(f_{pk}(m)) = m$$



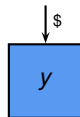
One-way trapdoor permutations

```
(sk, pk) ←  $\mathcal{K}()$ ;  
 $y \xleftarrow{\$} \{0, 1\}^n$ ;  
 $x^* \leftarrow f_{pk}(y)$ ;  
 $y' \leftarrow \mathcal{I}(x^*)$ ;  
return ( $y' = y$ )
```



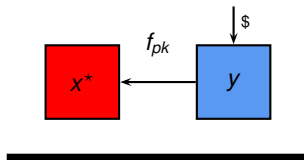
One-way trapdoor permutations

```
(sk, pk) ←  $\mathcal{K}()$ ;  
 $y \xleftarrow{\$} \{0, 1\}^n$ ;  
 $x^* \leftarrow f_{pk}(y)$ ;  
 $y' \leftarrow \mathcal{I}(x^*)$ ;  
return ( $y' = y$ )
```



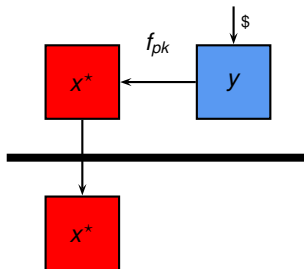
One-way trapdoor permutations

```
(sk, pk) ←  $\mathcal{K}$ ();  
 $y \xleftarrow{\$} \{0, 1\}^n$ ;  
 $x^* \leftarrow f_{pk}(y)$ ;  
 $y' \leftarrow \mathcal{I}(x^*)$ ;  
return ( $y' = y$ )
```



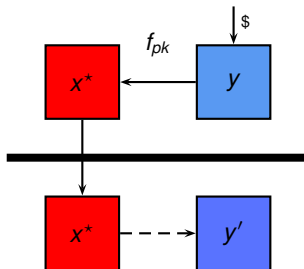
One-way trapdoor permutations

```
(sk, pk) ←  $\mathcal{K}$ ();  
 $y \xleftarrow{\$} \{0, 1\}^n$ ;  
 $x^* \leftarrow f_{pk}(y)$ ;  
 $y' \leftarrow \mathcal{I}(x^*)$ ;  
return ( $y' = y$ )
```



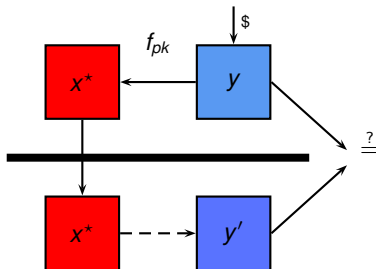
One-way trapdoor permutations

```
(sk, pk) ←  $\mathcal{K}()$ ;  
 $y \xleftarrow{\$} \{0, 1\}^n$ ;  
 $x^* \leftarrow f_{pk}(y)$ ;  
 $y' \leftarrow \mathcal{I}(x^*)$ ;  
return ( $y' = y$ )
```



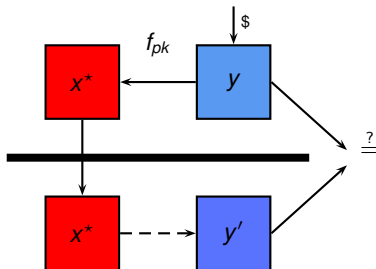
One-way trapdoor permutations

$(sk, pk) \leftarrow \mathcal{K}();$
 $y \xleftarrow{\$} \{0, 1\}^n;$
 $x^* \leftarrow f_{pk}(y);$
 $y' \leftarrow \mathcal{I}(x^*);$
return $(y' = y)$



One-way trapdoor permutations

$(sk, pk) \leftarrow \mathcal{K}();$
 $y \xleftarrow{\$} \{0, 1\}^n;$
 $x^* \leftarrow f_{pk}(y);$
 $y' \leftarrow \mathcal{I}(x^*);$
return $(y' = y)$



$\Pr_{\text{OW}(\mathcal{I})}[y' = y]$ small

Optimal Asymmetric Encryption Padding

Encryption $\mathcal{E}_{\text{OAEP}(pk)}(m) :$

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

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

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

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

Decryption $\mathcal{D}_{\text{OAEP}(sk)}(c) :$

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

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

if $([s \oplus G(r)]_{k_1} = 0^{k_1})$

then $\{m \leftarrow [s \oplus G(r)]^k;\}$

else $\{m \leftarrow \perp;\}$

return m

\oplus exclusive or \parallel concatenation $[\cdot]$ projection 0 zero bitstring

Optimal Asymmetric Encryption Padding

Encryption $\mathcal{E}_{\text{OAEP}(pk)}(m) :$

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

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

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

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

Decryption $\mathcal{D}_{\text{OAEP}(sk)}(c) :$

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

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

if $([s \oplus G(r)]_{k_1} = 0^{k_1})$

then $\{m \leftarrow [s \oplus G(r)]^{k_1};\}$

else $\{m \leftarrow \perp;\}$

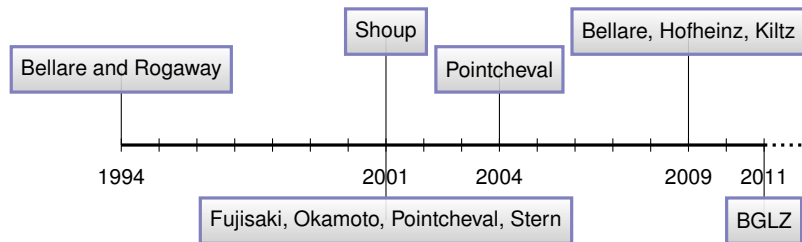
return m

For every IND-CCA adversary \mathcal{A} against $(\mathcal{K}, \mathcal{E}_{\text{OAEP}}, \mathcal{D}_{\text{OAEP}})$,
there exists a PDOW adversary \mathcal{I} against (\mathcal{K}, f, f^{-1}) st

$$|\Pr_{\text{IND-CCA}(\mathcal{A})}[b' = b] - \frac{1}{2}| \leq$$

$$\Pr_{\text{PDOW}(\mathcal{I})}[y' = y] + \frac{3q_D q_G + q_D^2 + 4q_D + q_G}{2^{k_0}} + \frac{2q_D}{2^{k_1}}$$

OAEP: Optimal Asymmetric Encryption Padding



1994 Purported proof of chosen-ciphertext security

2001 1994 proof gives weaker security; desired security holds

- for a modified scheme
- under stronger assumptions

2004 Filled gaps in 2001 proof

2009 Security definition needs to be clarified

2011 Fills 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.* Bellare and 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).* Halevi, 2005

Computer-aided cryptographic proofs

Provable security as deductive relational verification
of open probabilistic parametrized programs

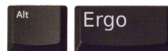
CertiCrypt (2006-2011): adhere to cryptographic methods

- ▶ same level of abstraction
- ▶ same guarantees
- ▶ same proof techniques



EasyCrypt (2009-): adhere to cryptographic practice

- ▶ automation and scalability
- ▶ support for high level steps
- ▶ accessible to cryptographers



A language for cryptographic games

$\mathcal{C} ::=$	skip	skip
	$\mathcal{V} \leftarrow \mathcal{E}$	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

- ▶ \mathcal{E} : (higher-order) expressions
 - ▶ \mathcal{D} : discrete sub-distributions
 - ▶ \mathcal{P} : procedures
- } user extensible
- . oracles: concrete procedures
 - . adversaries: constrained abstract procedures

pRHL: a relational Hoare logic for games

- ▶ Judgment

$$\models \{P\} c_1 \sim c_2 \{Q\}$$

- ▶ Validity

$$\forall m_1, m_2. (m_1, m_2) \models P \implies (\llbracket c_1 \rrbracket m_1, \llbracket c_2 \rrbracket m_2) \models Q^\sharp$$

- ▶ Proof rules

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

$$P \rightarrow e\langle 1 \rangle = e'\langle 2 \rangle$$

$$\frac{\models \{P \wedge e\langle 1 \rangle\} c_1 \sim c'_1 \{Q\} \quad \models \{P \wedge \neg e\langle 1 \rangle\} c_2 \sim c'_2 \{Q\}}{\models \{P\} \text{ if } e \text{ then } c_1 \text{ else } c_2 \sim \text{if } e' \text{ then } c'_1 \text{ else } c'_2 \{Q\}}$$

+ random samplings, procedures, adversaries. . .

- ▶ Verification condition generator

Example: Bellare and Rogaway 1993 encryption

Game IND-CPA(\mathcal{A}) :

$(sk, pk) \leftarrow \mathcal{K}(\);$

$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

$b \xleftarrow{\$} \{0, 1\};$

$c^* \leftarrow \mathcal{E}_{pk}(m_b);$

$b' \leftarrow \mathcal{A}_2(c^*);$

return $(b' = b)$

Encryption $\mathcal{E}_{pk}(m)$:

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

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

$y \leftarrow f_{pk}(r) \parallel s;$

return y

For every IND-CPA adversary \mathcal{A} , there exists an inverter \mathcal{I} st

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

Proof

Game hopping technique

Game IND CPA :

$(sk, pk) \leftarrow \mathcal{K}();$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

$b \xleftarrow{\$} \{0, 1\};$
 $c^* \leftarrow \mathcal{E}_{pk}(m_b);$
 $b' \leftarrow \mathcal{A}_2(c^*);$
return $(b' = b)$

Encryption $\mathcal{E}_{pk}(m)$:

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

Game G :

$(sk, pk) \leftarrow \mathcal{K}();$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

$b \xleftarrow{\$} \{0, 1\};$
 $c^* \leftarrow \mathcal{E}_{pk}(m_b);$
 $b' \leftarrow \mathcal{A}_2(c^*);$
return $(b' = b)$

Encryption $\mathcal{E}_{pk}(m)$:

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

Game G' :

$(sk, pk) \leftarrow \mathcal{K}();$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

$b \xleftarrow{\$} \{0, 1\};$
 $c^* \leftarrow \mathcal{E}_{pk}(m_b);$
 $b' \leftarrow \mathcal{A}_2(c^*);$
return $(b' = b)$

Encryption $\mathcal{E}_{pk}(m)$:

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

Game OW :

$(sk, pk) \leftarrow \mathcal{K}();$

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

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

$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $c^* \leftarrow x \parallel s;$
 $b' \leftarrow \mathcal{A}_2(c^*);$
 $y' \leftarrow [z \in L_H^A \mid f_{pk}(z) = x];$
return y'

1. For each hop

- ▶ prove validity of pRHL judgment
- ▶ derive probability claim(s)

2. Obtain security bound by combining claims

3. Check execution time of constructed adversary

Conditional equivalence

$\mathcal{E}_{pk}(m) :$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $h \leftarrow H(r);$
 $s \leftarrow h \oplus m;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c



$\mathcal{E}_{pk}(m) :$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $h \xleftarrow{\$} \{0, 1\}^k;$
 $s \leftarrow h \oplus m;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c

$$\models \{ \text{true} \} \text{ IND-CPA} \sim \mathbf{G} \left\{ (\neg r \in L_H^A) \langle 2 \rangle \rightarrow \equiv \right\}$$

$$|\Pr_{\text{IND-CPA}}[b' = b] - \Pr_{\mathbf{G}}[b' = b]| \leq \Pr_{\mathbf{G}}[r \in L_H^A]$$

Equivalence

$\mathcal{E}_{pk}(m) :$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $h \xleftarrow{\$} \{0, 1\}^k;$
 $s \leftarrow h \oplus m;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c



$\mathcal{E}_{pk}(m) :$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $h \leftarrow s \oplus m;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c

$$\models \{ \text{true} \} \mathbf{G} \sim \mathbf{G}' \{ \equiv \}$$

$$\Pr_{\mathbf{G}}[r \in L_H^A] = \Pr_{\mathbf{G}'}[r \in L_H^A] \quad \Pr_{\mathbf{G}}[b' = b] = \Pr_{\mathbf{G}'}[b' = b] = \frac{1}{2}$$

Equivalence

$\mathcal{E}_{pk}(m) :$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $h \xleftarrow{\$} \{0, 1\}^k;$
 $s \leftarrow h \oplus m;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c



$\mathcal{E}_{pk}(m) :$
 $r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $h \leftarrow s \oplus m;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c

$$\models \{ \text{true} \} \mathbf{G} \sim \mathbf{G}' \{ \equiv \}$$

$$|\Pr_{\text{IND-CPA}}[b' = b] - \frac{1}{2}| \leq \Pr_{\mathbf{G}'}[r \in L_H^A]$$

Reduction

Game IND CPA :

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

Encryption $\mathcal{E}_{pk}(m)$:

$r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c

Game OW :

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

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

$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $c^* \leftarrow x \parallel s;$
 $b' \leftarrow \mathcal{A}_2(c^*);$
 $y' \leftarrow [z \in L_H^A \mid f_{pk}(z) = x];$
return y'

$$\models \{ \text{true} \} \mathbf{G}' \sim \text{OW} \{ (r \in L_H^A) \langle 1 \rangle \rightarrow (y' = y) \langle 2 \rangle \}$$

$$\Pr_{\mathbf{G}'} [r \in L_H^A] \leq \Pr_{\text{OW}(\mathcal{I})} [y' = y]$$

Reduction

Game IND CPA :

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

Encryption $\mathcal{E}_{pk}(m)$:

$r \xleftarrow{\$} \{0, 1\}^\ell;$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $c \leftarrow f_{pk}(r) \parallel s;$
return c

Game OW :

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

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

$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
 $b \xleftarrow{\$} \{0, 1\};$
 $s \xleftarrow{\$} \{0, 1\}^k;$
 $c^* \leftarrow x \parallel s;$
 $b' \leftarrow \mathcal{A}_2(c^*);$
 $y' \leftarrow [z \in L_H^A \mid f_{pk}(z) = x];$
return y'

$$\models \left\{ \text{true} \right\} \mathbf{G}' \sim \text{OW} \left\{ (r \in L_H^A) \langle 1 \rangle \rightarrow (y' = y) \langle 2 \rangle \right\}$$

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

Case studies

- ▶ Public-key encryption
- ▶ Signatures
- ▶ Hash function designs
- ▶ Block ciphers
- ▶ Zero-knowledge protocols
- ▶ Differential privacy
- ▶ (Computational) differential privacy
- ▶ Authenticated key exchange protocols

Compiler

Approximate
pRHL

Compositionality

Current directions

- ▶ Compositional proofs

One of the most vexing basic problems in computer security is the problem of secure composition. [...] We predict that secure composition will receive the increasing attention that it deserves. Boneh and Mitchell, 2012

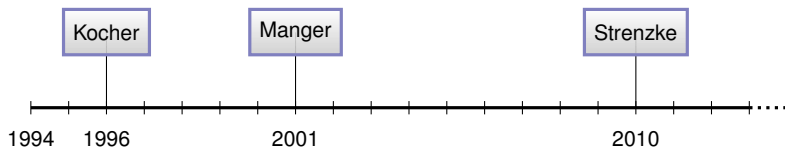
- ▶ Real-world cryptography

Real-world crypto is breakable; is in fact being broken; is one of many ongoing disaster areas in security. Bernstein, 2013

- ▶ Synthesis of secure cryptographic schemes

Do cryptosystems reflect [...] the situations that are being catered for? Or are they accidents of history and personal background that may be obscuring fruitful developments?
After Landin, 1966

Real-world security of RSA-OAEP



- ▶ plaintext is variable-sized: careless parsing leads to padding oracle (Manger);
- ▶ RSA is permutation only on strict subset of the domain considered ($[0..2^k]$): careless error handling leads to timing attacks;
- ▶ PKCS#1 prescribes some error messaging, rarely considered in existing proofs.

Proving “real-world” security of RSA-OAEP: outline

- ▶ Adapt the OAEP security proof to a *low-level model* of the RSA PKCS#1 v2.1 standard
- ▶ Consider an extended adversary model:
 - Control and access to low-level encodings of inputs and outputs,
 - Oracles also return a *leakage trace* meant to model side-channels
- ▶ Extend and leverage CompCert’s *semantic preservation* results to obtain a low-level, leakage-aware security result on the compiled ASM code

A Low-Level Model...

Decryption $\mathcal{D}_{\text{OAEP}(sk)}(c)$:

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

Decryption $\mathcal{D}_{\text{OAEP}(sk)}(res, c)$:

if $(c \in \text{MsgSpace}(sk))$
 $\{ (b0, s, t) \leftarrow f_{sk}^{-1}(c);$
 $h \leftarrow H(s); i \leftarrow 0;$
 while $(i < hLen + 1)$
 $\{ s[i] \leftarrow t[i] \oplus h[i]; i \leftarrow i + 1; \}$
 $g \leftarrow G(r); i \leftarrow 0;$
 while $(i < dbLen)$
 $\{ p[i] \leftarrow s[i] \oplus g[i]; i \leftarrow i + 1; \}$
 $l \leftarrow \text{payload_length}(p);$
 if $(b0 = 0^8 \wedge [p]_l^{hLen} = 0..01 \wedge$
 $[p]_{hLen} = LHash)$
 then
 $\{rc \leftarrow \text{Success};$
 $\text{memcpy}(res, 0, p, dbLen - l, l);\}$
 else $\{rc \leftarrow \text{DecryptionError}; \}$
 else $\{rc \leftarrow \text{CiphertextTooLong}; \}$
 return $rc;$

...with Leakage

- ▶ Focus on Program Counter Security: adversary is given the list of program points traversed while executing the oracle
- ▶ Leakage due to the computation of the permutation is kept abstract
- ▶ Axioms formalize our leakage assumptions on their implementation
- ▶ Security assumption (PDOW) is slightly adapted to deal with abstract leakage

CompCert and PC Security

- ▶ CompCert guarantees that traces of events are preserved by compilation;
- ▶ Events are calls to the environment (system calls, *random sampling, hashing, key generation*), and branching decisions (each basic block starts with an event)
- ▶ Extend the CompCert run-time with a formally specified, trusted Multi-Precision Integer Arithmetic library, assumed to satisfy “good enough” leakage resistance
- ▶ Syntactic check on final ASM code guarantees that the final annotations are sufficient.

Perspectives on real-world security



Still a model.

- ▶ Adversary and execution models are still somewhat idealized
- ▶ Not clear how to prove memory obliviousness
- ▶ Consider more active side-channels (fault injection ...)
- ▶ Prove security in a virtualized environment

The next 700 cryptosystems: ZooCrypt

- ▶ generate all schemes up to user-defined constraints
- ▶ automatically prove security, or existence of an attack, by combining the two views of cryptography

Using symbolic methods for

- ▶ Finding attacks
- ▶ Synthesis of decryption algorithm
- ▶ In proof system for
 - Computing symbolic entropy
 - Finding symbolic reduction

Minimality in cryptography

- ▶ OAEP (1994):

$$f((m \| 0) \oplus G(r) \| r \oplus H((m \| 0) \oplus G(r)))$$

not that Optimal; needs redundancy

- ▶ SAEP (2001):

$$f(r \| (m \| 0) \oplus G(r))$$

tighter reduction; needs redundancy

- ▶ ZAEP:

$$f(r \| m \oplus G(r))$$

tighter reduction, bit-optimal, redundancy-free

Conclusion

Cryptography is

- ▶ a thriving research area at the crossroads of many fields
 - ▶ a great source of challenging problems
 - ▶ an exciting opportunity to apply PL and PV techniques
-
- ▶ Visit <http://www.easycrypt.info>
 - ▶ Download EasyCrypt
 - ▶ Attend first School and Workshop, July 16-19, 2013