

Computer-aided cryptographic proofs

Gilles Barthe
MPI-SP, Germany
IMDEA Software Institute, Spain

Modern cryptography

Shannon '49

- Mathematical proof of security
- Perfect secrecy is impossible

Diffie & Hellman '76

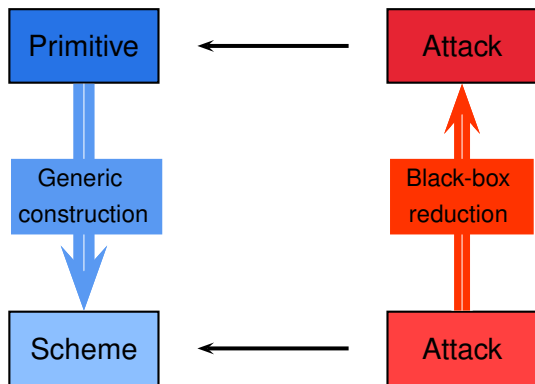
- Computational security
 - Asymptotic guarantees
- PPT adversary has negligible advantage

Goldwasser & Micali '82
Yao '82

Bellare & Rogaway '94

- Concrete bounds
- Adversary advantage to win in time t is $\leq p$

Reductionist proof



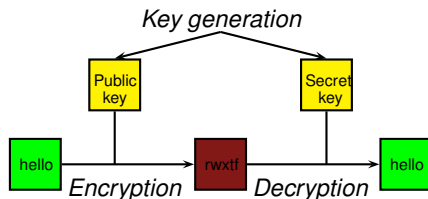
Public-key encryption

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,

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



Indistinguishability

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 $(b' = b)$

Indistinguishability

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

Indistinguishability

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

Indistinguishability

Game $\text{INDCPA}(\mathcal{A})$

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

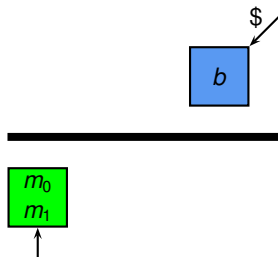
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

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

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

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

return $(b' = b)$



Indistinguishability

Game IND CPA(\mathcal{A})

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

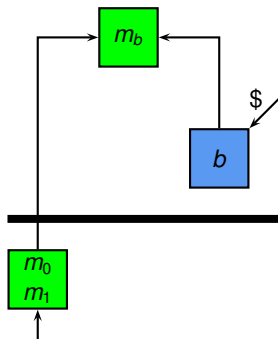
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$;

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

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

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

return $(b' = b)$



Indistinguishability

Game $\text{INDCPA}(\mathcal{A})$

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

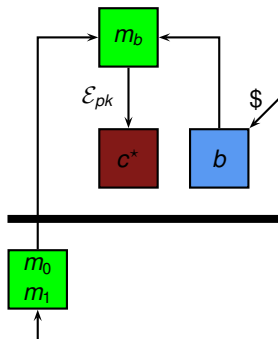
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

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

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

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

return $(b' = b)$



Indistinguishability

Game IND CPA(\mathcal{A})

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

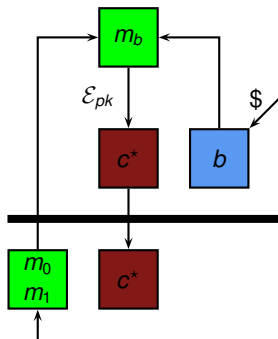
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

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

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

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

return $(b' = b)$



Indistinguishability

Game $\text{INDCPA}(\mathcal{A})$

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

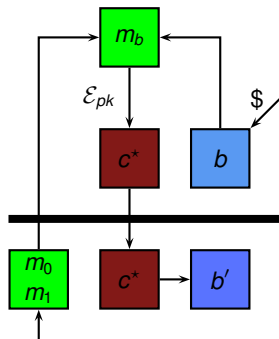
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$

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

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

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

return $(b' = b)$



Indistinguishability

Game IND CPA(\mathcal{A})

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

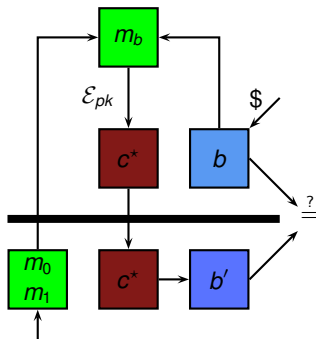
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$;

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

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

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

return $(b' = b)$



Indistinguishability

Game $\text{INDCPA}(\mathcal{A})$

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

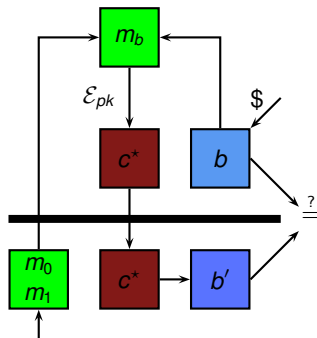
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$;

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

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

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

return $(b' = b)$



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

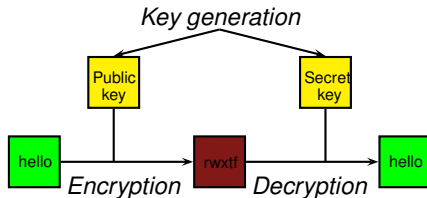
One-way trapdoor permutations

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

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

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

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



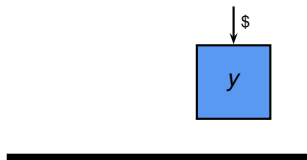
One-way trapdoor permutations

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



One-way trapdoor permutations

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



One-way trapdoor permutations

Game $\text{OW}(\mathcal{I})$

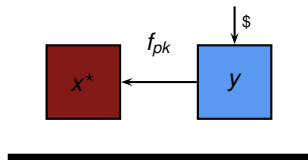
$(sk, pk) \leftarrow \mathcal{K}()$;

$y \xleftarrow{\$} \{0, 1\}^n$;

$x^* \leftarrow f_{pk}(y)$;

$y' \leftarrow \mathcal{I}(x^*)$;

return $(y' = y)$



One-way trapdoor permutations

Game $\text{OW}(\mathcal{I})$

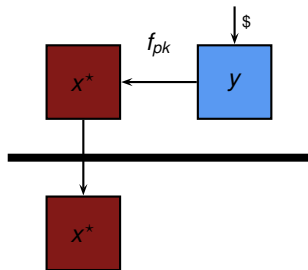
$(sk, pk) \leftarrow \mathcal{K}()$;

$y \xleftarrow{\$} \{0, 1\}^n$;

$x^* \leftarrow f_{pk}(y)$;

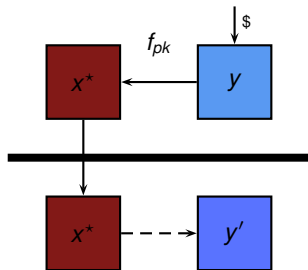
$y' \leftarrow \mathcal{I}(x^*)$;

return $(y' = y)$



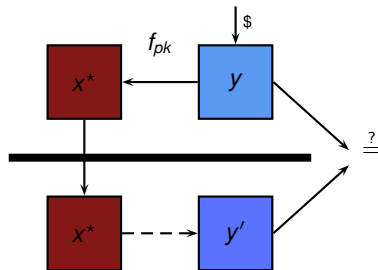
One-way trapdoor permutations

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



One-way trapdoor permutations

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



One-way trapdoor permutations

Game $\text{OW}(\mathcal{I})$

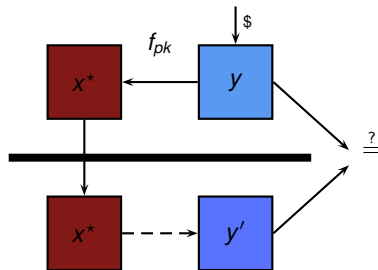
$(sk, pk) \leftarrow \mathcal{K}()$;

$y \xleftarrow{\$} \{0, 1\}^n$;

$x^* \leftarrow f_{pk}(y)$;

$y' \leftarrow \mathcal{I}(x^*)$;

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

Oracle $H(x)$:

if $x \notin L$ then

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

$L \leftarrow (x, r) :: L$;

return $L[x]$;

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

OAEP: provable security

Game INDCCA(\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 $(b' = b)$

Game SPDOW(\mathcal{I})

$(sk, pk) \leftarrow \mathcal{K}()$;
 $y \xleftarrow{\$} \{0, 1\}^{k_2}$; $z \xleftarrow{\$} \{0, 1\}^{k_3}$;
 $x^* \leftarrow f_{pk}(y \| z)$;
 $Y' \leftarrow \mathcal{I}(x^*)$;
return $(y \in Y')$

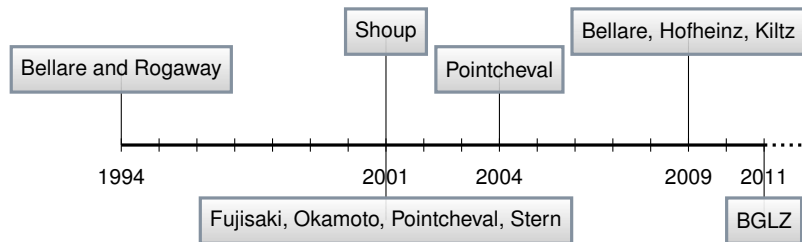
FOR ALL INDCCA adversary \mathcal{A} against $(\mathcal{K}, \mathcal{E}_{OAEP}, \mathcal{D}_{OAEP})$,
THERE EXISTS a SPDOW adversary \mathcal{I} against (\mathcal{K}, f, f^{-1}) st

$$\left| \Pr_{\text{INDCCA}(\mathcal{A})}[b' = b] - \frac{1}{2} \right| \leq \Pr_{\text{SPDOW}(\mathcal{I})}[y \in Y'] + \frac{3q_D q_G + q_D^2 + 4q_D + q_G}{2^{k_0}} + \frac{2q_D}{2^{k_1}}$$

and

$$t_{\mathcal{I}} \leq t_{\mathcal{A}} + q_D q_G q_H T_f$$

OAEP: provable security



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

Example: Bellare and Rogaway 1993 encryption

Game $\text{INDCPA}(\mathcal{A})$:	Encryption $\mathcal{E}_{pk}(m)$:
$(sk, pk) \leftarrow \mathcal{K}()$;	$r \xleftarrow{\$} \{0, 1\}^\ell$;
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$;	$s \leftarrow H(r) \oplus m$;
$b \xleftarrow{\$} \{0, 1\}$;	$y \leftarrow f_{pk}(r) \parallel s$;
$c^* \leftarrow \mathcal{E}_{pk}(m_b)$;	return y
$b' \leftarrow \mathcal{A}_2(c^*)$;	
return $(b' = b)$	

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

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

Proof

Game hopping technique

Game INDCPA :

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

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

1. For each hop
 - ▶ prove validity of pRHL judgment
 - ▶ derive probability claims
 - ▶ (possibly) resolve some probability expressions using pHL
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

$\models \{T\} \text{ IND CPA} \sim \mathbf{G} \{(\neg r \in L_A) \langle 2 \rangle \rightarrow =_{b,b'}\}$

$$\left| \Pr_{\text{IND CPA}} [b' = b] - \Pr_{\mathbf{G}} [b' = b] \right| \leq \Pr_{\mathbf{G}} [r \in L_A]$$

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  $c$ 
```

$$\models \{T\} \mathbf{G} \sim \mathbf{G}' \{=_{b,b',r,\mathcal{A}}\}$$

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

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  $c$ 
```

$$\models \{T\} \mathbf{G} \sim \mathbf{G}' \{=_{b,b',r,\mathcal{A}}\}$$

$$|\Pr_{\text{INDCPA}}[b' = b] - \frac{1}{2}| \leq \Pr_{\mathbf{G}'}[r \in L_{\mathcal{A}}]$$

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 $\mathcal{I}(x)$:

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

$$\models \{T\} \mathbf{G}' \sim \text{OW} \{(r \in L_{\mathcal{A}})\langle 1 \rangle \rightarrow (y' = y)\langle 2 \rangle\}$$

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

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 $\mathcal{I}(x)$:

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

$$\models \{T\} \mathbf{G}' \sim \text{OW} \{(r \in L_{\mathcal{A}}) \langle 1 \rangle \rightarrow (y' = y) \langle 2 \rangle\}$$

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

EasyCrypt

Domain-specific proof assistant

- ▶ proof goals tailored to reductionist proofs
- ▶ proof tools support common proof techniques (bridging steps, failure events, hybrid arguments, eager sampling. . .)

Control and automation from state-of-art verification

- ▶ interactive proof engine and mathematical libraries (a la Coq/ssreflect)
- ▶ back-end to SMT solvers

Many case studies:

- ▶ Encryption, signatures, key exchange, zero-knowledge, multi-party and verifiable computation, SHA3, voting, KMS

Verified implementations

- ▶ **FOR EVERY** adversary that breaks assembly code,
- ▶ **IF** assembly code is safe and leakage resistant,
- ▶ **AND** assembly code correctly implements algorithm,
- ▶ **THERE EXISTS** an adversary that breaks the algorithm

Jasmin — high-assurance cryptography

- ▶ Assembly in the head
- ▶ Verified compiler
- ▶ Relational verification of
 - provable security
 - cryptographic constant-time
 - functional correctness
- ▶ High-speed: faster than record breaking (unverified) code

Application: some TLS1.3 core components, SHA3

Maskverif and Maskcomp

- ▶ Program is secure at order t iff the joint distribution for a set of observations of size t is independent from secrets
- ▶ Challenges: combinatorial explosion and composition

MaskVerif

- ▶ Check probabilistic non-interference for large sets
- ▶ Software and hardware implementations

MaskComp

- ▶ Strong non-interference
- ▶ Information flow with cardinality constraints
- ▶ Generates code at arbitrary orders

Conclusions

- ▶ Many program properties are relational
- ▶ Relational verification of probabilistic programs by coupling
- ▶ Applications to cryptography and differential privacy