

Encryption Schemes & General Assumptions

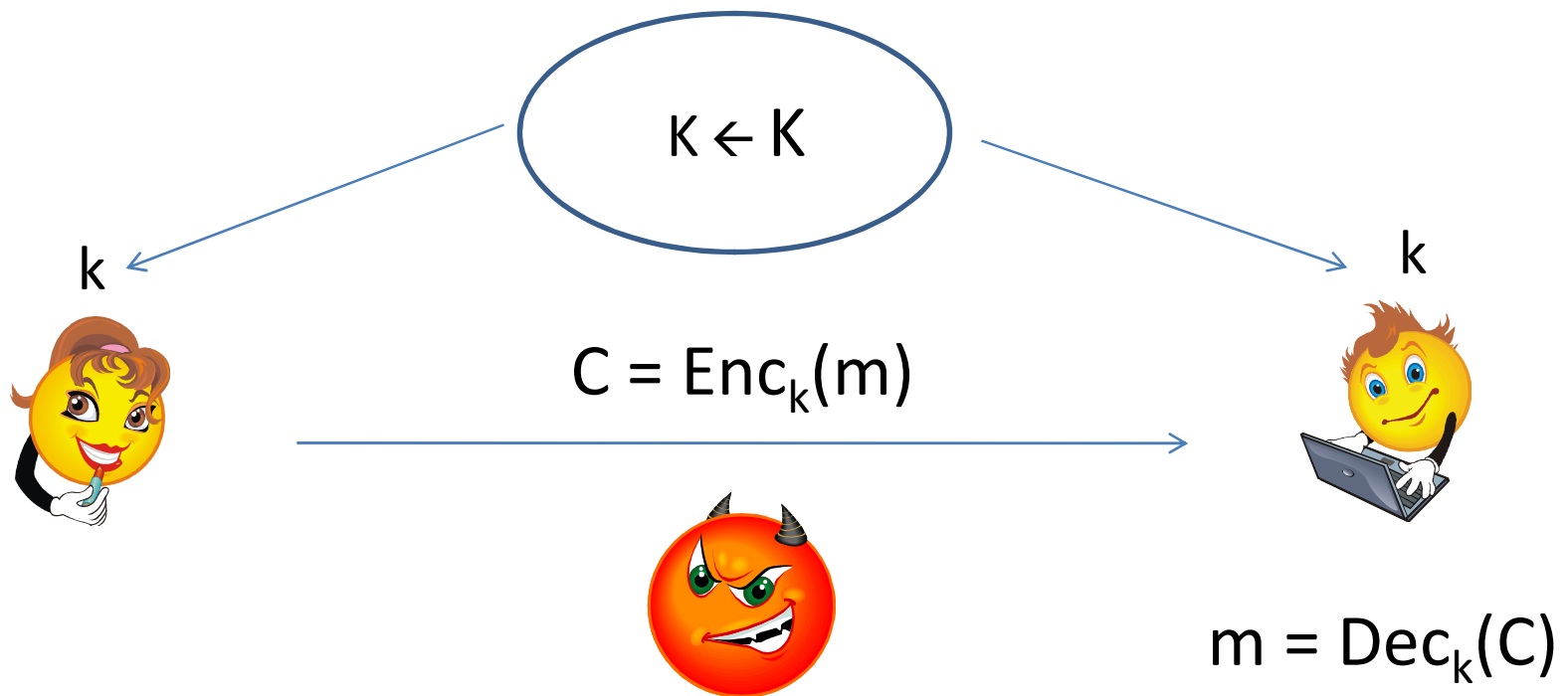
Payman Mohassel

University of Calgary

Encryption Schemes

- Private-key encryption
- Public-key encryption
- Basing cryptography on minimal assumptions
 - Gives insight to hardness of a primitive

Private-key Encryption



$$SE = (K, \text{Enc}, \text{Dec})$$

Goal of Adversary

- Tries to recover the key?
- Tries to recover the message?
- Tries to learn part of the message?
- Any partial info about the message?
 - Adversary wins if learns $f(m)$ for any f

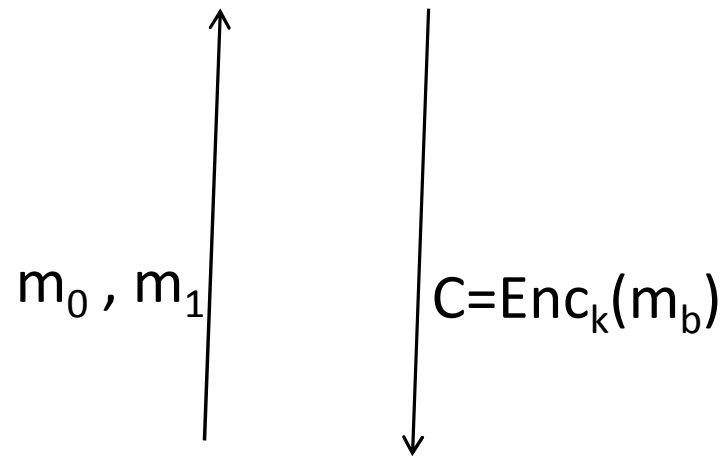
Semantic Security

- For any PPT adversary A
 - And any poly-time computable functions f, h
 - There is a PPT algorithm A'
- $p_1 = \Pr[A(\text{Enc}_k(m), h(m)) \rightarrow f(m)]$
- $p_2 = \Pr[A'(h(m)) \rightarrow f(m)]$
- $|p_1 - p_2|$ is negligible
- Captures our intuition for confidentiality

Indistinguishability

$k \leftarrow K(1^n) ; b \leftarrow \{0,1\}$

Challenger



PPT Adversary A

$b' \leftarrow A(C)$

$|\Pr[b' = b] - \frac{1}{2}|$ is negligible

Equivalent to Semantic Security

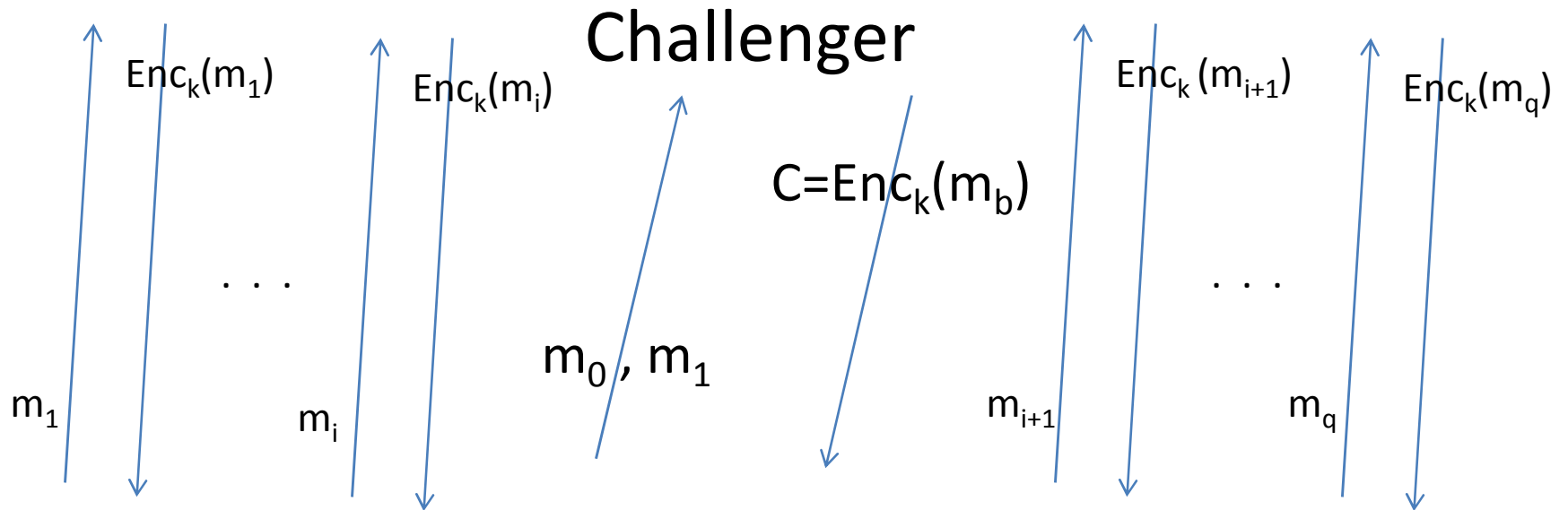
- Introduced
 - Goldwasser and Micali
- Proved equivalent
 - Goldwasser and Micali, 1984
 - Micali, Rackoff, Sloan, 1986
- IND definition is easier to work with

What does adversary see?

- Encryption of many messages?
- Choose the messages too?
 - Chosen plaintext attacks (CPA)
- Decryption of many ciphertexts?
 - Chosen ciphertext attacks (CCA)

IND-CPA

$k \leftarrow K(1^k); b \leftarrow \{0,1\}$



PPT Adversary A

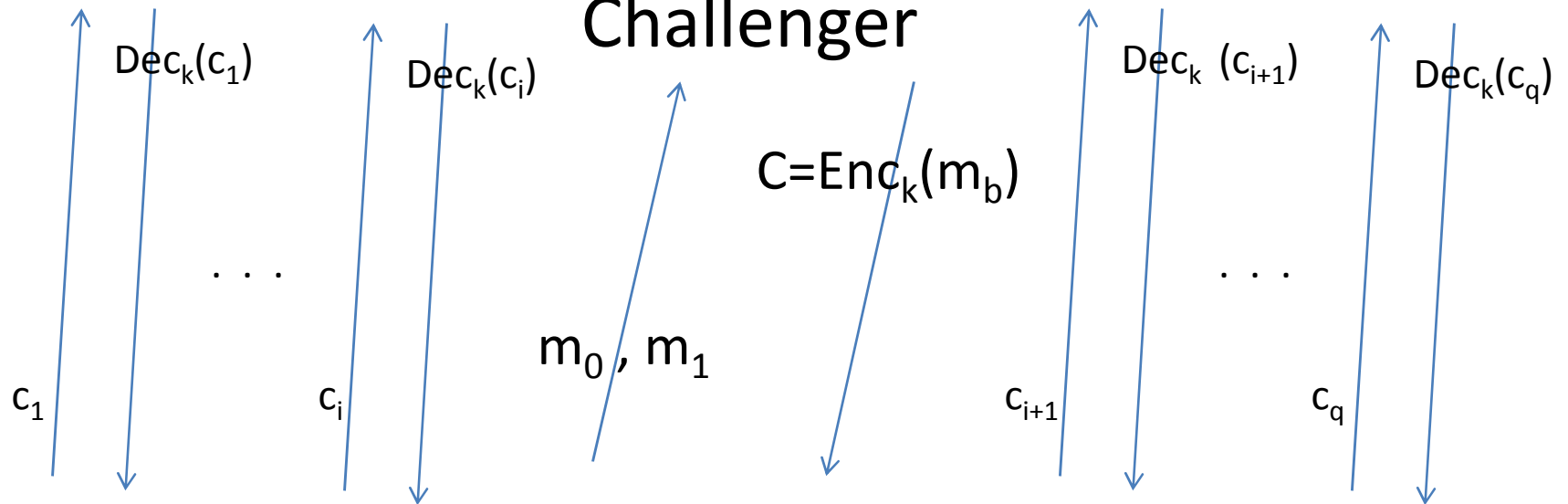
$b' \leftarrow A(C)$

$|\Pr[b' = b] - \frac{1}{2}|$ is negligible

IND-CCA

$k \leftarrow K(1^n) ; b \leftarrow \{0,1\}$

Challenger

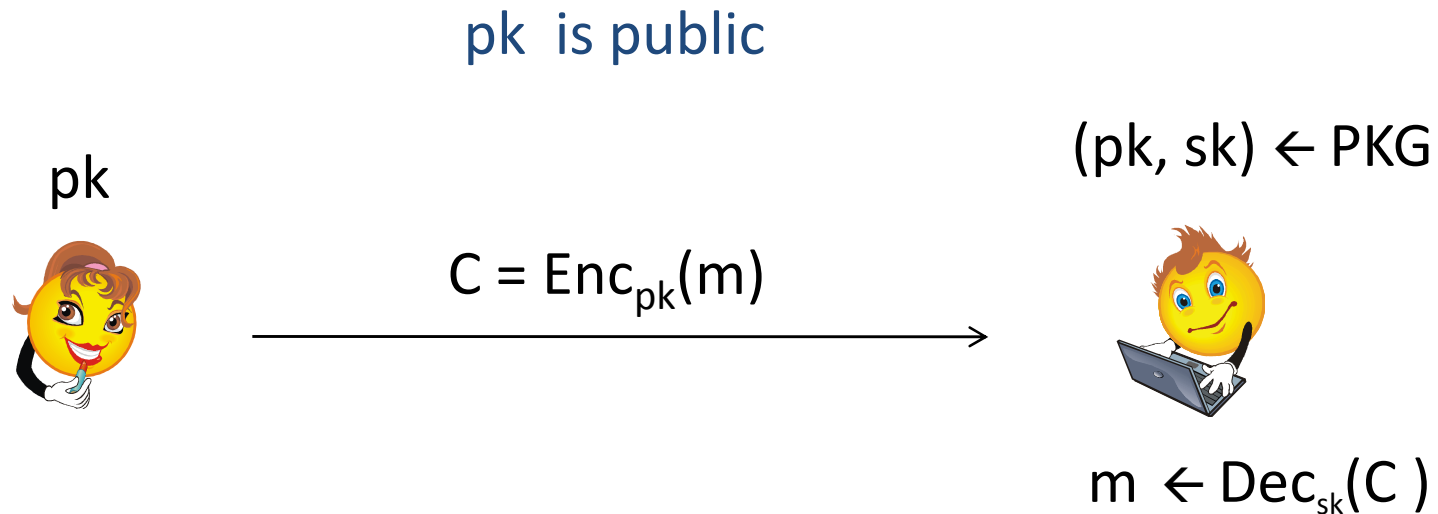


PPT Adversary A

$b' \leftarrow A(C)$

$|\Pr[b' = b] - \frac{1}{2}|$ is negligible

Public-Key Encryption



$\text{PKE} = (\text{PKG}, \text{Enc}, \text{Dec})$

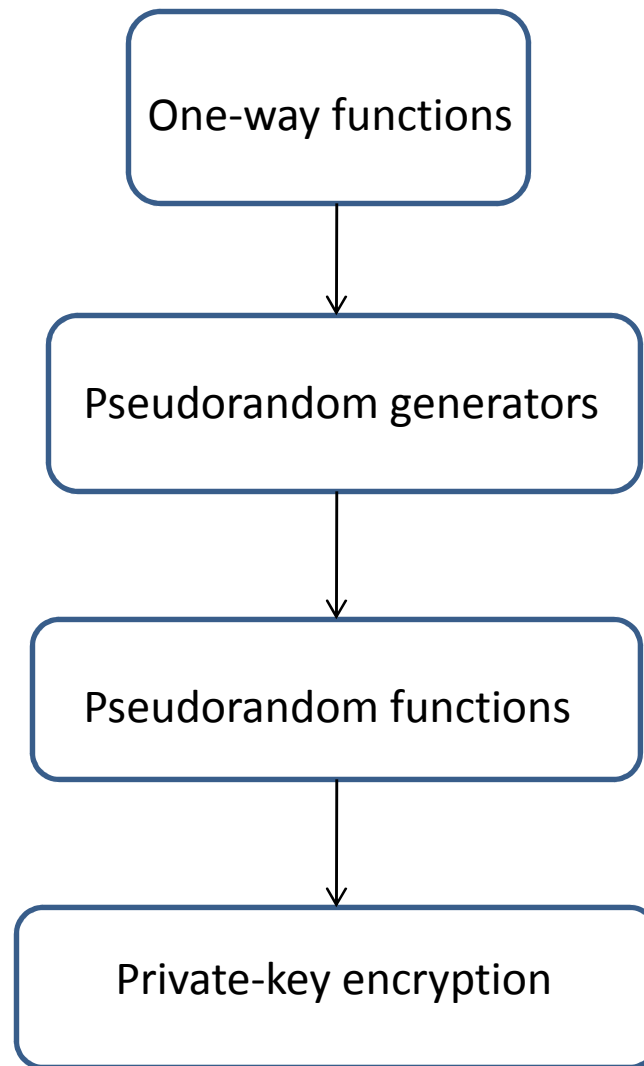
Similar Definitions

- IND-CPA, IND-CCA
- No need for encryption queries
- Adversary can encrypt on his own!

Simpler Primitives

- One-way functions (OWF)
 - Easy to compute, hard to invert
- Pseudorandom generators (PRG)
 - Given a random seed, outputs a longer (pseudo)random string
- Pseudorandom functions (PRF)
 - A keyed function
 - Behaves like a random function (on random keys)

Private-Key Encryption



One-way Functions

- Suggested by
 - Diffie and Hellman, 1976
- Formalized by
 - Yao, 1982

Security Game (OWF)

- $f: \{0,1\}^n \rightarrow \{0,1\}^{n'}$
- $x \leftarrow \{0,1\}^n; y = f(x)$
- Send challenge “ y ” to adversary A
 - $x' \leftarrow A$
- A wins if $f(x') = f(x)$
- f is one-way if no PPT adversary A wins with non-negl. probability

One-Way Functions

- You can always invert in exponential time
 - Enumerate all inputs
 - It is inherently a computational assumption
- Hard on average
 - Not just worst case!
 - NP-hardness is not sufficient
- **OWFs exist $\rightarrow P \neq NP$**
 - Their existence is a conjecture

Candidate OWFs

- Factoring
 - Let p, q be $n/2$ -bit primes
 - $f(p, q) = pq$
- Discrete log
 - A cyclic group G of order q
 - g is a generator
 - $f_{g, q}(x) = g^x$
 - Elliptic curves, etc.
- RSA problem
 - $f_{N, e}(x) = x^e \bmod N$ where $N = pq$
 - p, q are $n/2$ -bit primes

One-way Permutations

- If
 - $f: \{0,1\}^n \rightarrow \{0,1\}^n$
 - f is one-to-one
 - f is one-way
- Then f is a one-way permutation (OWP)
- RSA, Discrete-log

How can we use it?

- Use it to encrypt?
 - $f(m)$?
- It needs a bit more work

Hardcore Bits

- hc is a hardcore predicate for OWF f
 - can be computed efficiently
 - For every PPT adversary A
 - Given $f(x)$
 - $hc(x)$ looks uniform
 - $\Pr[b = hc(x) | x \leftarrow \{0,1\}^n, b \leftarrow A(f(x))]$ - $\frac{1}{2}$ is neglig.
- Probability is over the random choices of x , and A 's random coins
- Blum and Micali, 1982

Do all OWFs have a hc?

- We don't know
 - It is an open problem
- What do we know?
- Given a OWF f , we can construct a related OWF g with a hardcore predicate

Goldreich-Levin (1989)

- Given OWF f
- Let $g(x,r) = f(x) || r$ where $|x|=|r|=n$
- $hc(x,r) = \bigoplus r_i x_i$ ($i = 1$ to n)
 - XOR of a random subset of bits of x
- $hc(x,r)$ is a hardcore bit of g

Pseudorandom Generators

- $G: \{0,1\}^n \rightarrow \{0,1\}^{n'}, n' > n$
- For all PPT distinguishers D
 - $s \leftarrow \{0,1\}^n ; s' \leftarrow \{0,1\}^{n'}$
 - $\Pr[D(G(s)) \rightarrow 1] - \Pr[D(s') \rightarrow 1]$ is neglig.
 - $\Pr[D(G(U_n)) \rightarrow 1] - \Pr[D(U_{n'}) \rightarrow 1]$ is neglig.
- Blum-Micali and Yao, 1982

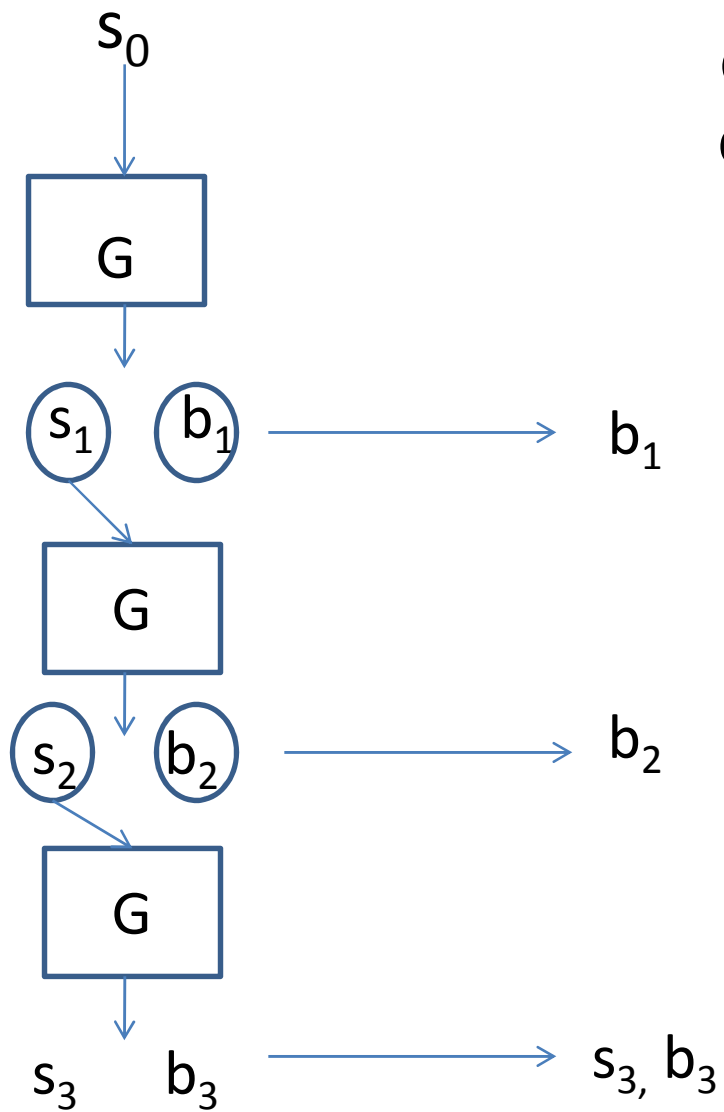
Expansion Factor 1

- Let $G(s) = f(s) \parallel hc(s)$
- G is a PRG with expansion factor 1
- OWP \rightarrow PRGs
 - Yao, 1982
- OWF \rightarrow PRGs
 - Hastad, Impagliazzo, Levin, Luby, 1999

Increasing the Expansion Factor

- Given a PRG $G: \{0,1\}^n \rightarrow \{0,1\}^{n+1}$
- We can construct $G': \{0,1\}^n \rightarrow \{0,1\}^{p(n)}$
 - For any polynomial $p(n)$

Construction



$$G: \{0,1\}^n \rightarrow \{0,1\}^{n+1}$$
$$G': \{0,1\}^n \rightarrow \{0,1\}^{n+3}$$

Pseudorandom Functions

- Goldreich, Goldwasser, Micali, 1984
 - Definition and construction
 - Private-key encryption based on PRFs

Pseudorandom Functions

- $F: K \times D \rightarrow R$
- $K = \{0,1\}^k$; $D = \{0,1\}^n$; $R = \{0,1\}^m$

Real Function

- $k \leftarrow K$
- On query x :
 - return $F_k(x)$

Rand Function

- Create Table T
- On query x :
 - if $T[x]$ empty
 - $T[x] \leftarrow R$
 - return $T[x]$

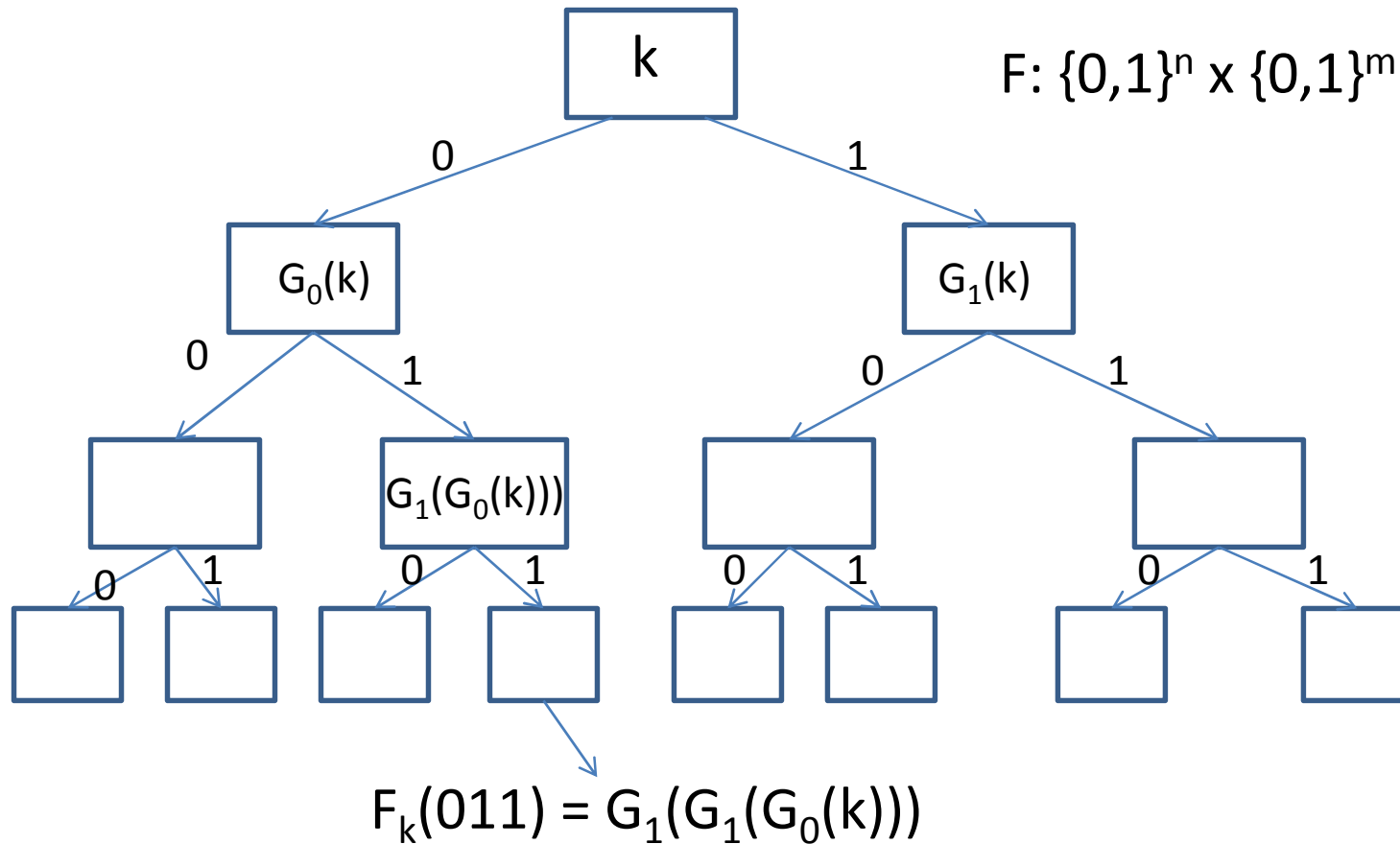
$\Pr[A^{\text{real-func}} \rightarrow 1] - \Pr[A^{\text{rand-func}} \rightarrow 1]$ is negl.

PRFs from PRGs

$$G(s) = G_0(s)G_1(s)$$

$$G: \{0,1\}^n \rightarrow \{0,1\}^{2n}$$

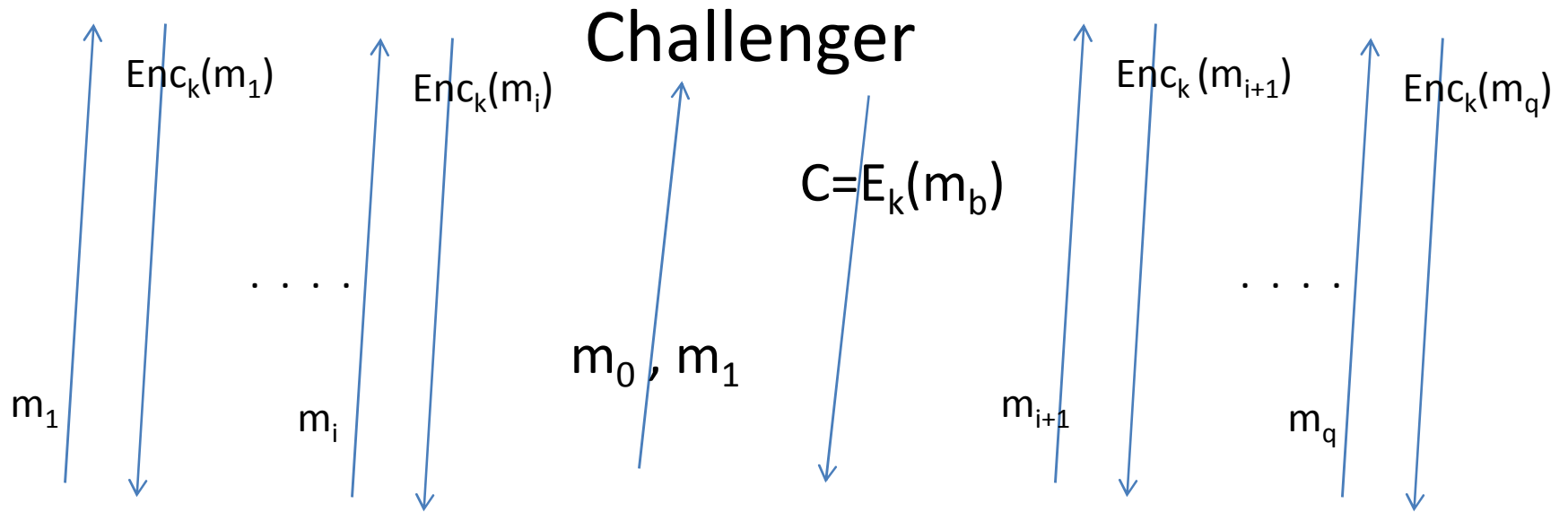
$$F: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$$



If G is a length-doubling PRG, F is a pseudorandom function

IND-CPA

$k \leftarrow \text{KG} ; b \leftarrow \{0,1\}$



PPT Adversary A

$b' \leftarrow A(C)$

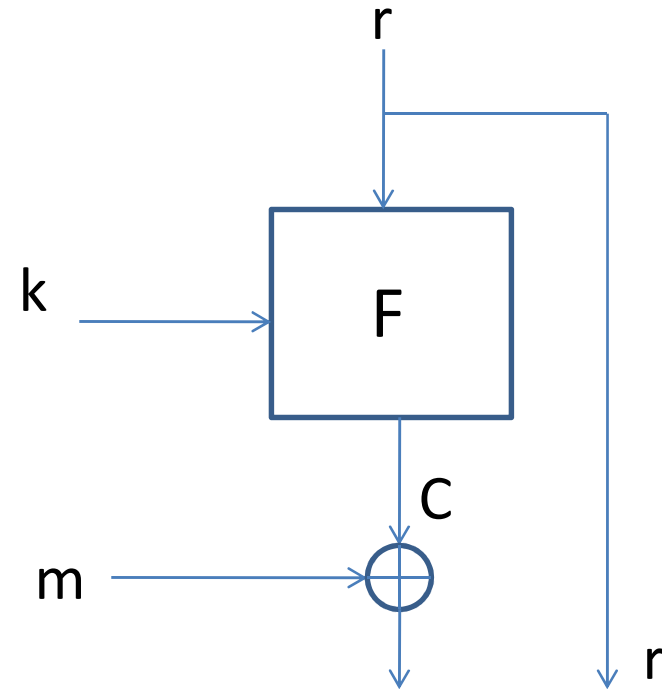
$|\Pr[b' = b] - \frac{1}{2}|$ is negligible

Encryption from PRFs

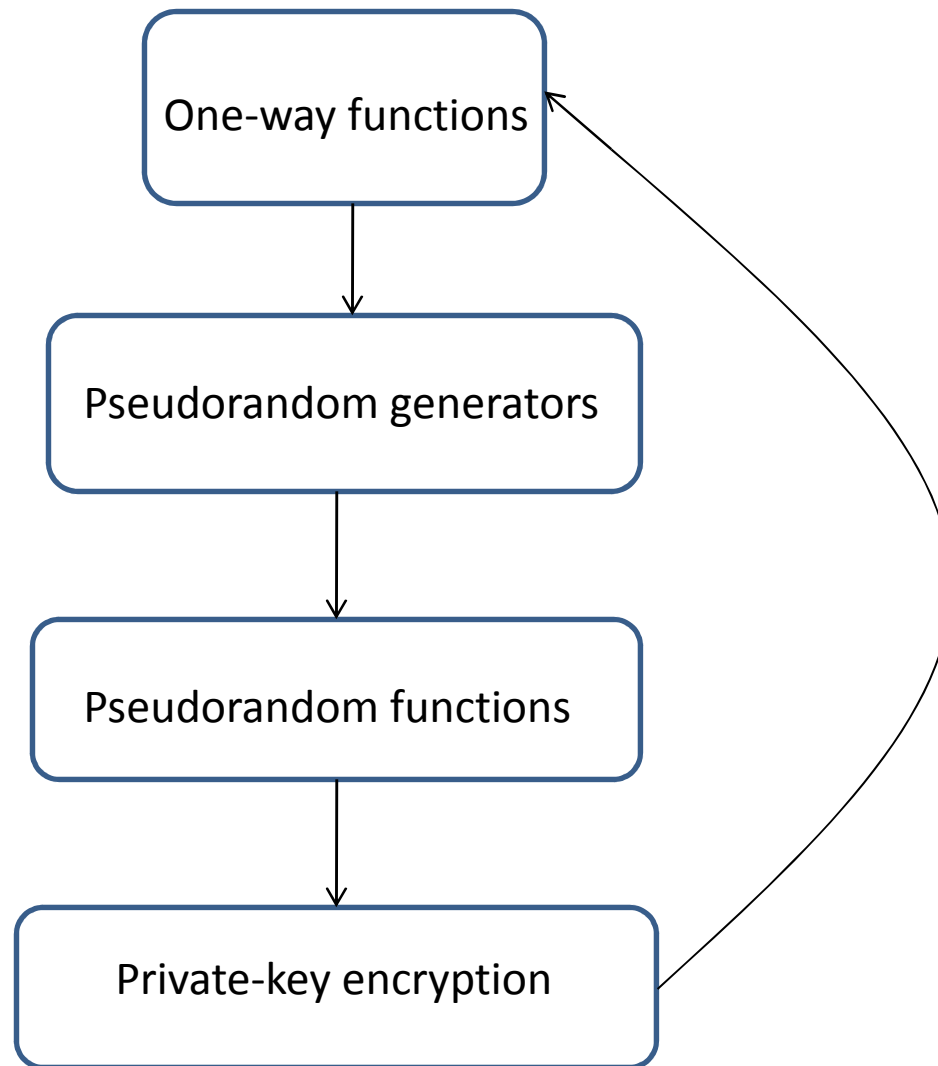
- First attempt
 - $\text{Enc}_k(m) = F_k(m)$
 - $\text{Dec}_k(m) = F_k^{-1}(m)$
 - It is NOT CPA-secure
- No deterministic encryption works

Second Attempt

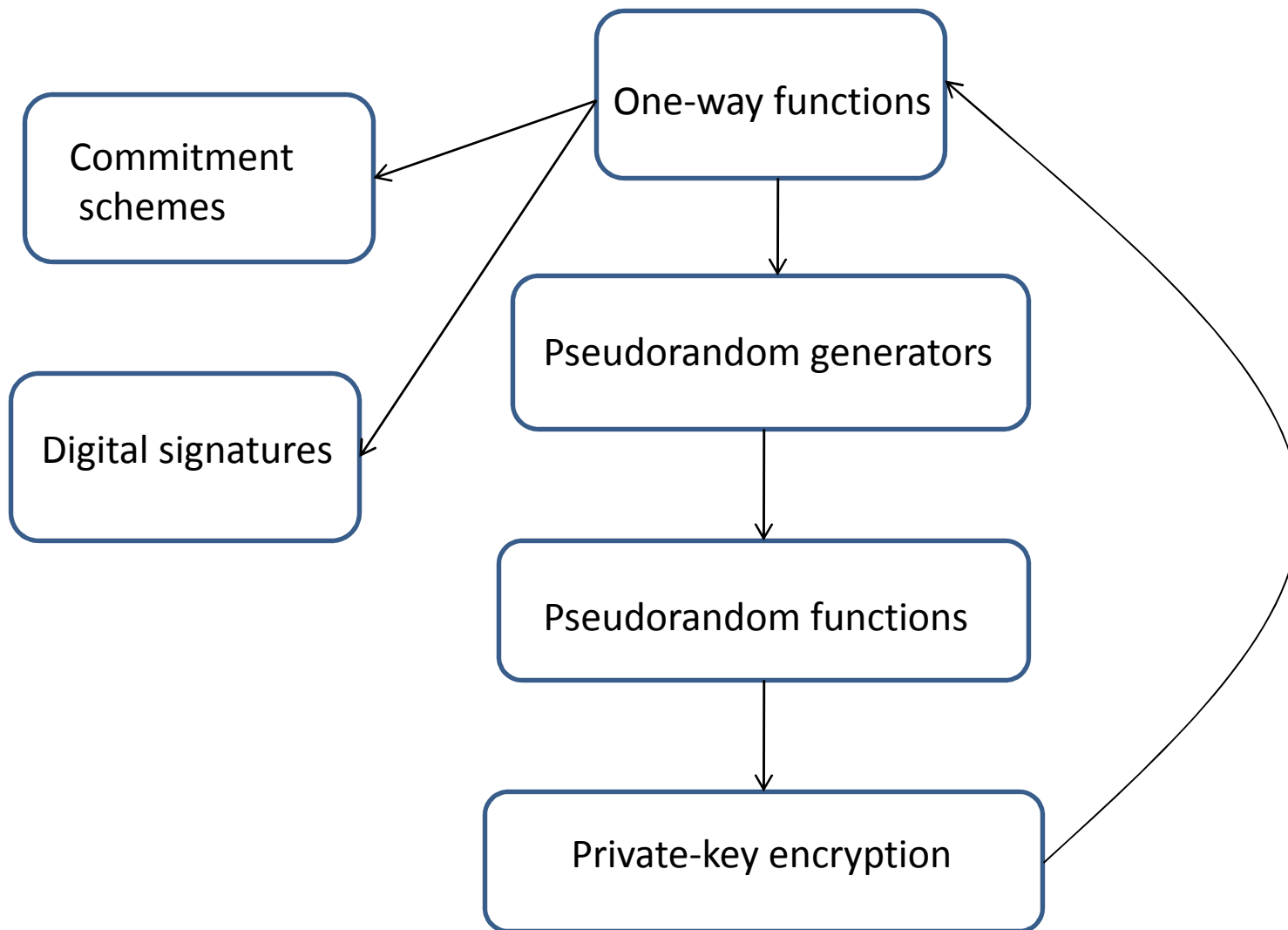
- $\text{Enc}_k(m,r)$
 - $(r, F_k(r) \oplus m)$
- $\text{Dec}_k(r,s)$
 - $m = F_k(r) \oplus s$
- $\text{SE} = (K, \text{Enc}, \text{Dec})$



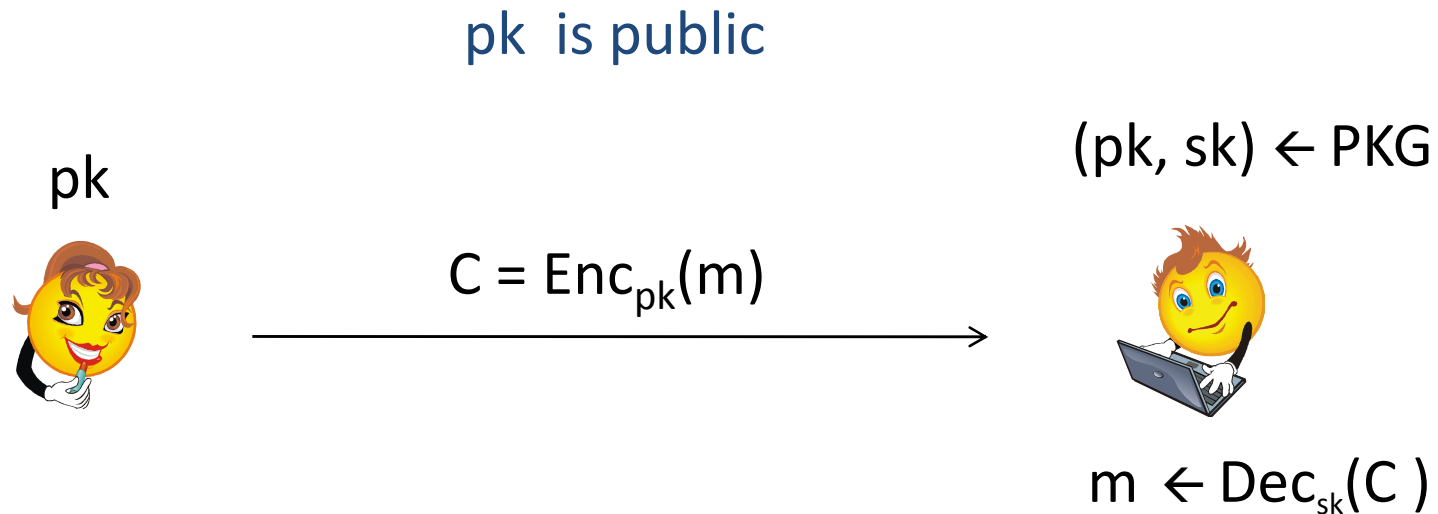
Private-Key Encryption



More Equivalences



Public-Key Encryption



$\text{PKE} = (\text{PKG}, \text{Enc}, \text{Dec})$

Separation Results

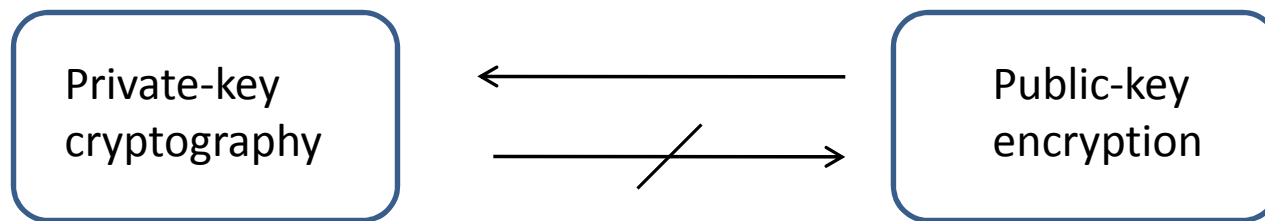
- Claim: Primitive **A** cannot be based on primitive **B**
- No techniques for ruling out all reductions

Black-box Constructions

- Black-box construction
 - A uses B as a black-box
 - No access to its internal structure
- Black-box security reduction
 - For any PPT adversary C breaking primitive A
 - Construct PPT adversary D^C breaking primitive B

Black-box Separations

- Rule out all black-box constructions
- Initiated by Impagliazzo and Rudic, 1989
- Most construction in cryptography are BB
- Non-BB constructions are impractical



- Impagliazzo and Rudic, 1989
- Construct an oracle \mathcal{O}
 - $A^{\mathcal{O}}$ break all Public-key encryption schemes
 - There is a private-key scheme that is secure relative to \mathcal{O}
 - A black-box reduction would contradict this

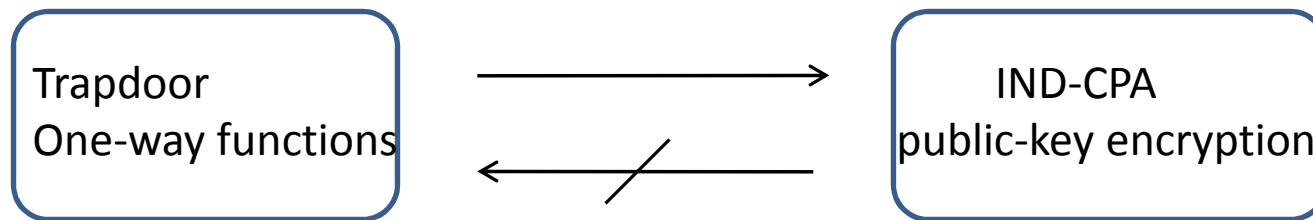
Trapdoor One-way Functions

- Easy to compute
- Hard to invert
- Easy to invert given a trapdoor!

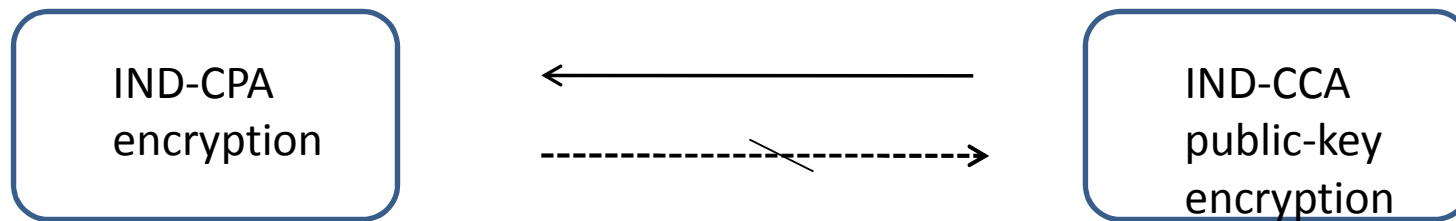
Trapdoor One-way Functions

- TDF = (Gen, F, F⁻¹)
- $(pk, td) \leftarrow \text{Gen}(1^k)$
- $y \leftarrow F(ek, x)$
- $x' \leftarrow F^{-1}(td, y)$
 - $F(ek, x') = y$
 - $F^{-1}(td, \cdot)$ is efficient
- For any PPT adversary A
 - $\Pr[F(ek, x) = F(ek, x') \mid \{0, 1\}^k \leftarrow x, y = F(ek, x), x' \leftarrow A(y, ek, F)]$ is negligible

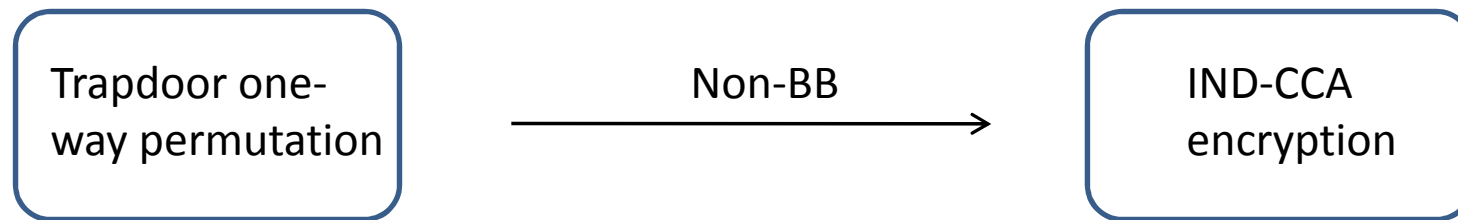
TDFs and PKE



- TDF = (Gen, F, F^{-1})
- Key generation
 - $(ek, td) \leftarrow \text{Gen}(1^k)$
- Encryption of bit b
 - $C = (F(ek, r), b \oplus hc(r))$
- Decryption of (c_1, c_2)
 - $r = F^{-1}(td, c_1)$; $b = c_2 \oplus hc(r)$
- Separation
 - Gertner, Malkin, Reingold, 2001

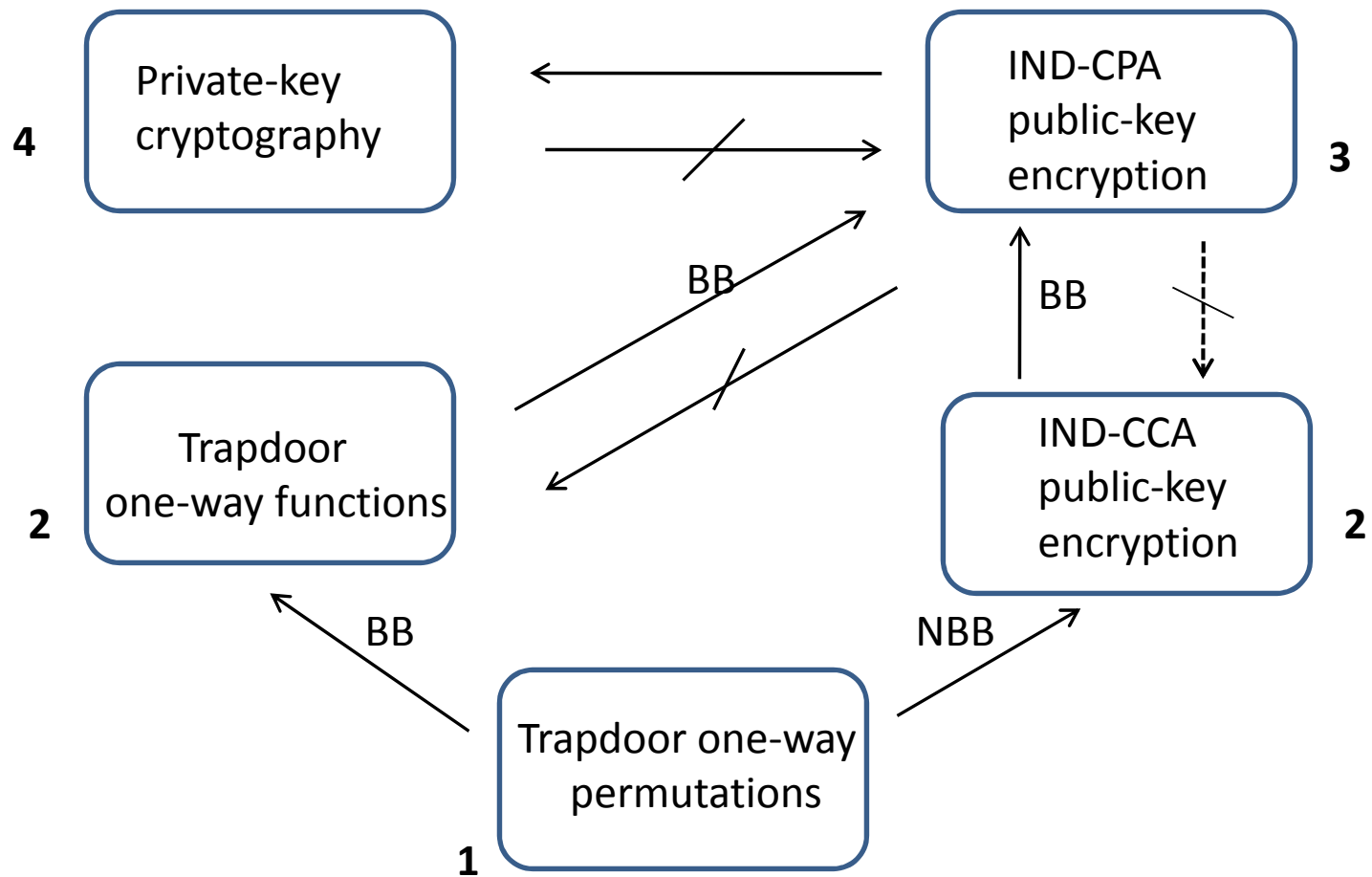


- Partial BB separation
 - Gertner, Malkin, Myers, 2007
 - If decryption of CCA scheme does not use encryption of CPA scheme

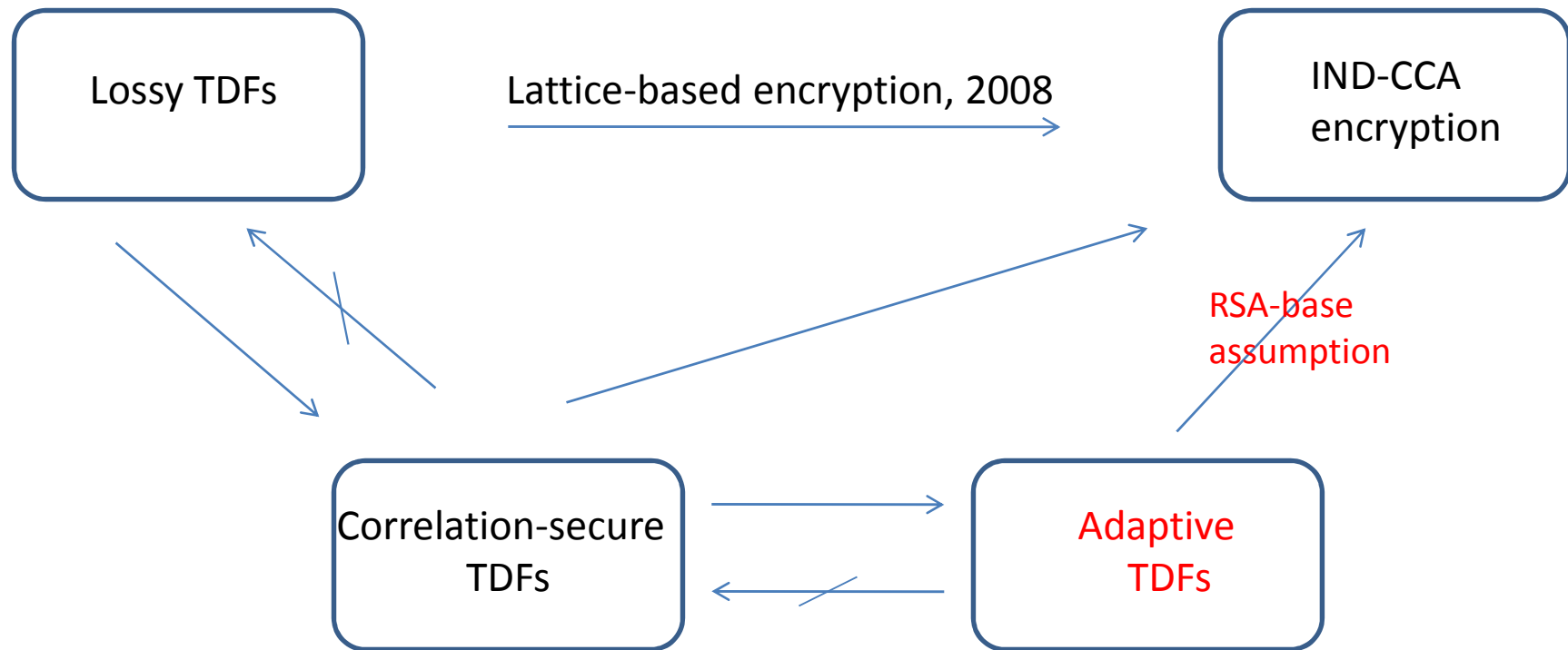


- Dolev, Dwork, Naor, 2000

Relations



Stronger TDF Assumptions



Thank You!