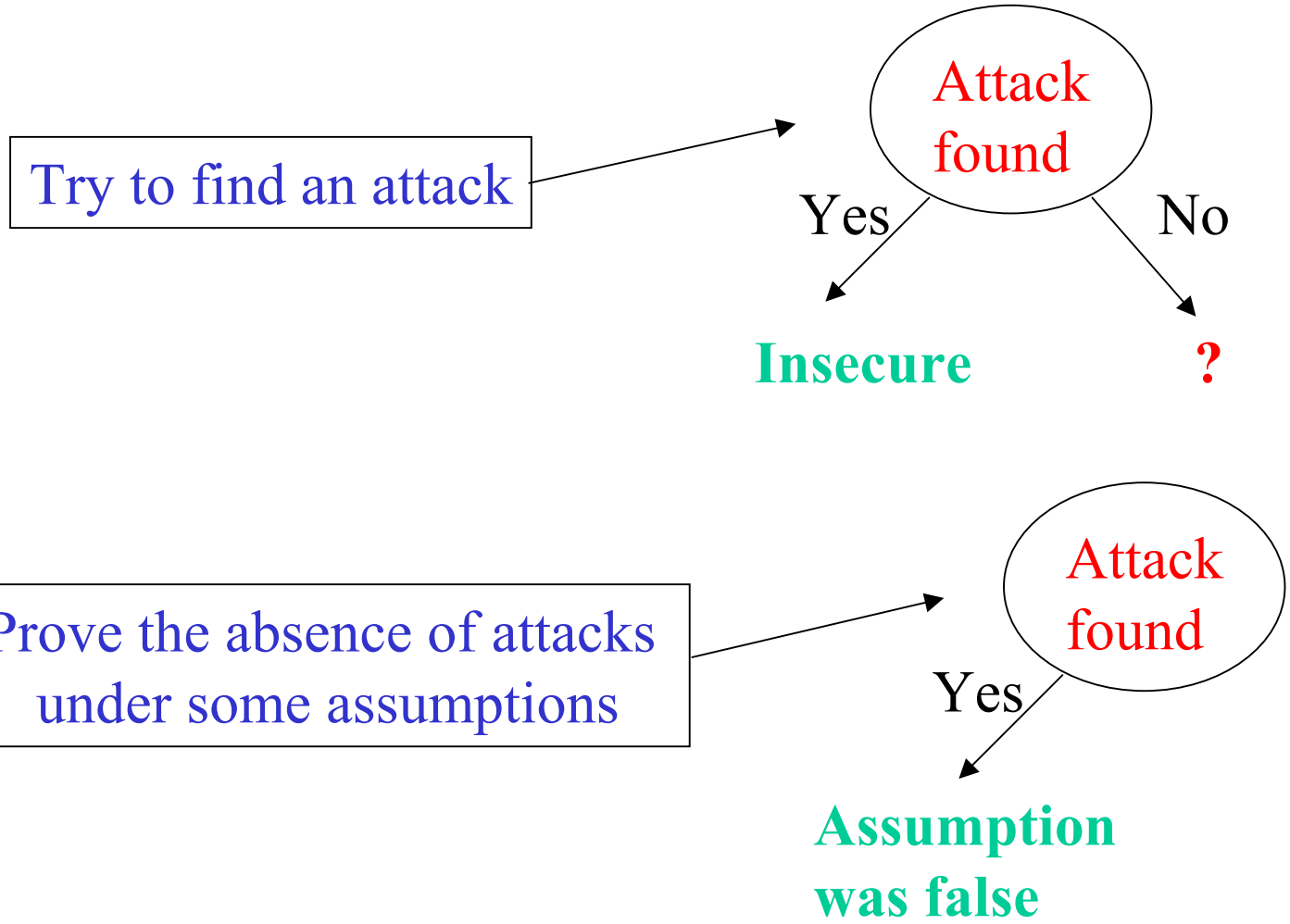
Provably-Secure Public-key Cryptosystems

Mihir Bellare

University of California at San Diego, USA

PKC 2001

How can we be confident that a given cryptosystem is secure?



Evolution of provable security: 1982-1995

Theoretical foundations

Glodwasser-Micali: Probabilistic Encryption

Hardcore bits for RSA & other one-way functions

Existence of provably secure schemes

Some efficient schemes e.g. Blum-Goldwasser

But minimal impact on practice

Evolution of provable security: 1995-present

Efficient proven secure schemes: RSA-OAEP, Cramer-Shoup

Adoption of proven-secure cryptosystems in Standards:

- RSA-OAEP in PKCS#1 v2.0, P1363a
- DHIES in ANSI X 9.63, P1363a, SECG

Standards bodies and even NIST ask for proofs supporting submissions.

Provably secure cryptosystems are widely deployed.

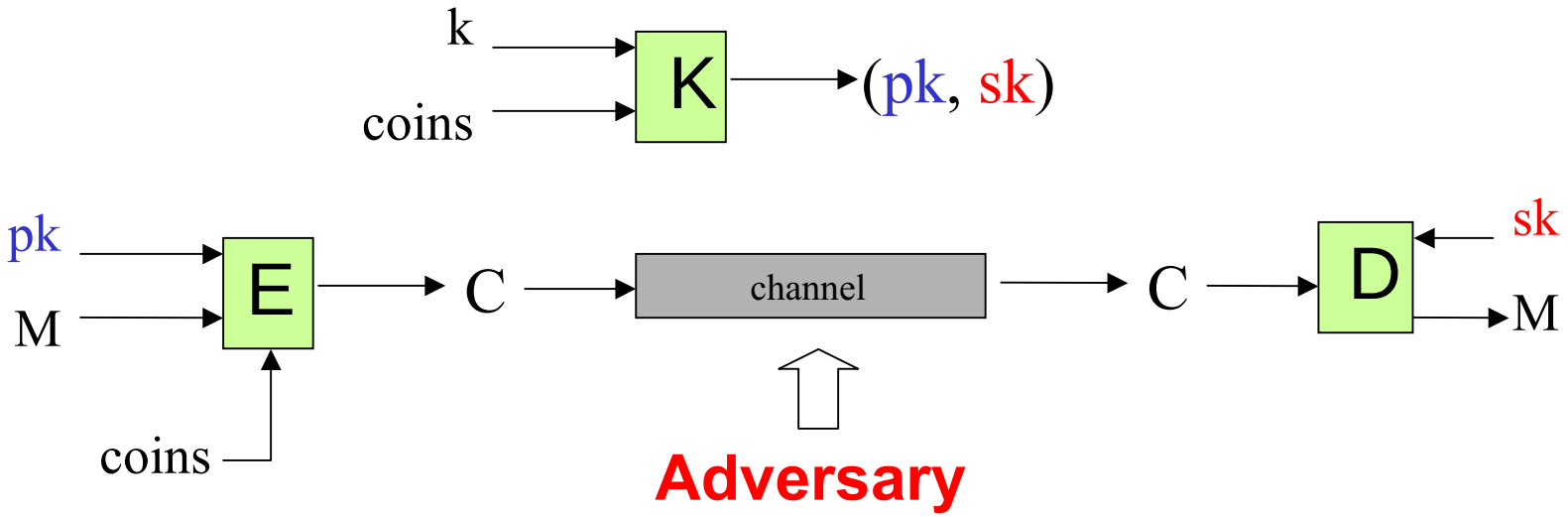
A growing audience

Provable-security is no longer just for theoretical cryptographers. Product developers, security architects and users want to know:

- Which systems to use
- How different cryptosystems compare

A public-key encryption scheme is specified by 3 algorithms:

- Key generation algorithm K
- Encryption algorithm E
- Decryption algorithm D



Example: El Gamal cryptosystem over group G = <g> of size q

$K(k)$	$E_X(k)$	$D_x(Y, \alpha)$
$x \leftarrow Z_q$	$y \leftarrow Z_q$	$K \leftarrow Y^x$
$X \leftarrow g^x$	$Y \leftarrow g^y$	$M \leftarrow \alpha / K$
Return (X, x)	$K \leftarrow X^y$	Return M
	$\alpha \leftarrow KM$	
	Return (Y, α)	

Elements of the provable-security approach

- Definitions / notions of security
- Assumptions
- Reductions

Encryption is supposed to provide privacy of the data.

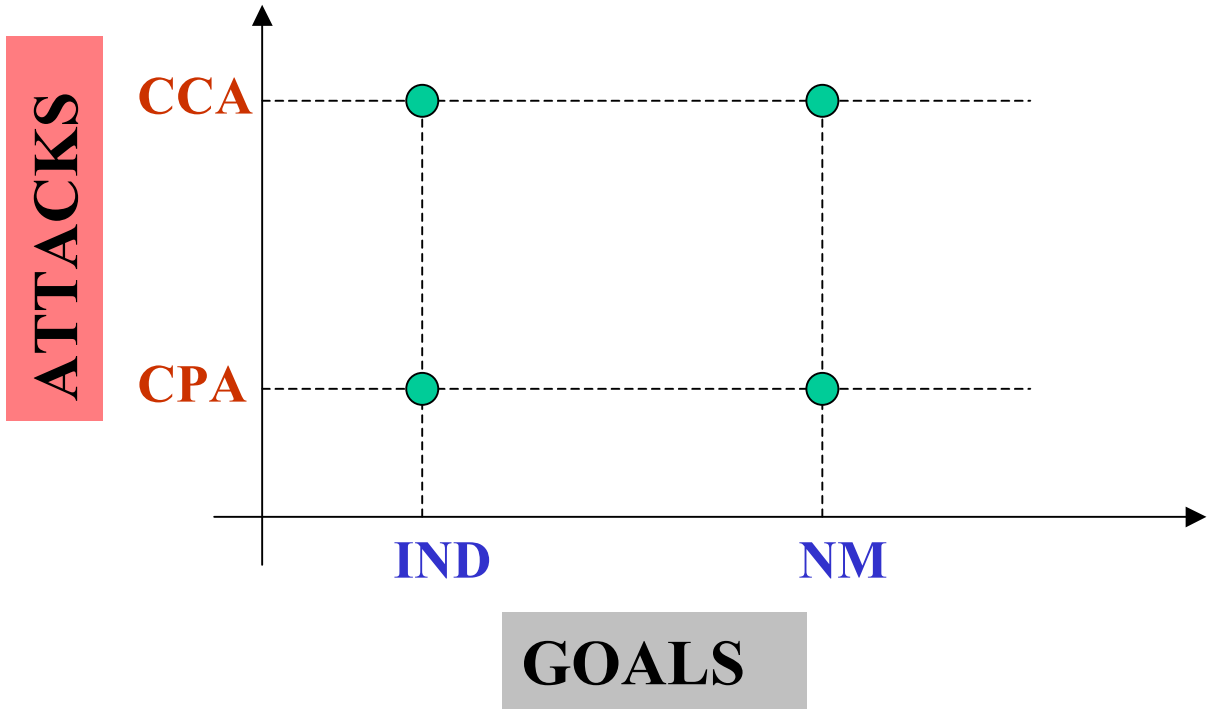
But what exactly does this mean?

Security means	**But…**
Recovery of secret key is infeasible	True if data is sent in the clear
Obtaining plaintext from ciphertext is infeasible	Might be able to obtain half the plaintext
etc	etc

So what is a secure encryption scheme?

Not an easy question to answer …

Security-goals and Adversary-attacks

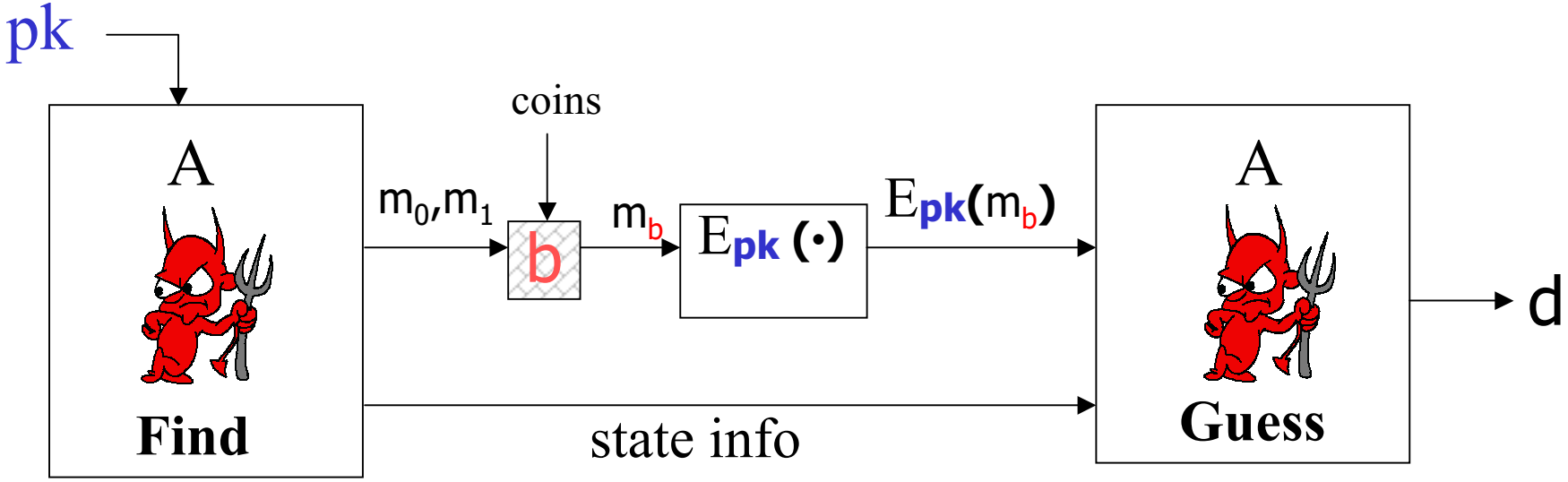


Four notions of security:

IND-CCA NM-CCA

IND-CPA NM-CPA

Indistinguishability under chosen plaintext attack (IND-CPA)



Adversary wins if b = d.

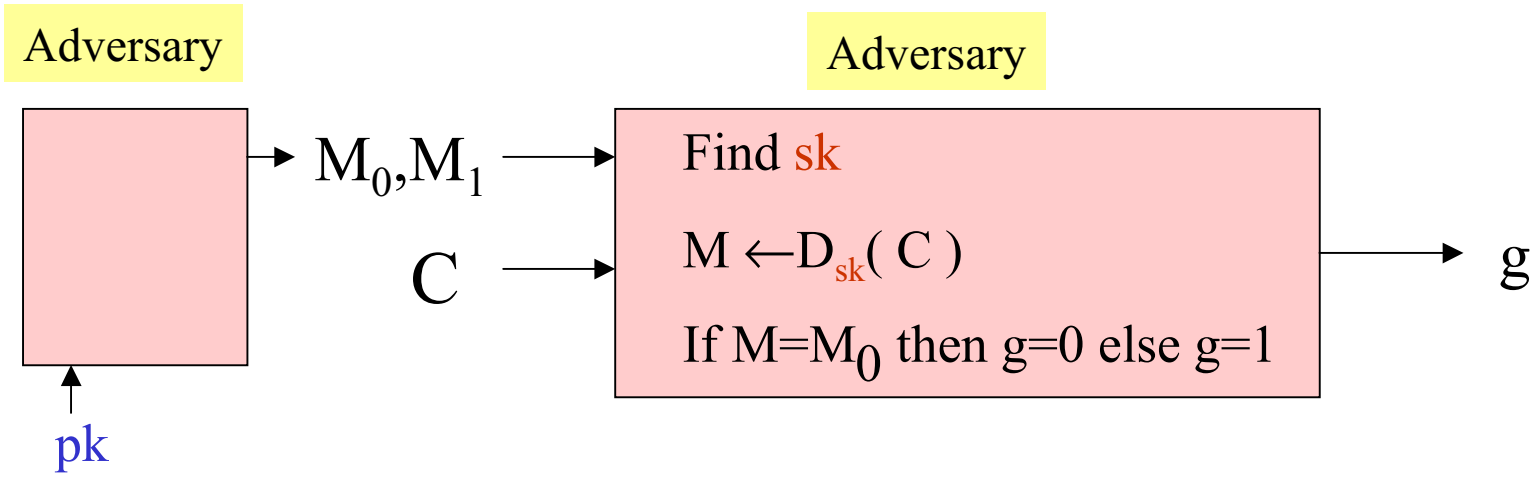
$$\mathrm{Adv}_{E}^{\text{ind-cpa}}(A) = 2\Pr[\text{win}] - 1$$

Scheme is IND-CPA if it is computationally infeasible to obtain a non-negligible advantage.

IND-CPA implies natural security attributes like

- Hard to get sk from pk
- Hard to find first bit of plaintext give ciphertext

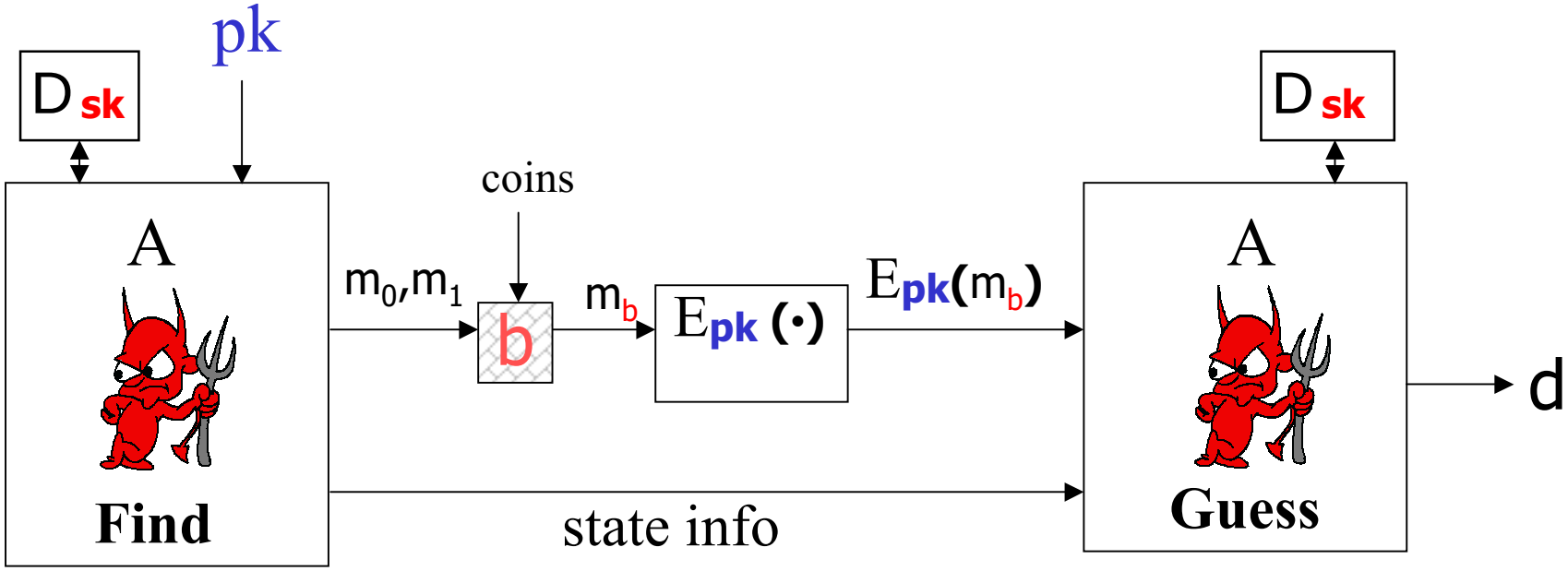
Why? Suppose for example first property failed. Can build an adversary winning the IND-CPA game



Chosen-ciphertext attacks

- Adversary has access to a decryption oracle
- Bleichenbacher: CCA attack on RSA PKCS#1 in SSL

Indistinguishability under chosen ciphertext attack (IND-CCA)



Adversary wins if b = d.

$$\mathrm{Adv}_{E}^{\text{ind-cca}}(A) = 2\Pr[\text{win}] - 1$$

Scheme is IND-CCA if it is computationally infeasible to obtain a non-negligible advantage.

Non-malleability: [Dolev, Dwork, Naor]

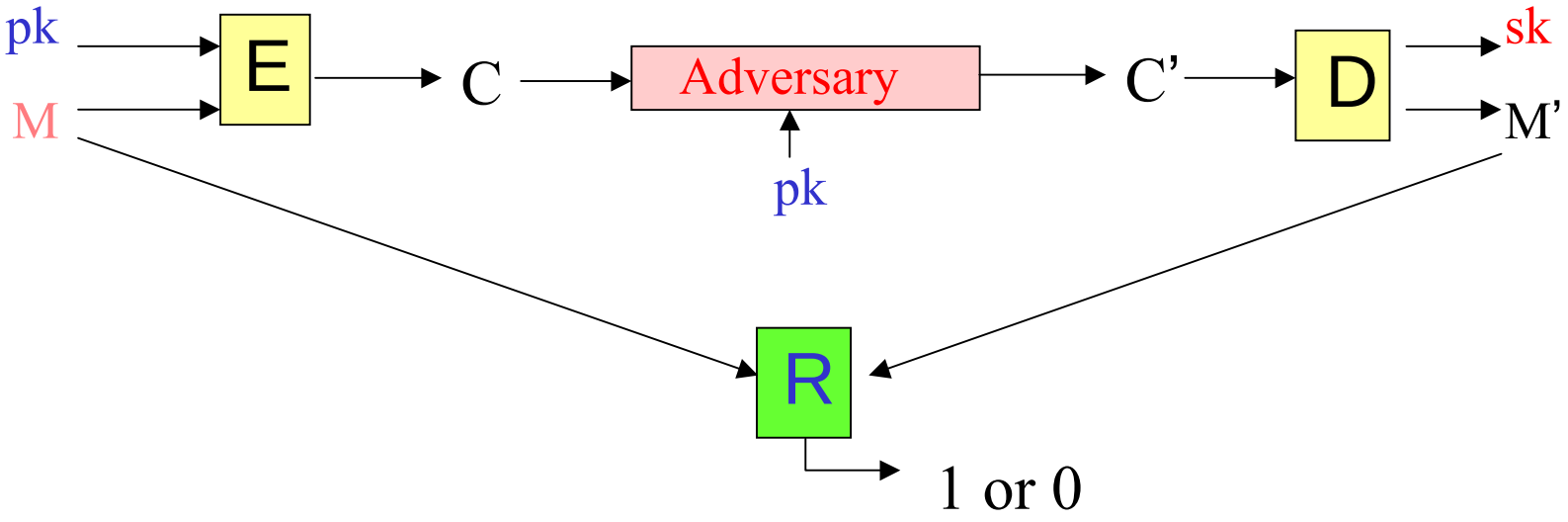
Idea: Adversary cannot modify a given ciphertext into another ciphertext such that the underlying plaintexts are "meaningfully related".

Bad:

Where this might arise: Modify your vote to be opposite of someone else's when votes are encrypted under authority public key.

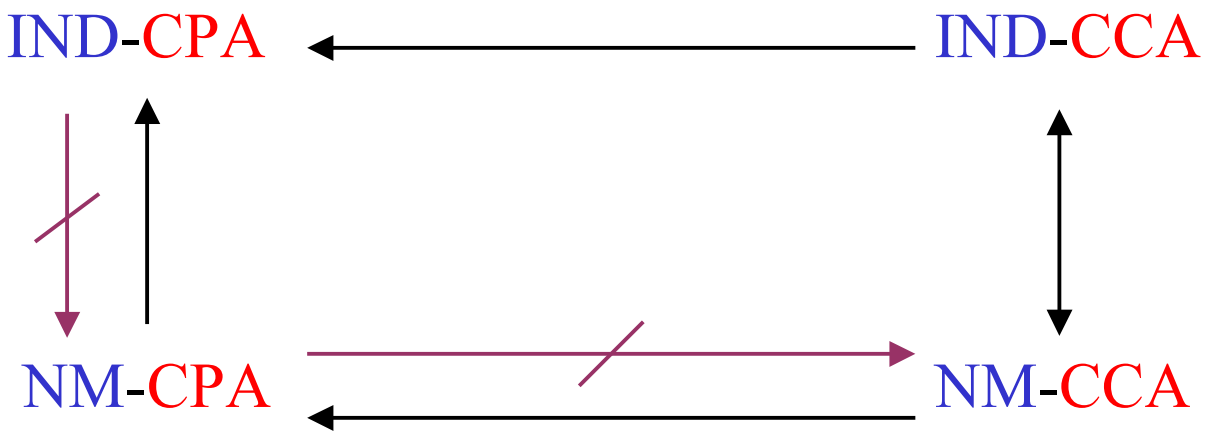
Non-malleability under chosen-plaintext attack (NM-CPA)

M is a distribution of plaintexts, from which M is drawn at random.
R is a relation on plaintexts. Adversary tries to make $R(M,M')=1$.



Scheme is NM-CPA if adversary's winning probability is about the same as when $C \leftarrow E_{pk}(M'')$ for some other $M'' \xleftarrow{R} \mathsf{M}$

Relations among notions [BDPR98, DDN00]



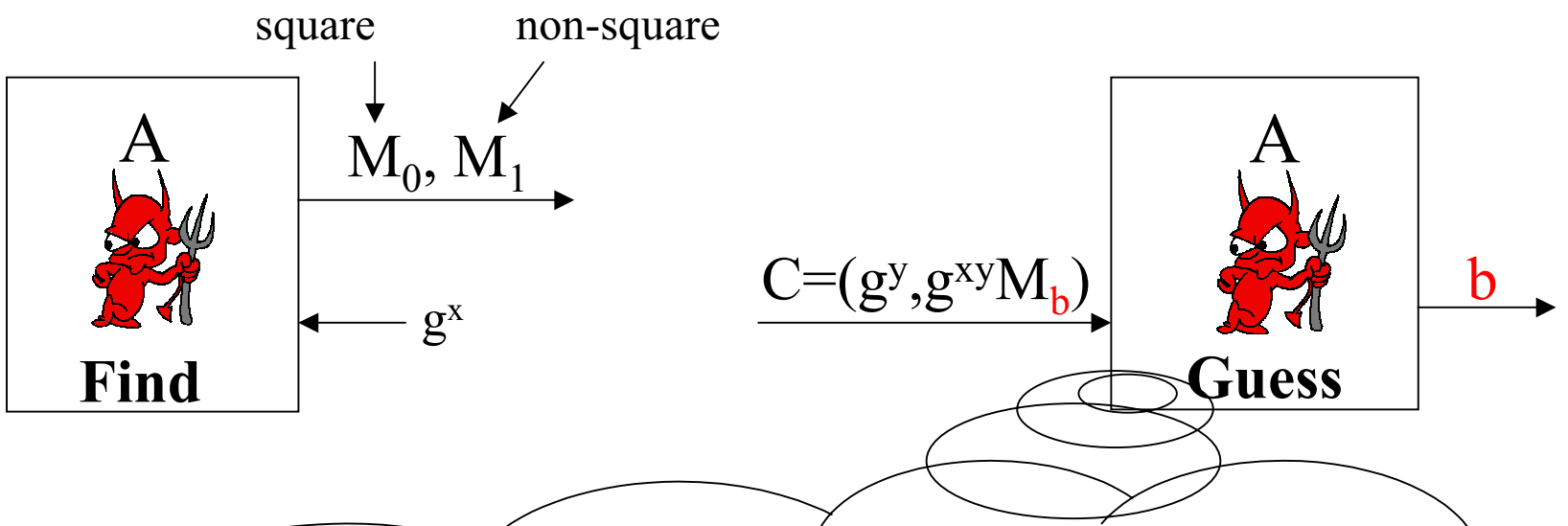
Implication: A ⟶ B : Any scheme meeting notion A also meets notion B

Separation: A ⟶̸ B : There exists a scheme meeting notion A but not meeting notion B

Notions/ Models

- Targets for scheme designers
- Settings for cryptanalysts to find new attacks

El Gamal over Z_p^* is not IND-CPA



g^{xy} is a square iff g^x, g^y are both squares.

g^{xy}	$g^{xy}M_b$	M_b
square	square	square$\Rightarrow M_0$
square	non-square	non-square$\Rightarrow M_1$
non-square	square	non-square$\Rightarrow M_1$
non-square	non-square	square$\Rightarrow M_0$

Want to build cryptosystems meeting these goals. Where do we start?

Atomic primitives / hard problems

- RSA is one-way
- Discrete log problem is hard
- Factoring is hard
- Shortest lattice vector problem is hard
- Computational Diffie-Hellman problem is hard
- Decisional Diffie-Hellman problem is hard
- Decoding random linear codes is hard
- Hard problems on braid groups
-
-

Methodology: Reductions

Given an adversary
breaking the cryptosystem

Construct an adversary
breaking the assumption

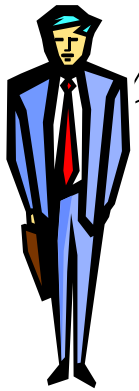
There is no such thing as a proof of security, really. There are only reductions.

I have a new, efficient, provably-secure cryptosystem

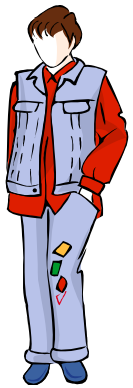
Based on what assumptions?

That the cryptosystem is secure

I have a new, efficient,
provably-secure cryptosystem



What definition of security is it achieving?
What is the assumption?
Can I see the proof?



Provable-security will not help if

- Implementation is incorrect
- Keys are compromised
- There are real attacks not covered in the model
- Proof is incorrect
- Scheme is misused
-
-

CDH

Given: (g^x, g^y)

Compute: g^{xy}

DDH

Given: (g^x, g^y, g^z)

Question: Is $z=xy$ or is z random?

Group	Z_p^*	Prime order subgroup of Z_p^*
CDH	hard	hard
DDH	easy	hard
El Gamal is IND-CPA?	no	yes

Example: If the DDH problem is hard for group G then The El Gamal Scheme over G is IND-CPA

Given adversary breaking The El Gamal cryptosystem

Construct an adversary breaking DDH over G

(g^x, g^y, g^z)

Distinguisher B

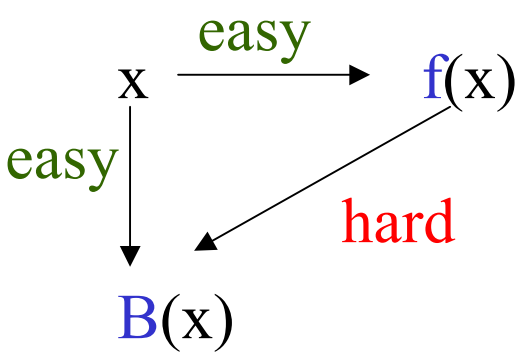
- Give g^x to A as the El Gamal public key
- Let M_0, M_1 be the messages A returns
- Pick a random bit b
- Let challenge ciphertext be $C = g^z M_b$
- Give C to A and get back a bit d
- If b=d then input was a DH triple

Foundations

Is it possible to achieve strong privacy goals like IND-CPA, IND-CCA, NM-CPA, NM-CCA?

What are the minimal assumptions under which one can do so?

Hardcore bits for a OWF f



$pk = f$, $sk = f^{-1}$

Bit-by-bit encryption:

- E(0)=f(x) where B(x)=0
- E(1)=f(x) where B(x)=1

Hardcore bits found for: RSA, discrete-exponentiation, any one-way function

Yields cryptosystems that are IND-CPA secure

Cryptosystems that are IND-CCA / NM-CCA secure

Naor-Yung, Dolev-Dwork-Naor

- Based on any trapdoor permutation
- Use NIZK proofs [BDMP,FLS]

Foundations

Continues to be an active research area

- New notions of security and relations
- New hard problems and cryptosystems based on them
- Reduced assumptions
- Improved results about hardcore bits
- Efficiency improvements
-
-

Meanwhile, in practice

pk = N, e

Simple embedding schemes:

$E_{N,e}(M)$

$x = \text{Embed}(M)$; $y = x^e \bmod N$

Example: RSA PKCS#1

x =	00	02	pad	00	M

Typically M is a key for a symmetric cipher (128 bits)

This scheme cannot be proven IND-CPA assuming only the one-wayness of RSA

Moving theory into practice

Can one design a simple embedding RSA encryption scheme that is provably-secure assuming only that RSA is one-way?

Contrasting approaches:

Theoretical

- Make minimal assumptions
- Schemes "as efficient as possible"

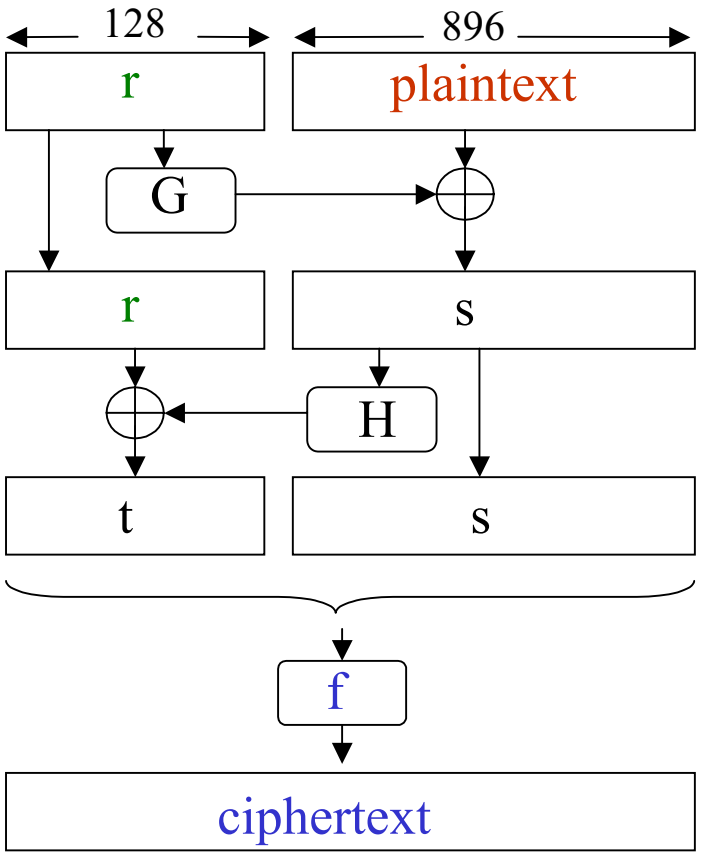
Practice-oriented provable-security

- Schemes as efficient as practical ones
- Prove "as much as possible"

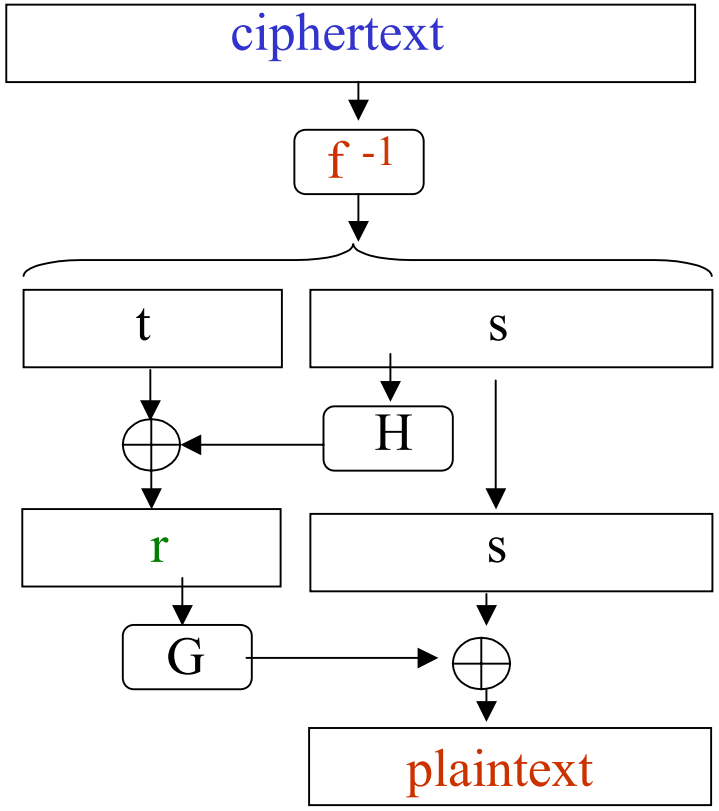
OAEP1 [BR94]

$pk = f, \quad sk = f^{-1}$

Encrypt



Decrypt



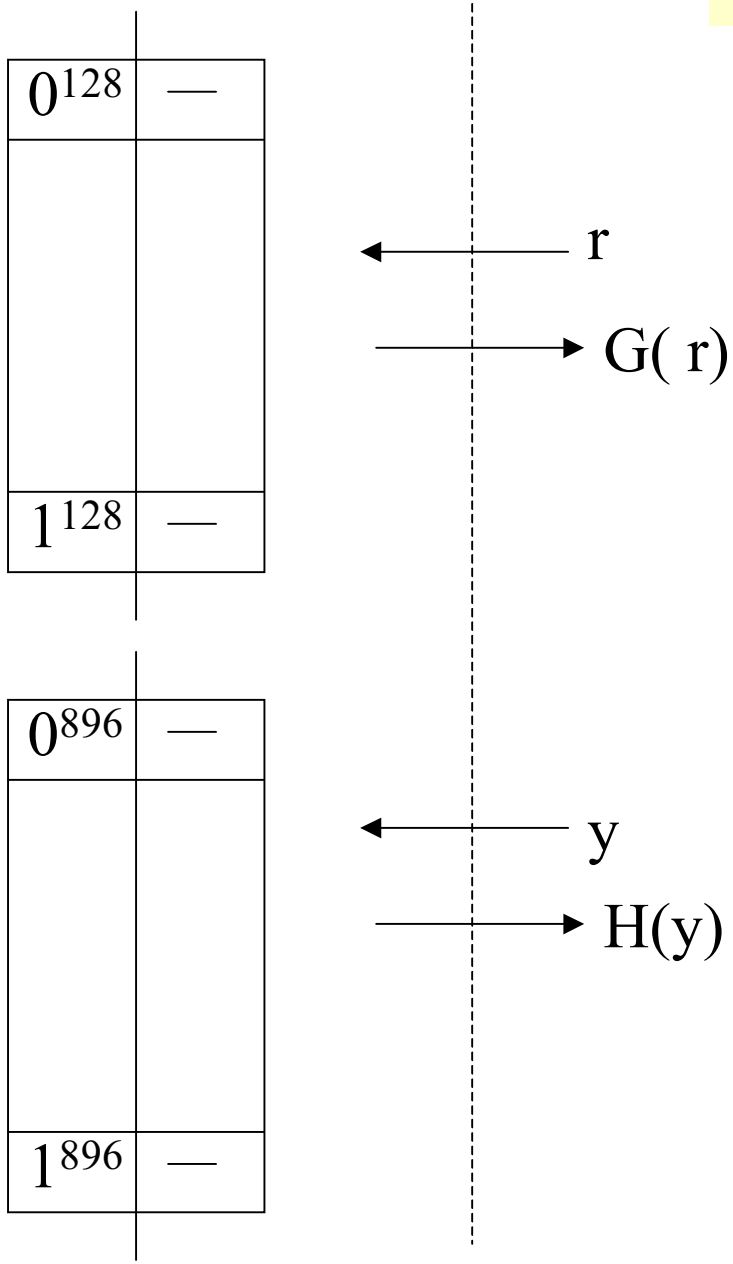
How do we implement G, H?

In analysis: G, H are random functions

In practice: G, H are defined via SHA-1.

Theorem [BR94]: If f is a one-way trapdoor permutation then f-OAEP1 is IND-CPA in the random oracle model.

Random Oracle Model



Parties: Sender, receiver

adversary

Limited number of queries allowed

Reduction for f-OAEP1

Given adversary breaking the IND-CPA of f-OAEP1 in the random oracle model

Construct an inverter for f

f, y →

Inverter I

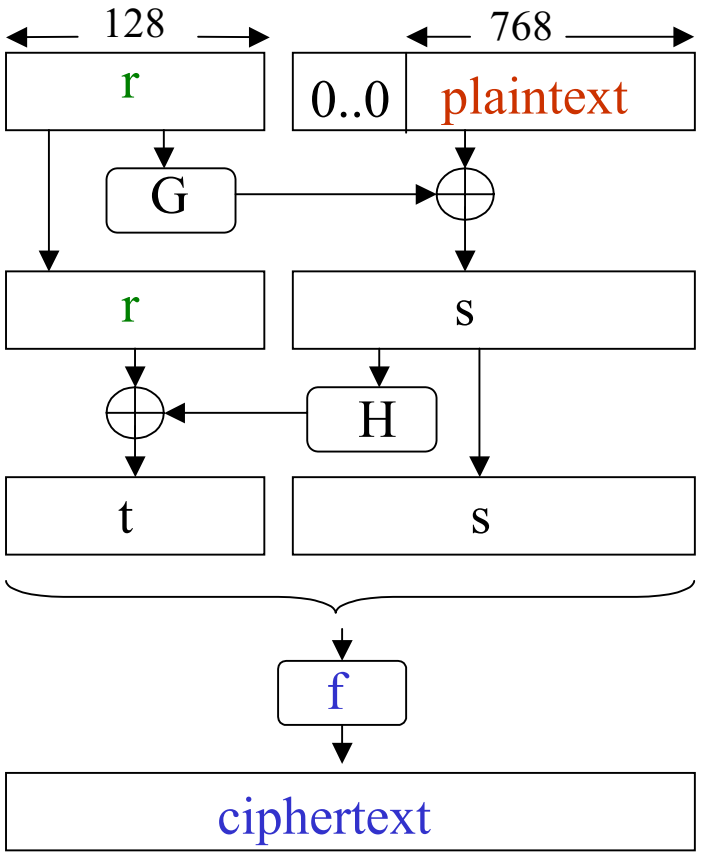
- Run A with pk=f
- Simulate oracles G,H
- Use y as challenge ciphertext
- Watch oracle queries to find $f^{-1}(y)$

→ $f^{-1}(y)$

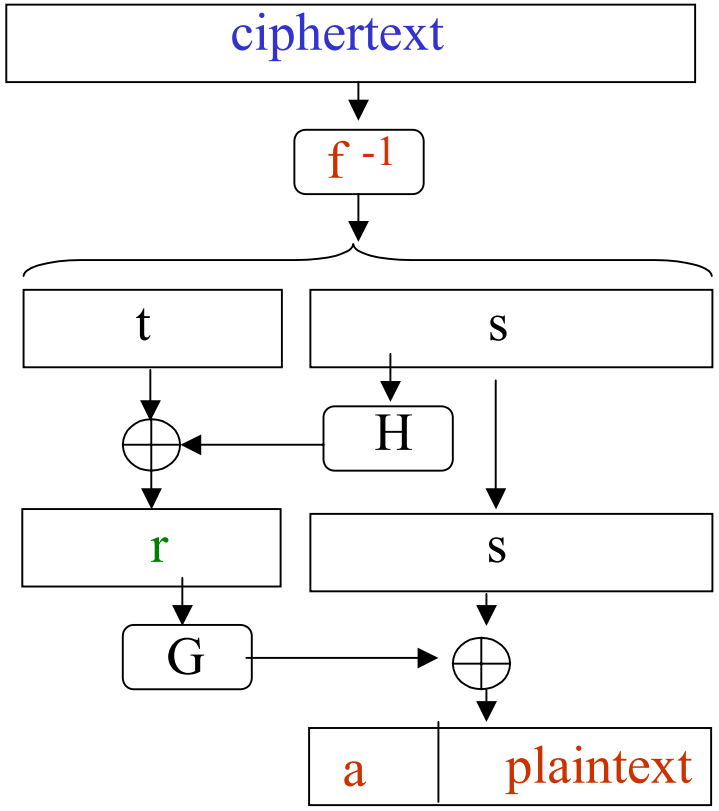
OAEP2 [BR94]

$pk = f, \quad sk = f^{-1}$

Encrypt



Decrypt



Accept iff $a = 0..0$

f-OAEP2

Claim[BR94]: If f is a one-way trapdoor permutation then f-OAEP2 is IND-CCA in the random oracle model.

RSA-OAEP2

1998: Bleichenbacher's attack showed that RSA-PKCS#1 is not IND-CCA.

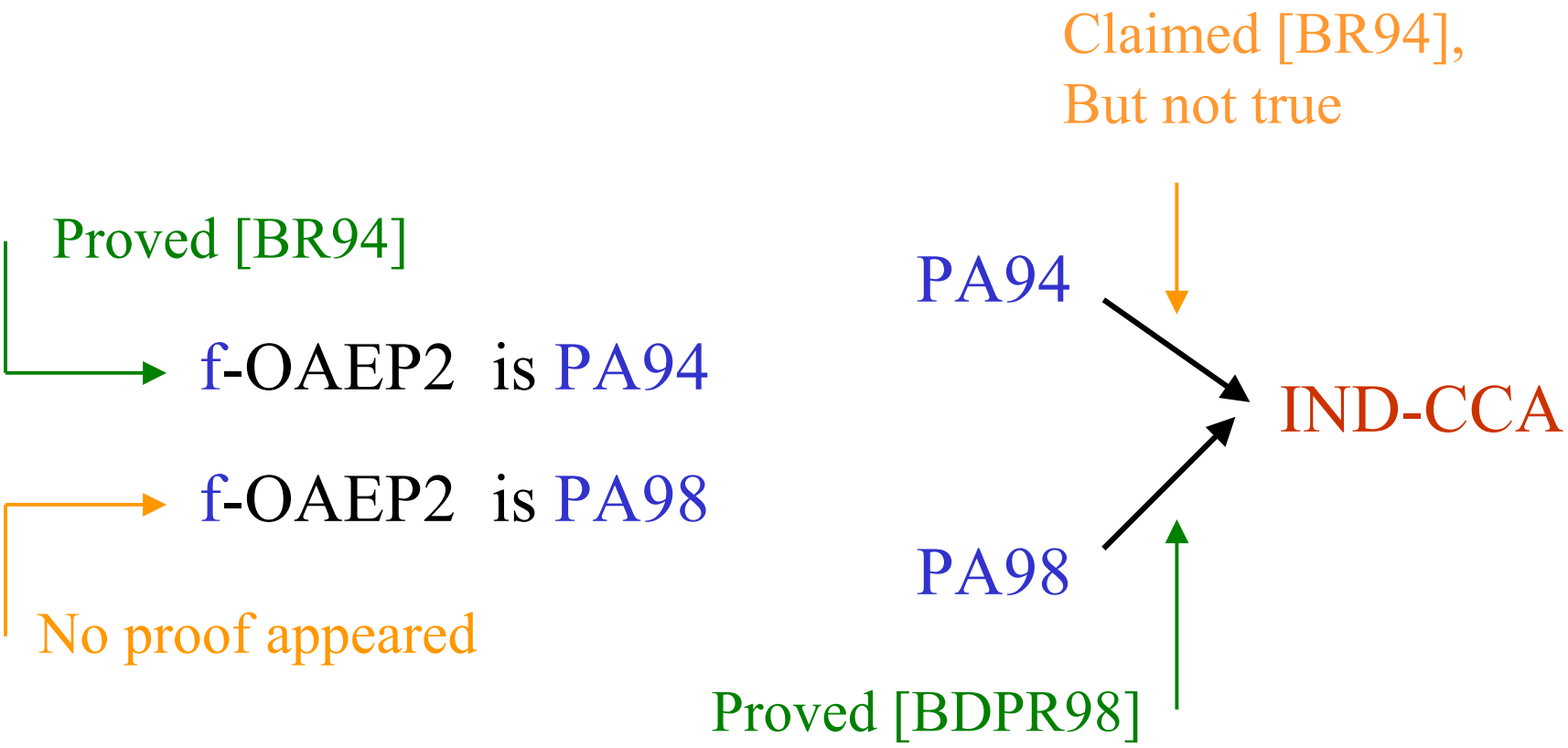
But this attack fails on RSA-OAEP2

RSA-OAEP2 adopted as PKCS#1 v2.0 and considered for other standards.

OAEP2 reconsidered [Shoup, Jan 01]

- No proof of claimed IND-CCA of f-OAEP2 has appeared
- It seems hard to provide one for general f
- But there is a proof for RSA3-OAEP2
- And for RSA-OAEP+

How did this happen?



PA = plaintext-aware

RSA-OAEP2 is still alive [OPS, Jan 01]

Theorem [OPS01]: If f is a partially one-way trapdoor permutation then f-OAEP2 is IND-CCA in the random oracle model.

Theorem [OPS01]: RSA is partially one-way

Novel use of lattices in proof

All's well that ends well …

What does a proof in the random oracle model buy us in practice?

CGH: There are schemes secure in the random oracle model but admitting no secure instantiation.

Proof in random oracle model rules out generic attacks, but not all attacks.

A proof in the random oracle model seems to provide security in practice, but we don't really know why.

Cramer-Shoup Cryptosystem

Cost:

- Encryption/Decryption: 2.5 times El Gamal cost
- Key sizes: 3 times those of El Gamal

Proven to be IND-CCA assuming

- Decision Diffie-Hellman problem is hard
- Hash function used is collision-resistant

No random oracle assumptions

A remarkable scheme with a clever proof!

Comparison

Scheme	RSA-OAEP2	Cramer-Shoup
Assumption	RSA is one-way	DDH
RO model?	Yes	No
Exponentiations	1	5

Which is better? No clear answer.