Public Key Cryptography RSA and Rabin

Nicolas T. Courtois
- University College of London

Part 1

Public Key Cryptography

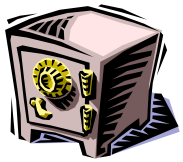
Three Stages in Information Security [Courtois]

3 degrees of evolution:

1. Protections that are secret.

2. Based on a secret key.

3. Private-public key solutions.



Something Quite Unusual and Unexpected

- Re-Birth of Cryptology:
  Invention of Public Key Cryptosystems [1970s].
- The very existence of public key systems is surprising and does not emerge naturally.
  - Even less natural than CR-hashing etc.
    - VERY VERY DIFFICULT to find a new public key scheme (not more than one “good” scheme is invented every 10 years).
- Public key solutions to problems give usually much better security than techniques with a [shared] secret key !
  - Simple reason: key management, the private key is at one single location.

Vocabulary

Public-Key Cryptography ==

Asymmetric Cryptography

For encryption also known as

"Non-secret encryption"

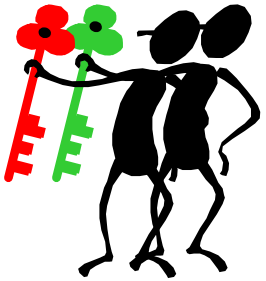
= there is no secret in encryption, there is one in decryption.

- A Private Key =a.k.a.= Secret Key
- A Public Key.

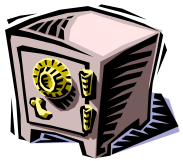
Third Stage – Public Key Cryptography

No shared key, 

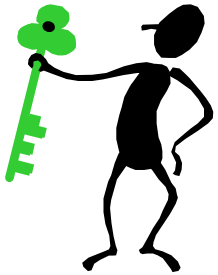
One private and
one public key.



Private key:

generated and stored securely…  

Third Stage – Public Key Cryptography



Public key:

can be distributed to many parties.

Does not have to be public (but the system remains secure when it is).

Public Key Schemes

Symmetric == Conventional Schemes

= 1 algorithm.

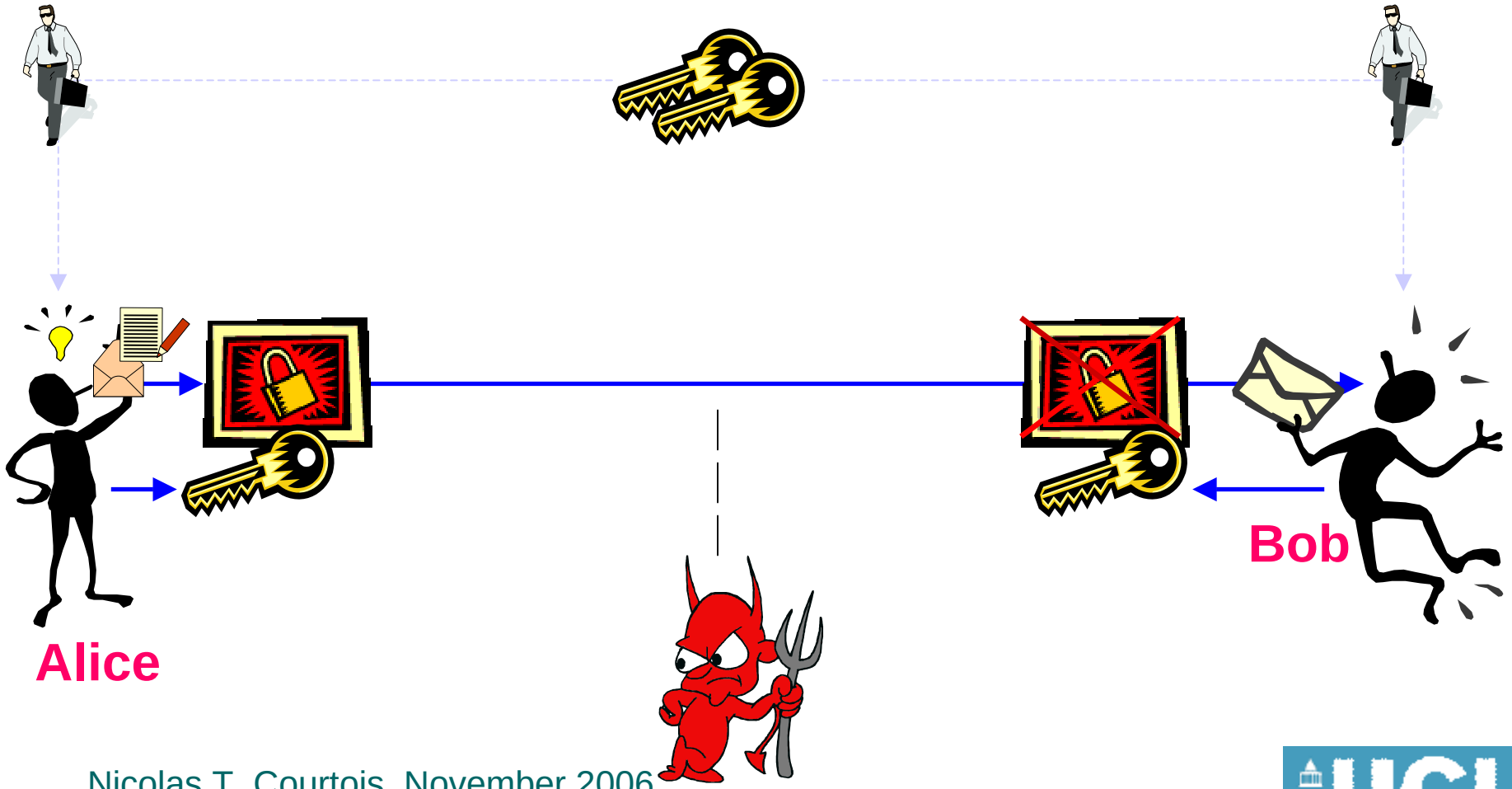
Asymmetric == Public-Key Cryptography

= 3 algorithms:

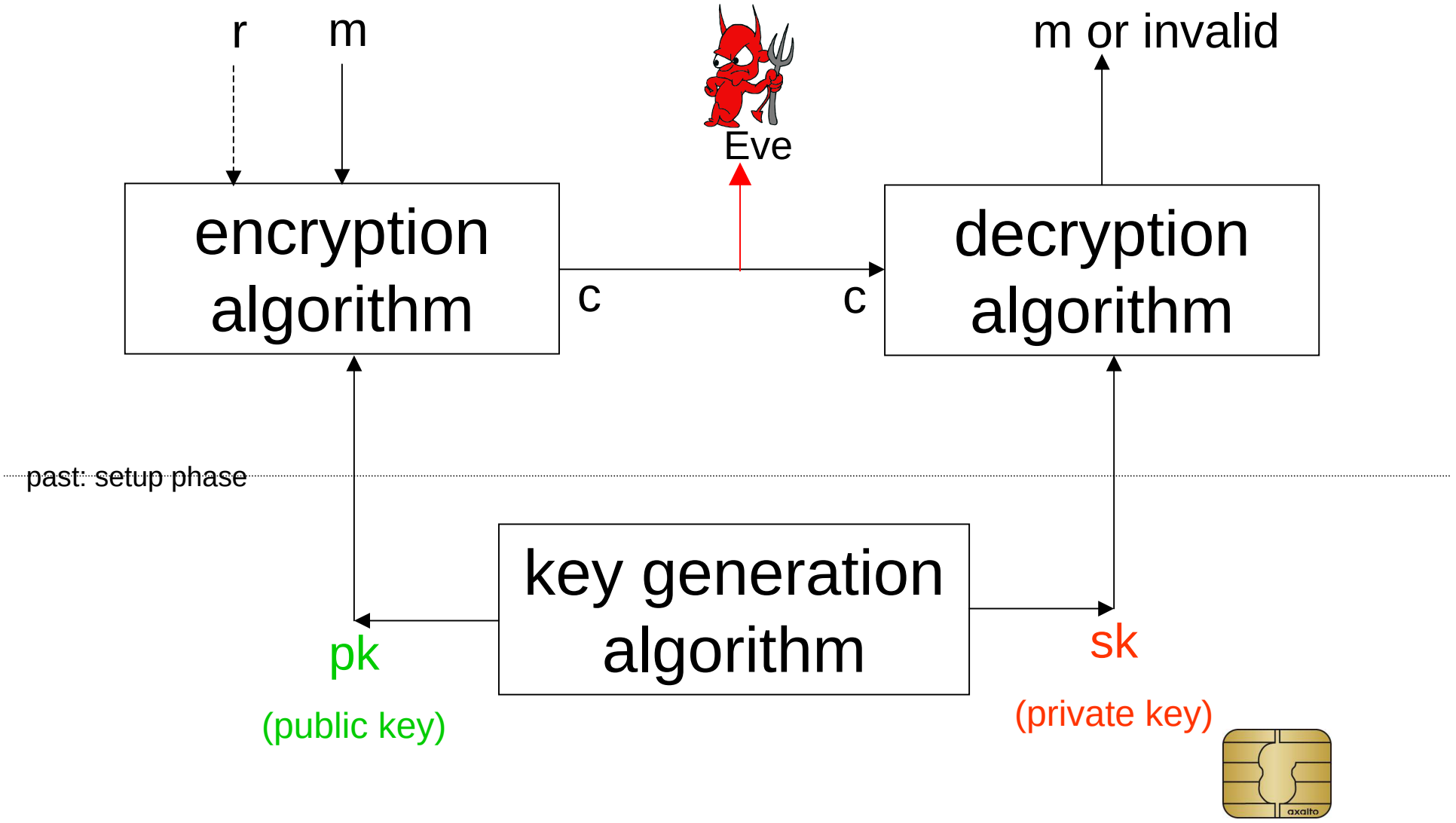
(3 for encryption and signature, in some other cases can be more than 3).

- Key Generation Algorithm
- Encryption / Signature Verification Algorithm.
- Decryption / Signature Algorithm.

***Secret-Key Encryption



Public Key Schemes (Encryption)



Part 1.1.

What PK Encryption Can/Cannot Achieve and what Kind of Setup is Needed (PKI=Public Key Infrastructure)

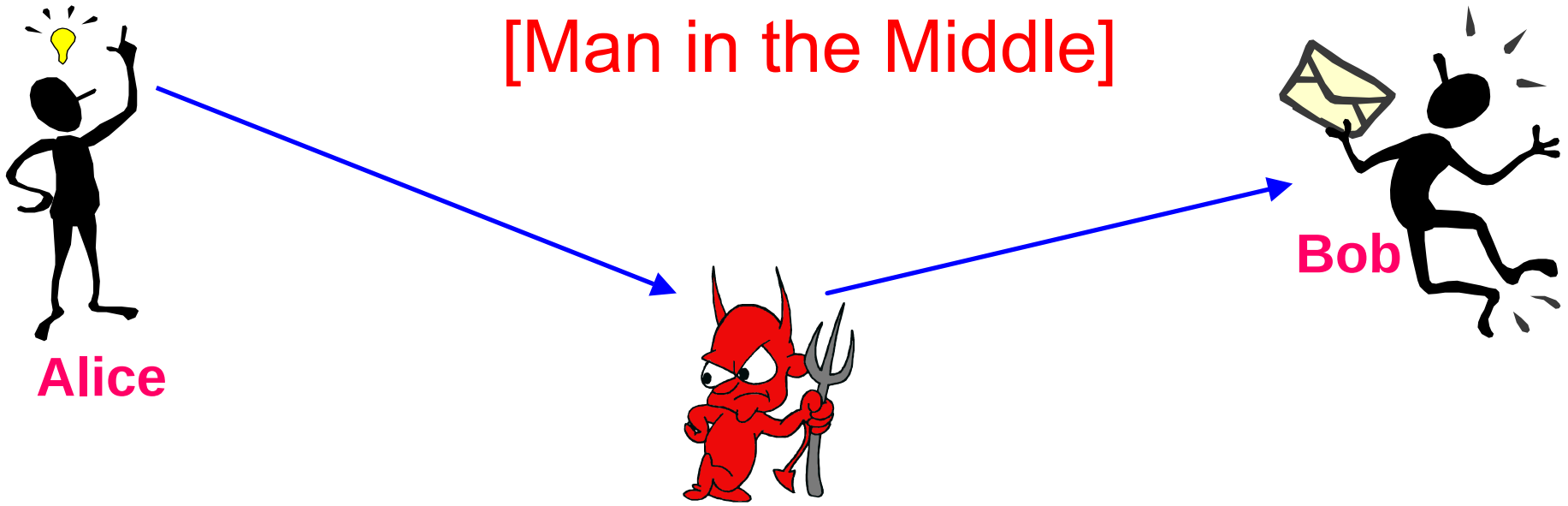
Fact:

Public Key Schemes deeply modify our society and the economy: Create new ways to

- Protect honest people
  - Allows to communicate securely at little cost.
    - Security will be based on “unbreakable” locks, based on mathematical hardness that even the NSA will not be able to break.
- Help dishonest people (unhappily, yes)
  - E.g. can be used to steal data in undetectable ways.
- Do business.
  - E.g. make online auction/casino operate securely.

What Is Achieved by PK Crypto ?

Fact: There is no security possible to two parties that do not know each other and communicate via a public channel.



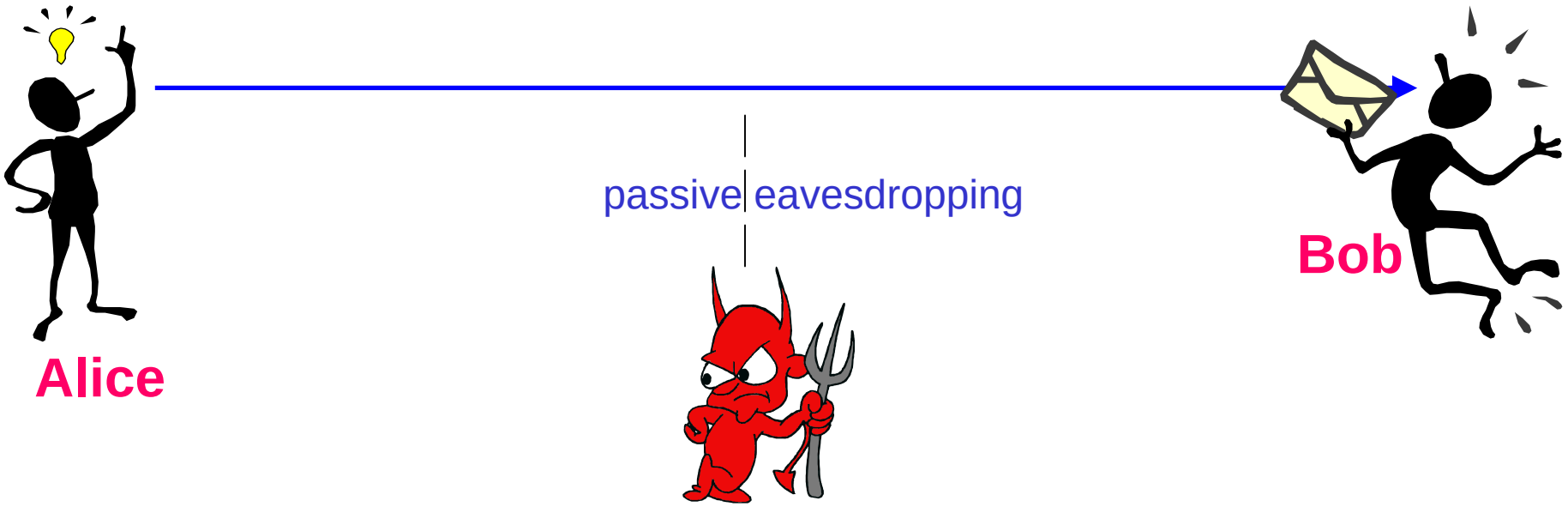
But…

Fact: There is no security possible to two parties that do not know each other and communicate via a public channel.

Security is however possible if there is “some authenticity” available.

Authentic Channel PK Crypto

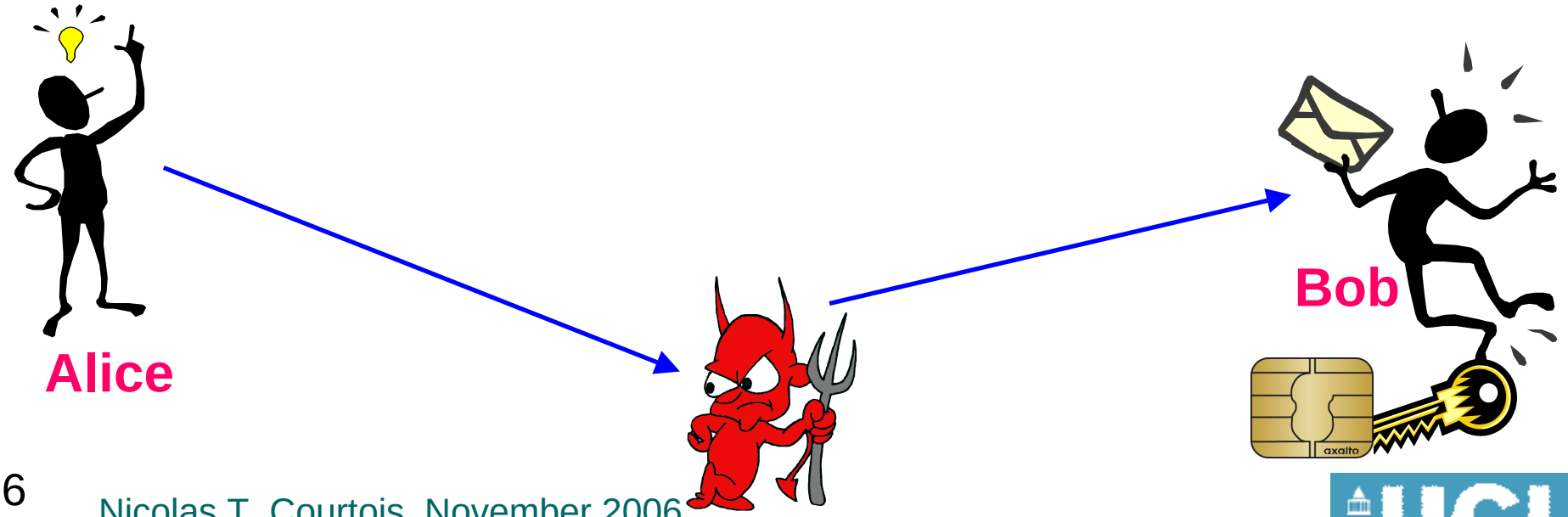
For example, if the channel is authentic:



Can be done With Even Less (!)

Security is however possible

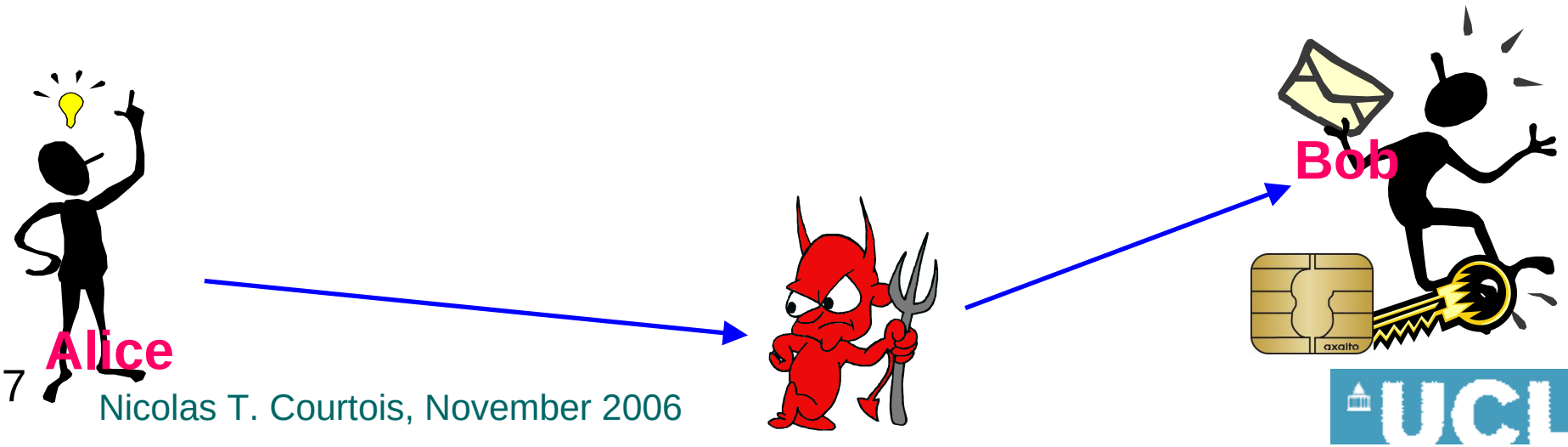
- [stronger] when the channel is authentic / authenticated (!!!).
- [weaker] when a public key of Alice is securely hold by Bob.



Can be done With Even Less (!)

Security is however possible

- [stronger] when the channel is authentic / authenticated (!!!).
- [weaker] when a public key of Alice is securely hold by Bob.
- [even weaker] when at least one authentic public key is hold by all parties. Can be used to certify other keys with digital signatures.



Verification Process – Requirements

- Authentic public key,
- Secure verification process.
  - Done with public key (and in many cases the process can be public too, no secrecy should be required).

  - May require a smart card BUT it is a very different requirement from security of a private key:
    keep it authentic and correct
    (as opposed to keep it confidential and secure at all times).

Can be done With Even Less (!)

Security is however possible … with some public key – called "ROOT of TRUST" + trusted verification (verification process only needs to be secure for <1ms or so)

PK Crypto is ALL ABOUT trading security for authenticity.

(and there is no security without an authentic public key.)

=> Example: If Windows is hacked and there is no TPM/smart card, there is no security for e-Commerce or e-Banking.

* Problems with the PKI Systems

- Cf. Ellison and Schneier: “Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure” http://www.schneier.com/paper-pki.pdf
- Ben Laurie: Seven and a Half Non-risks of PKI.

  http://www.apache-ssl.org/7.5things.txt

Part 1.2.

Basic Attacks in PK Encryption

Basic Attacks (3.)

We assume that the public key is public. Then the adversary always is able to do an adaptively chosen plaintext attack.

- Public Key Only === Adaptive CPA. } basic, illusion of security
- "Reaction Attack": check if ciphertext c is valid.
- "Plaintext-Checking Attack" (PCA): check if a given pair (m,c) is valid. } wholly new scenarios appear
- CCA1-Indifferent/Lunchtime Chosen Ciphertext [NY'90]
  - Decryption of anything of his choice but only BEFORE the adversary knows which ciphertext c he wants to decrypt.
- CCA2-Adaptive Chosen Ciphertext [RS'91].
  - A.k.a. Chosen-ciphertext security (maximum level).

} security against insider attacks !

Part 2

RSA

[Rivest Shamir Adleman, 1977]

[Ellis, Cocks, CESG, UK, c. 1973]

RSA Setup (1) [same for Rabin]

- Generate two primes.
  - Recommended (though nobody can prove it is necessary for RSA security) – “safe” primes, $p=2p'+1$.
- $n = pq$.
- Publish n.
- Keep p and q private.

RSA Setup (2) [same for Rabin]

- Pick e = the encryption exponent.
  - Must be such that GCD(e,φ(n))=1.
  - Frequent choices: e=3 and e=65537=2^{16}+1.
- Publish n, e.
  - In every issue of Le Monde ? - Expensive
  - Make it signed by a known authority = a CA.
    - E.g. Verisign.
- Let d=e^{-1} mod φ(n) be the decryption exponent.
- Keep d, p and q private.
  - BTW. It is sufficient to know <u>just any one</u> out of the following: d, p, q, φ(n).

Computing in the Group $Z_n^*,*$

Key/trapdoor information

that allows to "unlock" $Z_n^*,*$

== the size of the group

== $\varphi(n)$.

Recall: Factoring $n=pq$ <=> compute $\varphi(n)$ (as shown before)

Key Information that Allows to "Unlock" Z_n^*,*

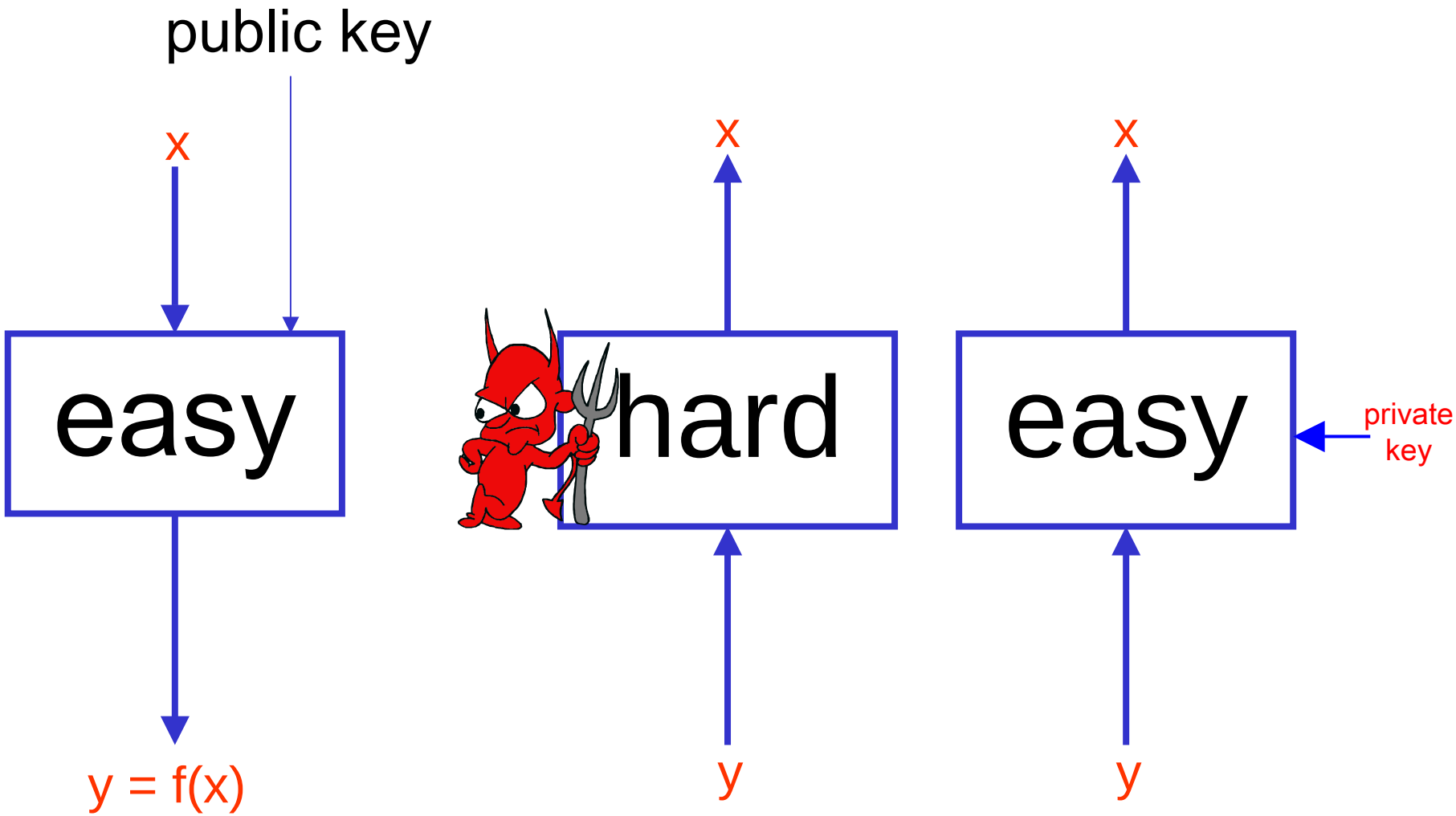
With $\varphi(n)$: we can compute $d=e^{-1} \bmod \varphi(n)$.

- Then $\forall$ x if $y = x^e \bmod n$ then $x = y^d \bmod n$.

Without $\varphi(n)$, nobody knows how to compute y !

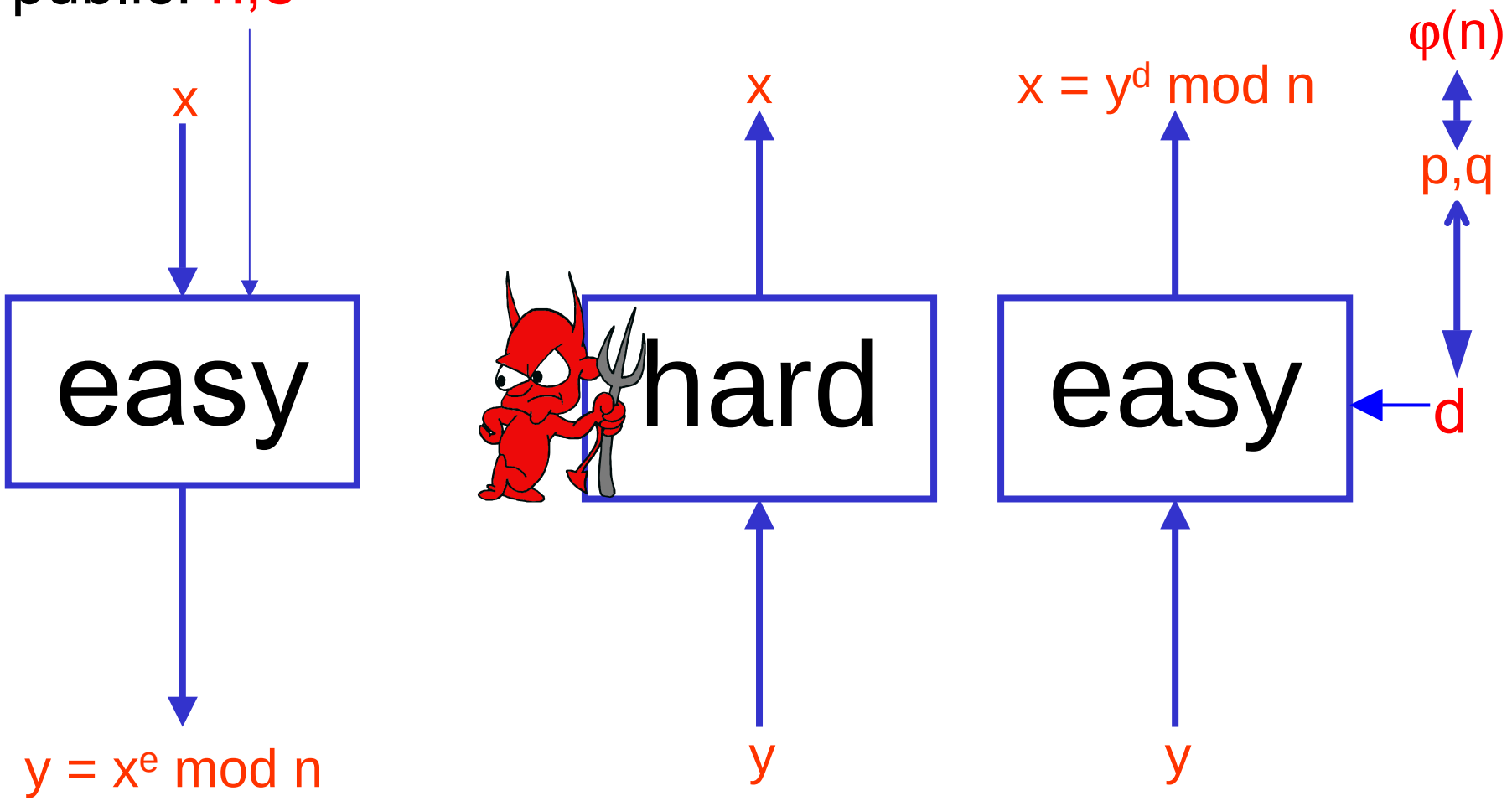
- The number of things that we know how to compute given only n is quite limited (+,-,*,/,exponentiation,GCD,reduction mod n).

Trapdoor One-Way Function - Definition



RSA - Trapdoor One-Way PERMUTATION

public: n,e



Textbook RSA

- The most basic PK encryption method.
- Never use it, is a building block to build secure encryption systems.
- Needed:
  - RSA textbook+”good” randomised padding.

Textbook RSA Encryption AND Signature

Assume $m=0\ldots n-1$. With very high probability $m \in Z_n^*$.

Let $d=e^{-1} \bmod \varphi(n)$.

- Encryption: $c=m^e$.
  - Bijective.
- Decryption: $m=c^d$.
  - Correctness: Fermat-Euler Thm.

- Signature: $\sigma = m^d$.
- Verification: $m \stackrel{?}{=} \sigma^e$.

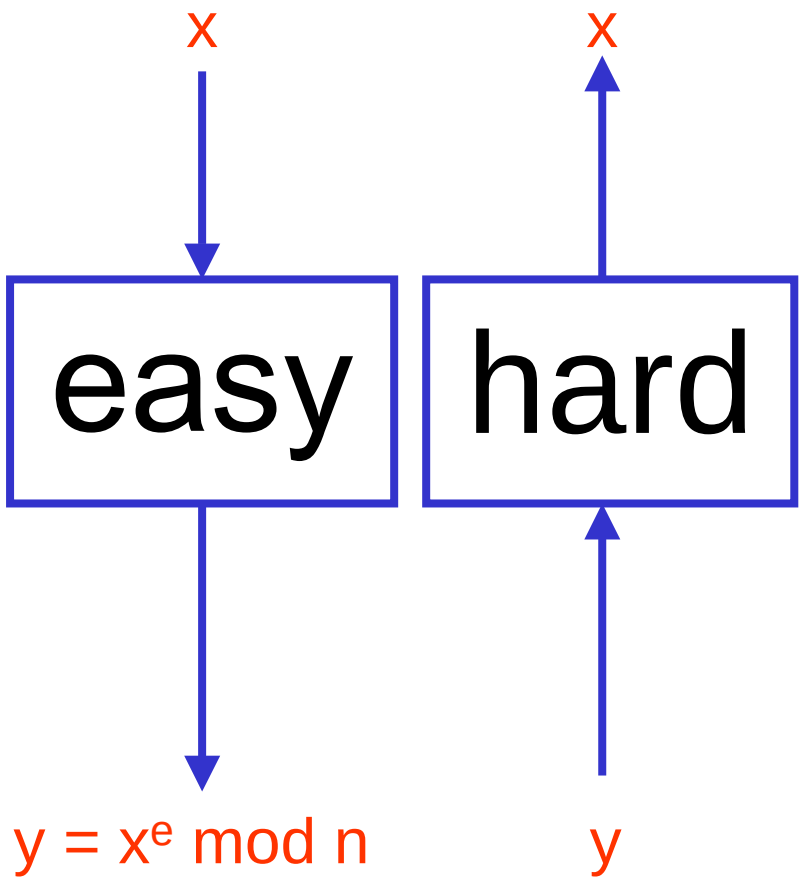
Security of RSA

Not yet a secure public key encryption scheme...
[cannot be deterministic etc..]

The only thing that one should require [for fear of being very disappointed and your security being badly cracked] from “textbook RSA” is One-Wayness (OW):

The RSA Problem

RSA Problem ==
"Textbook RSA" encryption
is OW-CPA2 secure.



Definition: The RSA Problem.

Given y chosen uniformly
at random from 0…n-1.

Find x such that $x=\sqrt[e]{y}$.

Note: $x \mapsto x^e \bmod n$ is one-to-one.

**One-Wayness

One-Wayness is weak, and certainly insufficient for privacy:

- Example: it is hard to recover the whole plaintext, but it may be very easy to recover 90 % of it.

- Surprisingly enough, cryptology allows to build very strong security on this weak notion…
  - Example: RSA-BR given at the end of these slides is a provably secure semantically secure [IND-CCA2] scheme.

Part 2.1.

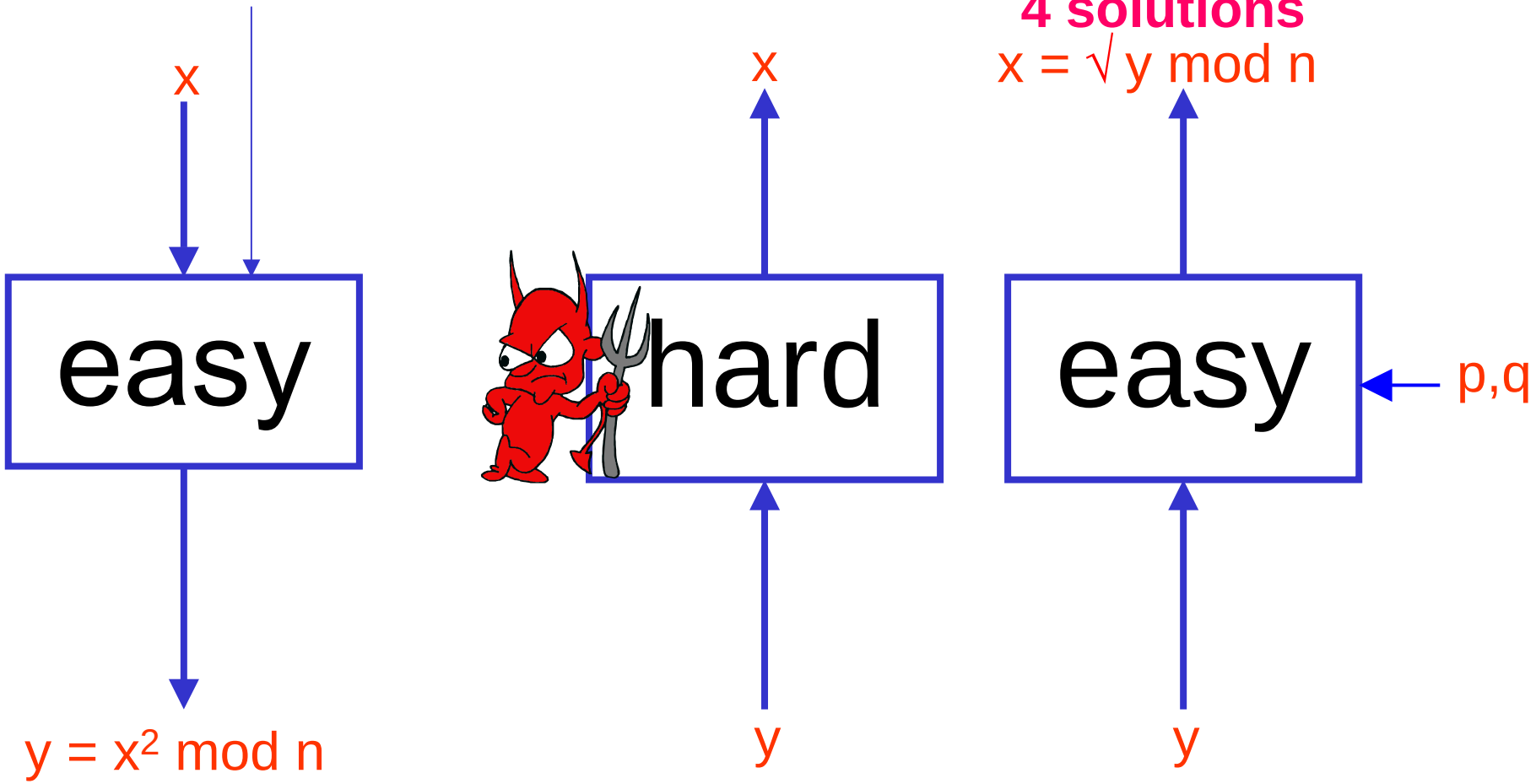
Rabin

[Similar, variant of RSA, but which is not RSA]

Rabin - Trapdoor One-Way Function

public: n

slight problem:
4 solutions



****Compute $\sqrt{}$ mod n=pq

- We need factors of n !
  - Compute it modulo p.
  - Compute modulo q.
- Then use CRT to get $\sqrt{}$ mod n.

Problem of Rabin: There are 2*2 solutions.

- No problem for signatures.
- For encryption: Use redundancy:
  - Either $c = (m||h(m))^2 \bmod n$.
  - Or $c = (m^2 \bmod n \ || \ h(m))$.
    - Both methods are OK.

Security of Rabin vs. RSA

Based on factoring.

Arguably better than for RSA [no proof, no certitude].

This is because maybe RSA is weaker than factoring [so far there is no result in this direction].

Security of Rabin

factoring n ⇔ extract sq. roots mod n.

Proof: DIY.

Build an extractor of the factors of n.

- There are 4 roots.
- It fails with probability ½.
  - Therefore it works with probability ½.

Remark: works only if I can have “private” random bits…

How Secure is RSA ?

Factor 2048 modulus: 200 000 $.

Breaking Elliptic Curves: 725 000 $

=>nobody can claim these are broken…

For AES nothing, not even 1 dollar (!)

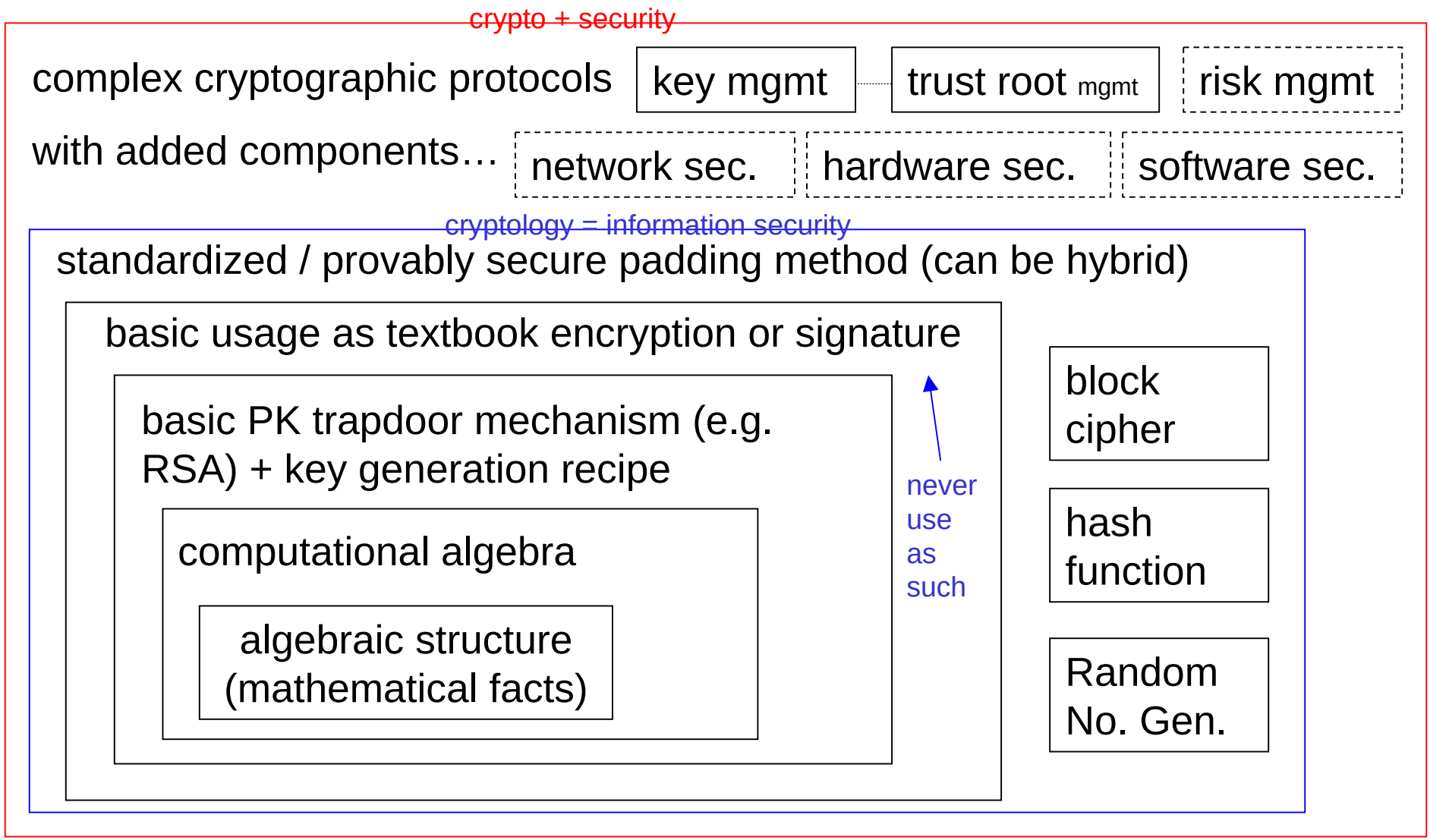
=>and 40 % of people believe it is already broken (internet poll among people concerned by IT security).

www.cryptosystem.net/aes/

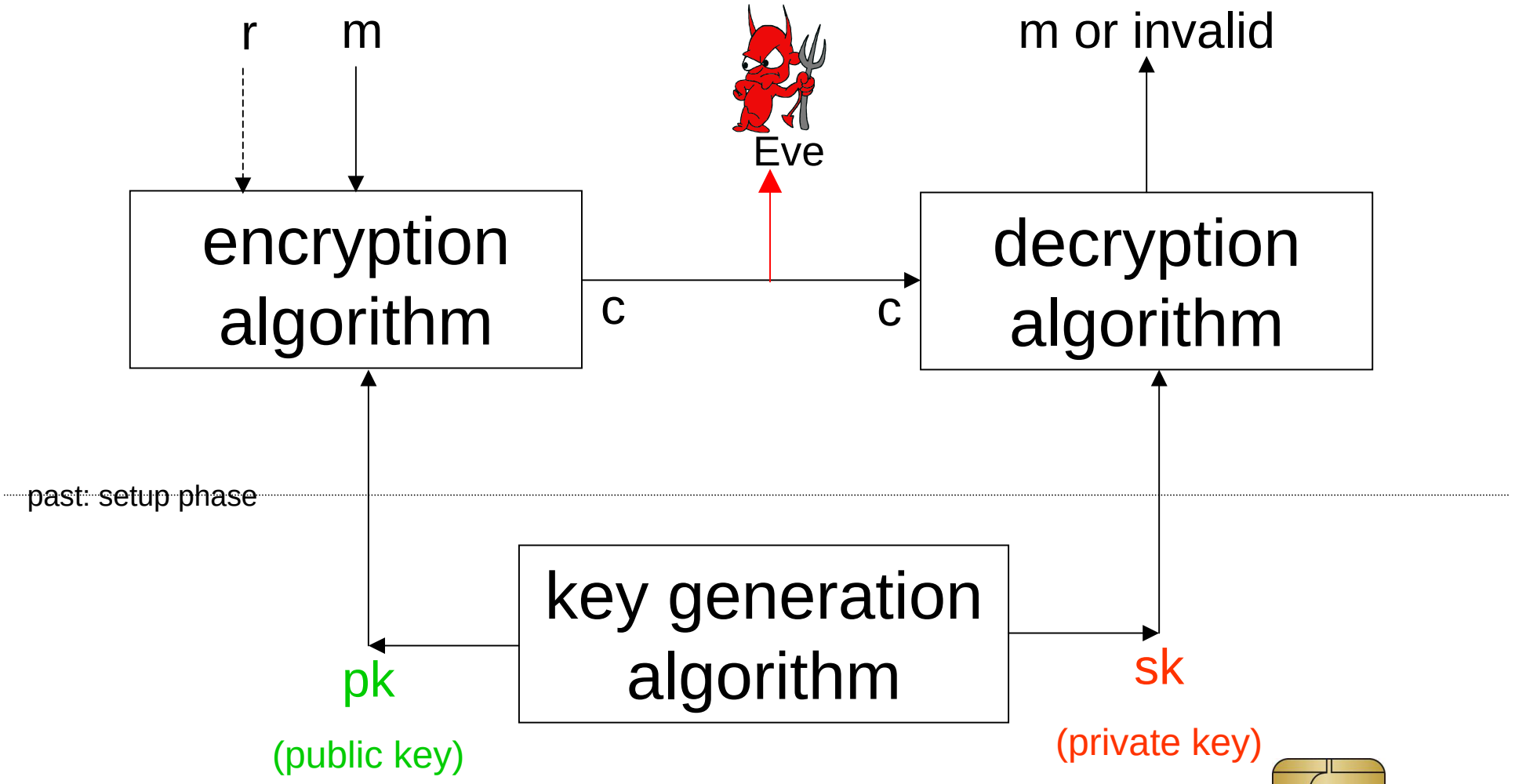
Part 3

Basic Usages of PK Encryption

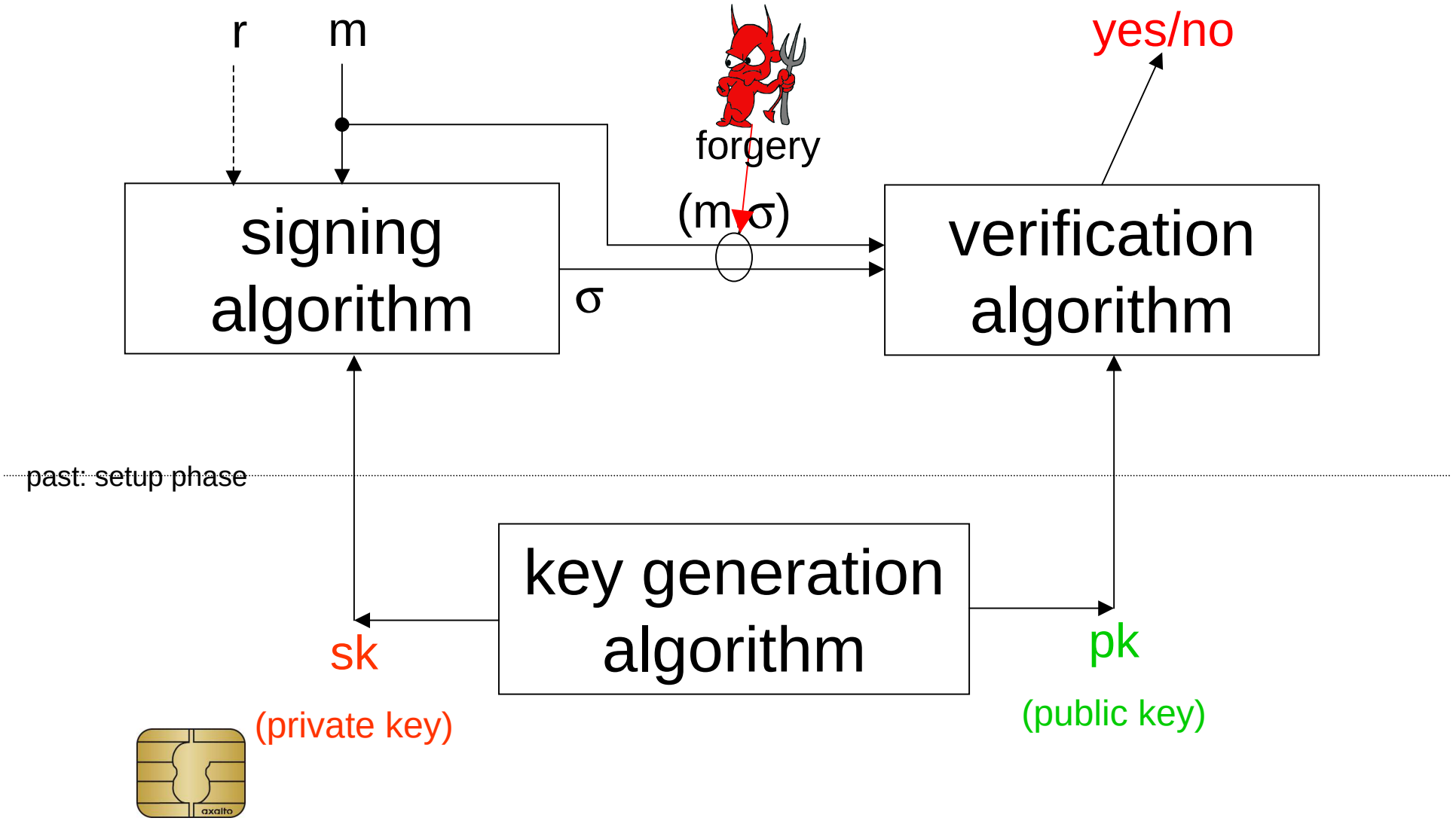
Design and Levels of Abstraction in PK Crypto



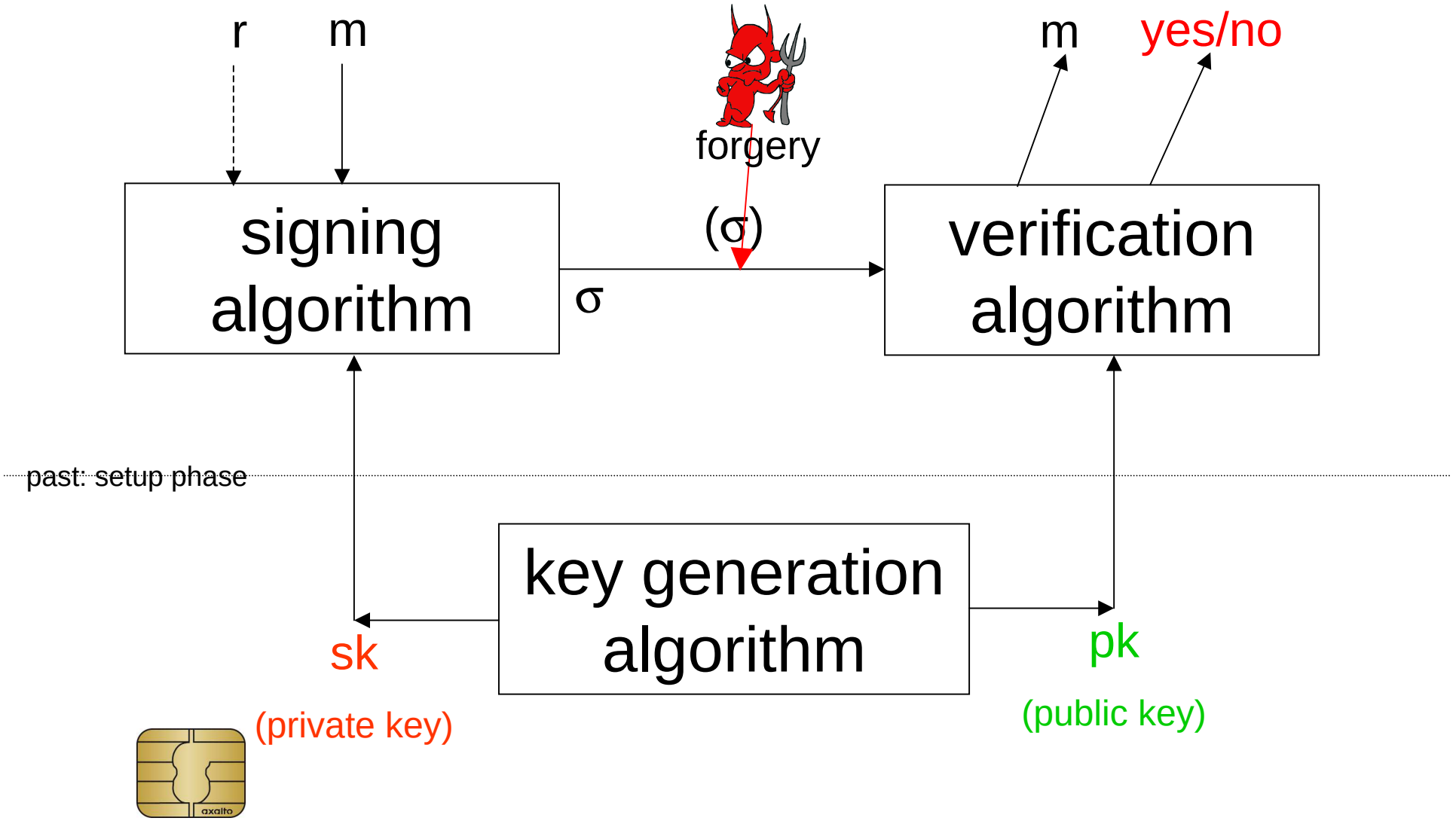
*Public Key Schemes (Encryption)



Public Key Schemes (Signature)



***PK Schemes (Signature with Message Recovery)



Fact:

Public Key Schemes are slow and work on messages of a fixed size.

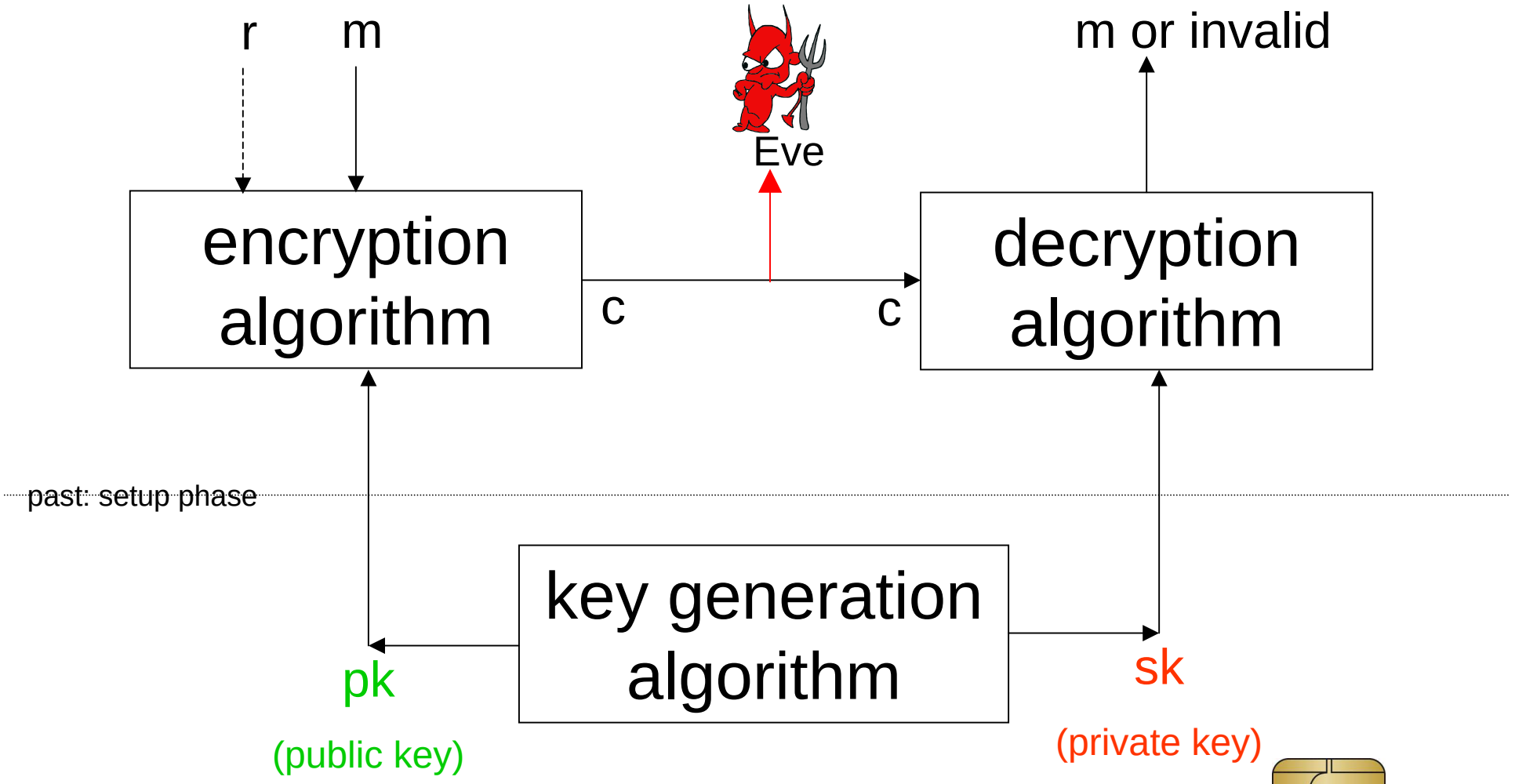
Goal: work for message of arbitrary size.

Solution:

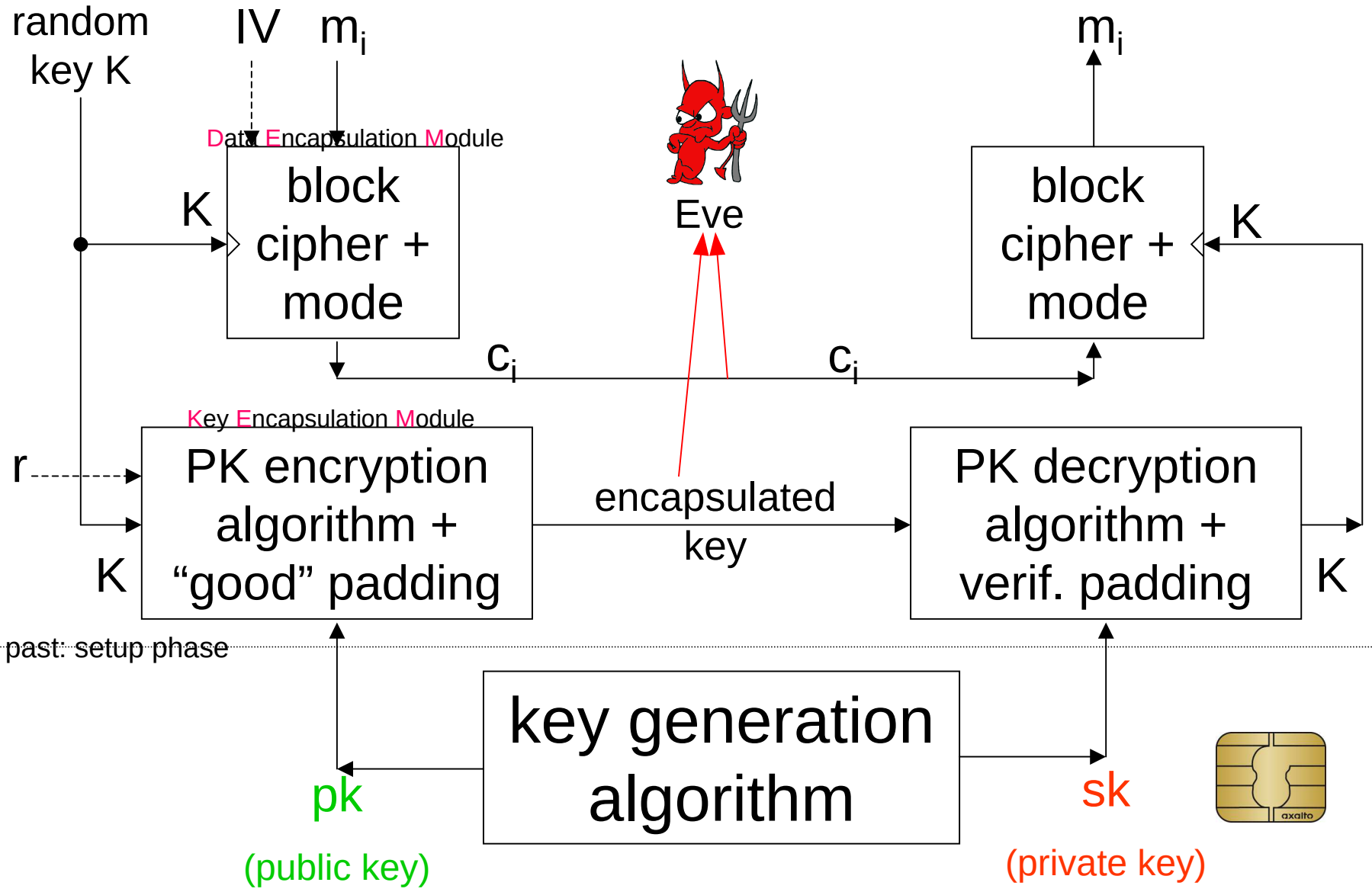
- For signatures => hash then sign paradigm.
- For encryption => hybrid encryption (e.g. as in PGP).

***Another problem: many public key encryption schemes are deterministic. Cannot be good (as explained before) => provably secure padding (later about this).

*“Textbook” Public Key Encryption --- Nobody Uses Literally:



Hybrid Encryption, PGP



Part 4

Implementation of RSA [and Rabin]

Speed of RSA

Compute x^e and reduce mod n?

- VERY BAD IDEA, will take ages.
- Instead: we use the properties of congruencies. Reduce systematically mod n to keep the data small.

=> Complexity is measured in the number of multiplications on $k=\log_2 n$ bit integers.

Cost of RSA / Rabin - Faster Cases

e=3: compute x, x^2, multiply.

Cost = 1 S +1 M. (on $k=\log_2 n$ bit integers).

Remark: squaring tends to be less expensive than x*x.

$e=2^{16}+1=65537$: compute x, x^2, x^4, x^8, …, $x^{2\wedge 16}$, multiply.

Cost = 16 S +1 M.

e=2:

Cost = 1 S. [Rabin encryption / Rabin signature verif.]

General Case

- e=full size, $k=\log_2 n$ bits.
- RSA decryption, d=full size.
- Rabin decryption: √ done mod p and mod q. Half size.

} about 6-8x speed-up with CRT possible

SQUARE AND MULTIPLY method:

Let $e = \Sigma_{i=1} 2^k e_i$ be the binary expansion of e.

- Compute $x, x^2, x^4, x^8, \ldots, x^{2\wedge(k-1)}$,
- Multiply all those for which $e_i=1$.

Cost: about k S +k/2 M on average.

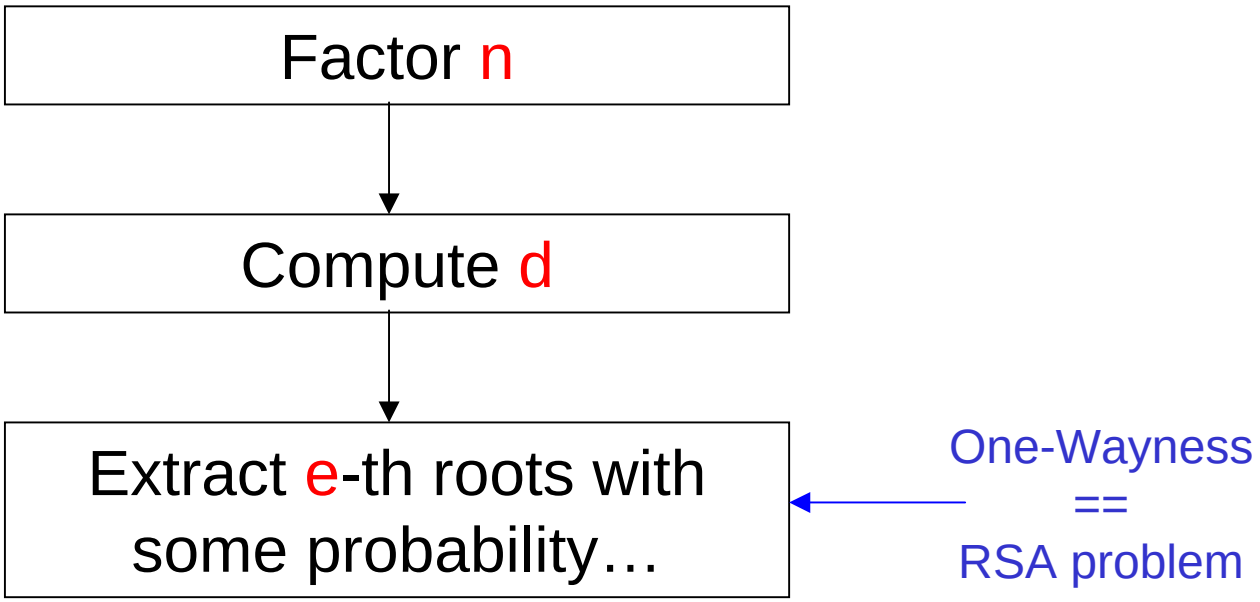
Cost of one modular multiplication: $O(k^2)$.

Cost of one multiplication: "nearly" linear in k [complex modern methods with FFT and careful implementation, cf. open-source GNU GMP library].

Part 5

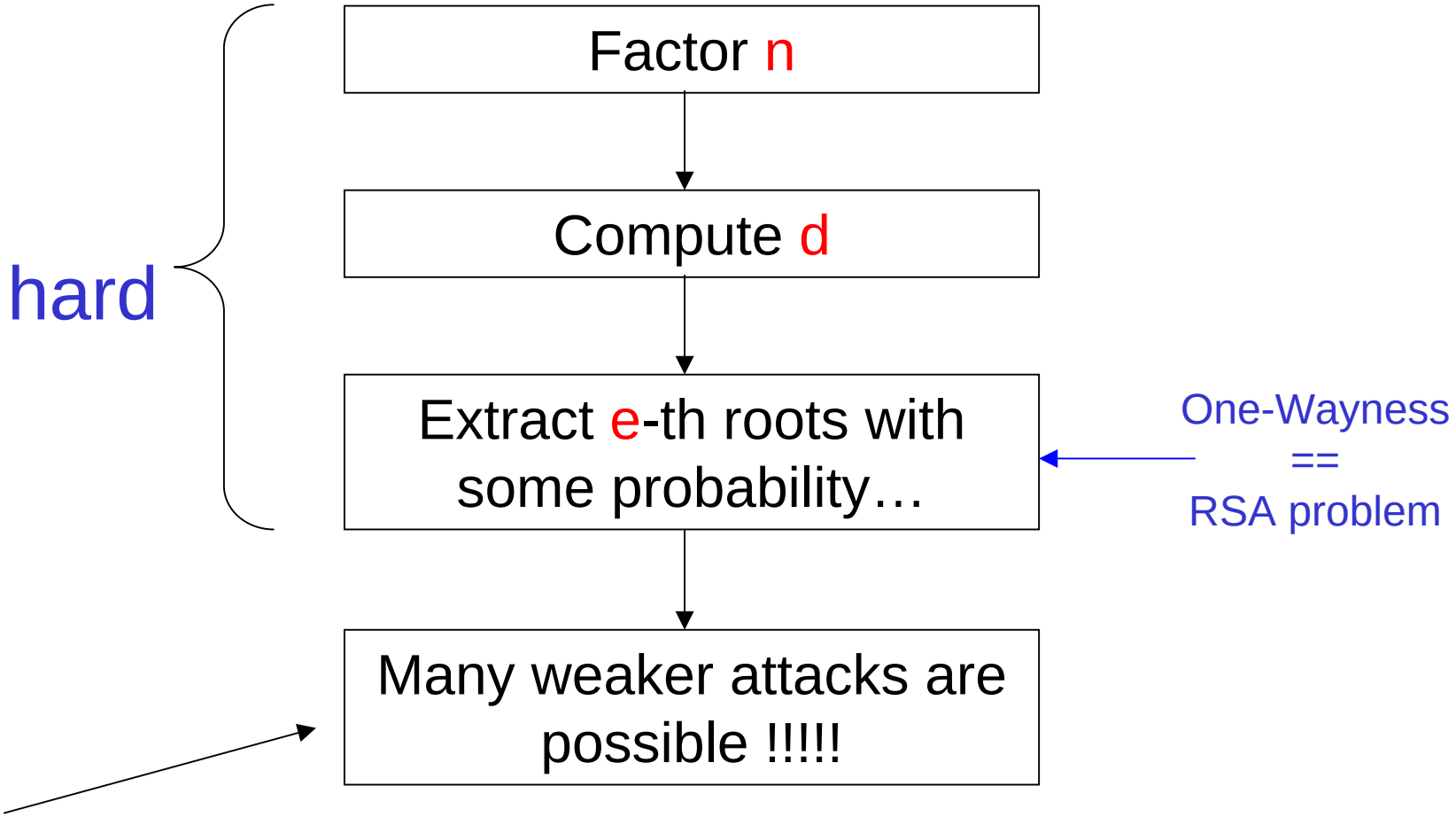
Security and Weak Points of RSA

Factoring vs. RSA



Maybe RSA is really weaker than factoring – not sure, so far there is no result in this direction.

Factoring vs. RSA



Doesn’t mean we can break RSA…

Textbook RSA – Attack 1

Malleability based on the Homomorphic Property:

$$E_{pk}(ab) = E_{pk}(a) * E_{pk}(b).$$

- Doesn't contradict One-Wayness of RSA.
- Means that RSA have to be used very carefully !!!

Textbook RSA – Attack 2

Interpolation Attack. Assume that $e=3$ and the same message m is encrypted with 3 different RSA moduli of 1024 bits each:

$$\left.\begin{aligned} c_1 &\equiv m^3 \bmod n_1 \\ c_2 &\equiv m^3 \bmod n_2 \\ c_3 &\equiv m^3 \bmod n_3 \end{aligned}\right\}$$

Attack: by CRT we compute:

$$C \equiv m^3 \bmod n_1 n_2 n_3$$

- Remark that m^3 has less than 3072 bits.
- So $C=m^3$ in $\mathbb{Z}$. So we can compute m as $m=\sqrt[3]{C}$.

***Textbook RSA – Attack 3

Alice and Bob use the same n
but different e and e'.

Attack 1: Alice can decrypt messages sent to Bob. (assuming knows d => the factors…).

Attack 2: If the same message is sent to Alice and Bob and GCD(e,e')=1, anybody can decrypt it.

Textbook RSA – Attack 2395…

Etc.

These mean that RSA
have to be used very carefully (!)

Happily, we know how (!).

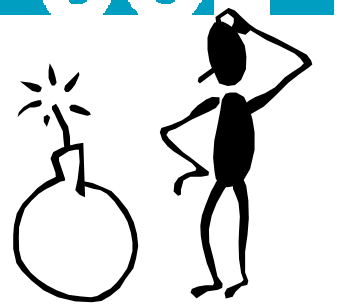
Part 6

Secure Encryption with RSA

Textbook RSA

- Never use it.
- Needed:
  - RSA textbook+"good" randomised padding.
- **next slide: Warning: many international standards are insecure. Some are broken. Other just lack security proof.

*Modern Cryptography:



Serious Issues:

- Only in about 1998 people understood what is a secure public key encryption system.
- Only in about 2001 people did understood how to use RSA properly.
- Most standards (e.g. ISO) are not secure.

Minefield:

Try-and-error just did not work…

Requirements

- Heuristic requirements:
  - Randomised encryption:
    - all values that go into RSA should be randomised (different at each encryption) and pseudo-random (should “look” and behave as random numbers, we only assume that RSA is OW, i.e. the RSA problem is hard, this assures security only for “random” x...).
  - It should make use of algebraic homomorphic properties of RSA impossible to do any attack (non-malleability etc..).
- Exact requirements, security proof etc.
  - See Crypto 2 course.
    - Technical: Random Oracle Assumption vs. Standard Model security.
- We study just few examples that can be directly applied in practice.
  - Encryption: today.
  - Signature: next week.

Bellare-Rogaway 1993

A first known secure method to encrypt with RSA.

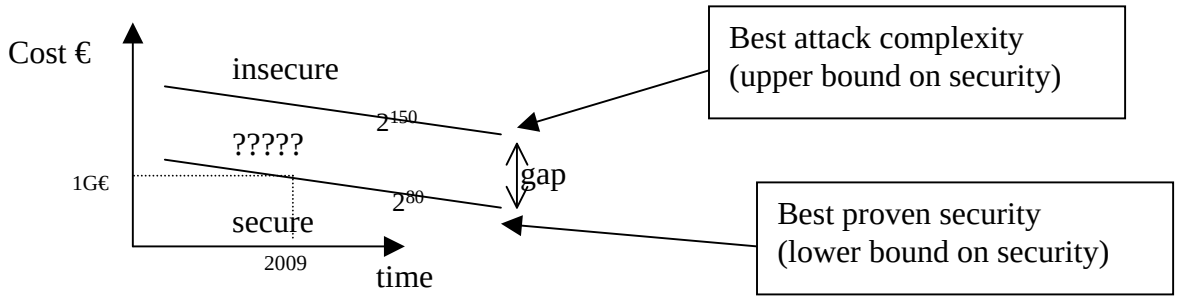
⇒ Better understood recently.

⇒ Has a "tight" security proof: which means that security is exactly equal to the security of RSA. Which means that as long as RSA 1024 is not broken, this method can be used with RSA 1024 bits.

- ⇒ With some other methods [e.g. RSA-OAEP proven secure by Stern et al.], one needs to use RSA 4096 bits today because only then the security is proven… This works well, it is just much slower in practical implementation.

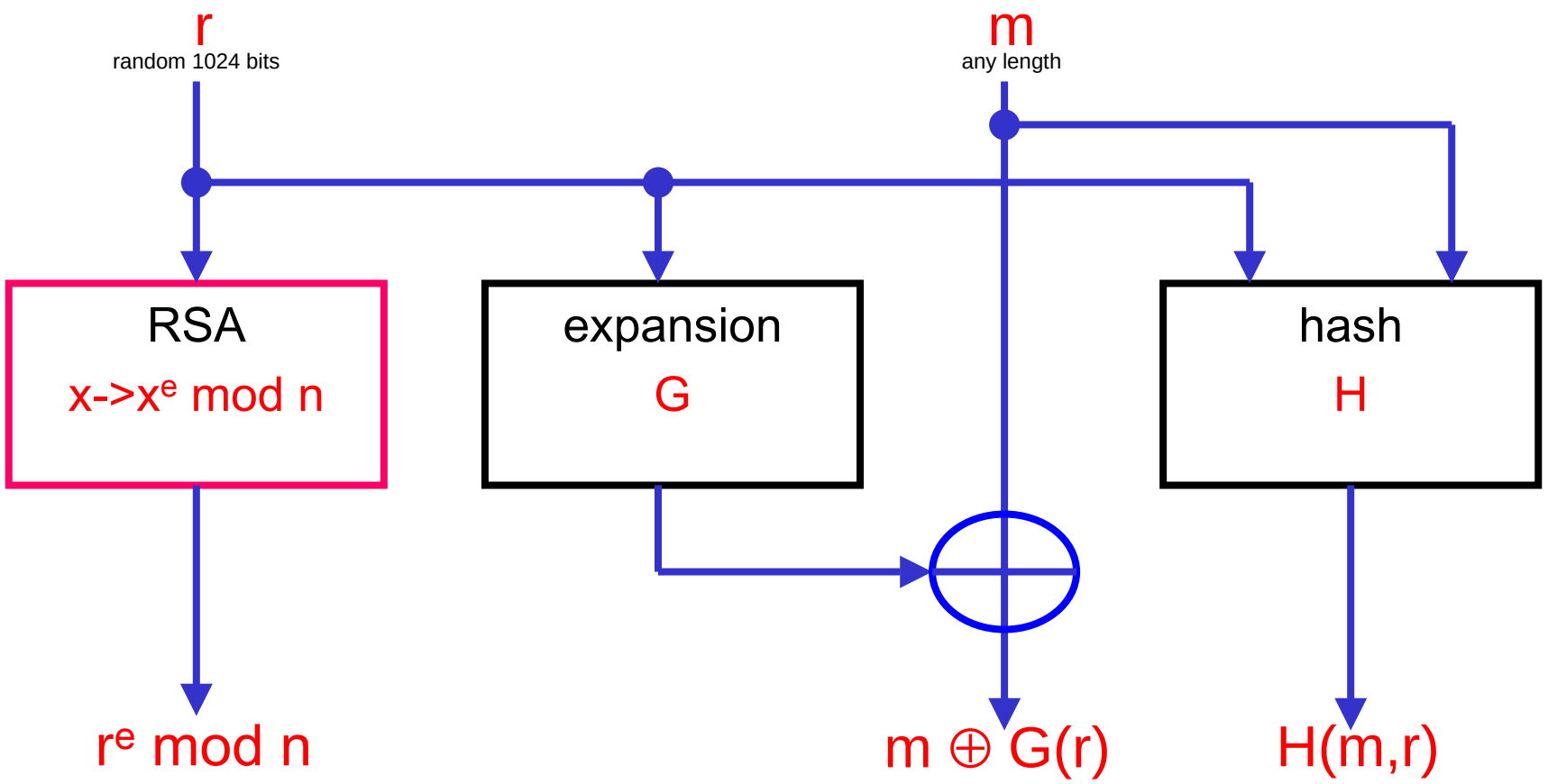
⇒ ☹ Security proof works [cf. Pointcheval slides] only for trapdoor OW permutations: we only know RSA.

Security Proofs and Attacks



Bellare-Rogaway [BR'93]

Uses a hash function H and an expansion function (PRNG or a synchronous stream cipher) G. Can encrypt arbitrary length messages !



Bellare-Rogaway [BR'93]

Decryption:

- obvious,
- must check if H(m,r) is correct.
  - Implementation tip: If incorrect, do not raise an exception, or output a specific error message, better to output a random message !
  - Do not allow timing attacks either.

Bellare-Rogaway [BR'93]

Security: Semantically Secure [and Non-Malleable] under the most general Adaptively Chosen Ciphertext attack scenario.

Technical name: IND-CCA2 = "Maximum security".

- Strong secrecy that works even in the presence of internal attackers and even if the attacker has a lot of a priori knowledge on the messages.