Introduction to Symmetric Cryptography

María Naya-Plasencia

Introduction to Symmetric Cryptography

María Naya-Plasencia
Inria, France

Summer School on real-world crypto and privacy
Šibenik, Croatia - June 11 2018

Outline

- Introduction
- One Time pad - Stream Ciphers
- Block Ciphers - Operation Modes
- Hash function
- Symmetric Cryptanalysis: Foundation of Trust
- Differential (and Linear) Cryptanalysis
- New Directions

Symmetric Cryptography

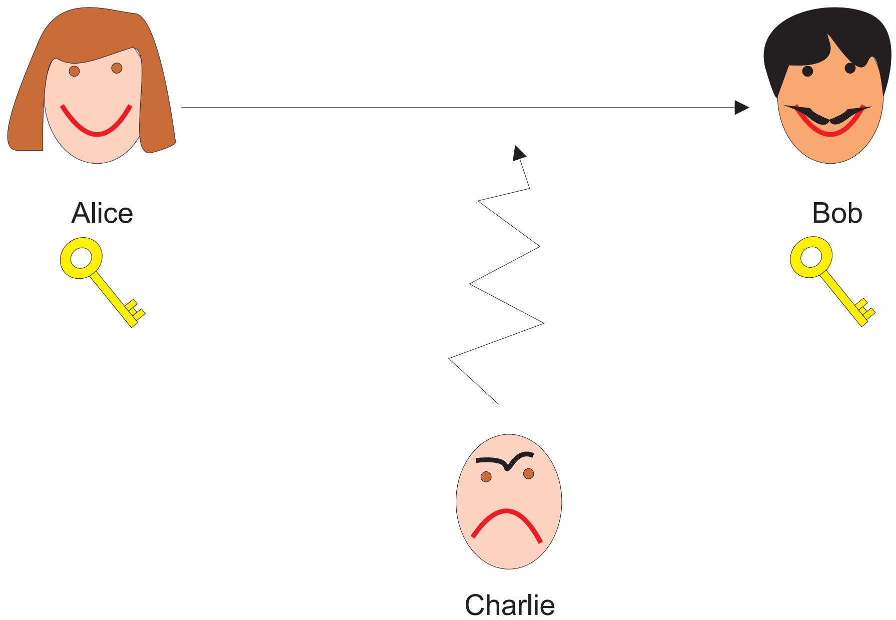
Cryptography

- Cryptography : hiding/protecting information against malicious adversaries.

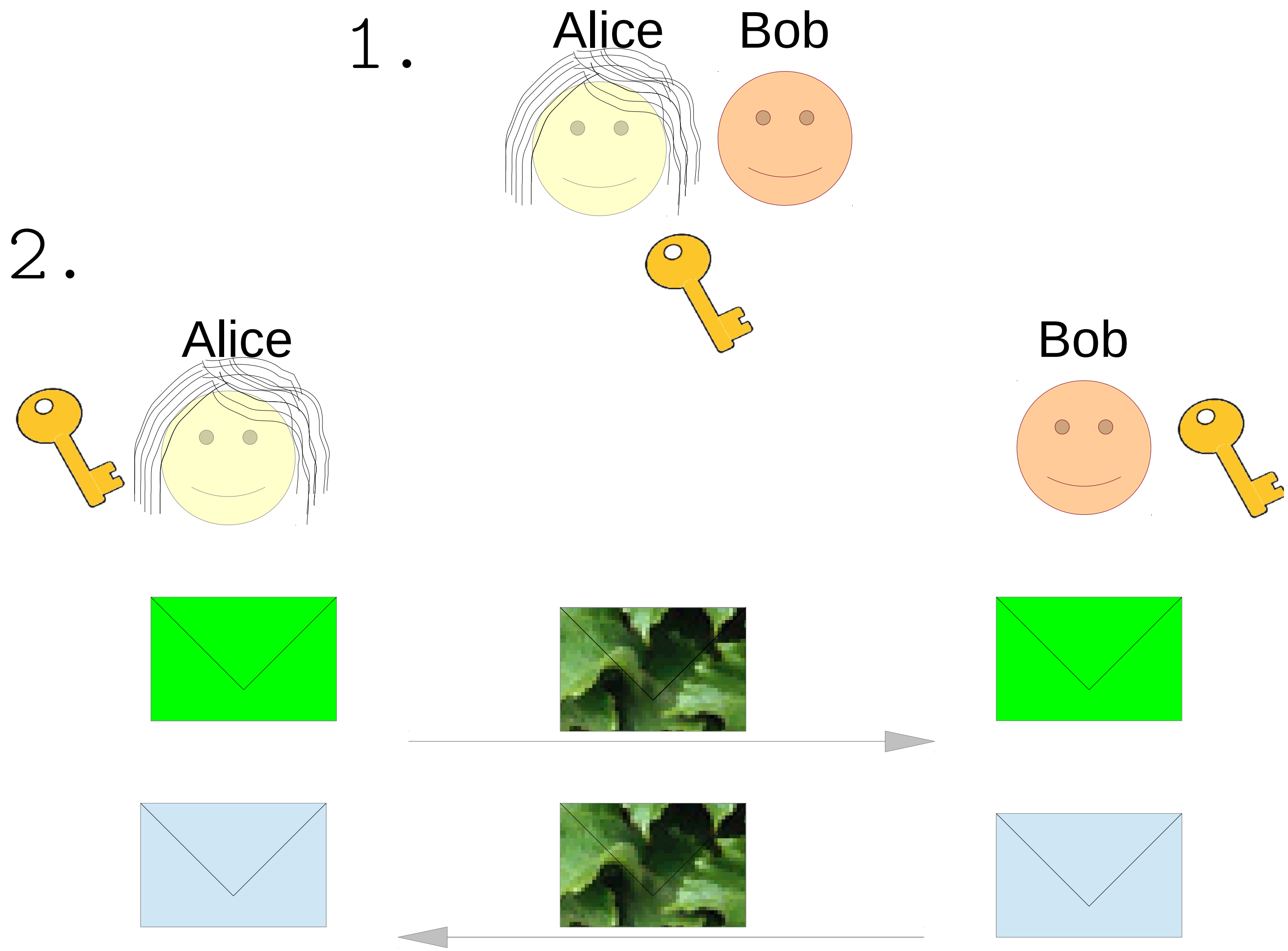
- Main aims:
  Confidentiality $\Rightarrow$ usually with the help of a key
  Authentication
  Integrity
  ...

Cryptography - Encryption

Symmetric encryption and Asymmetric encryption

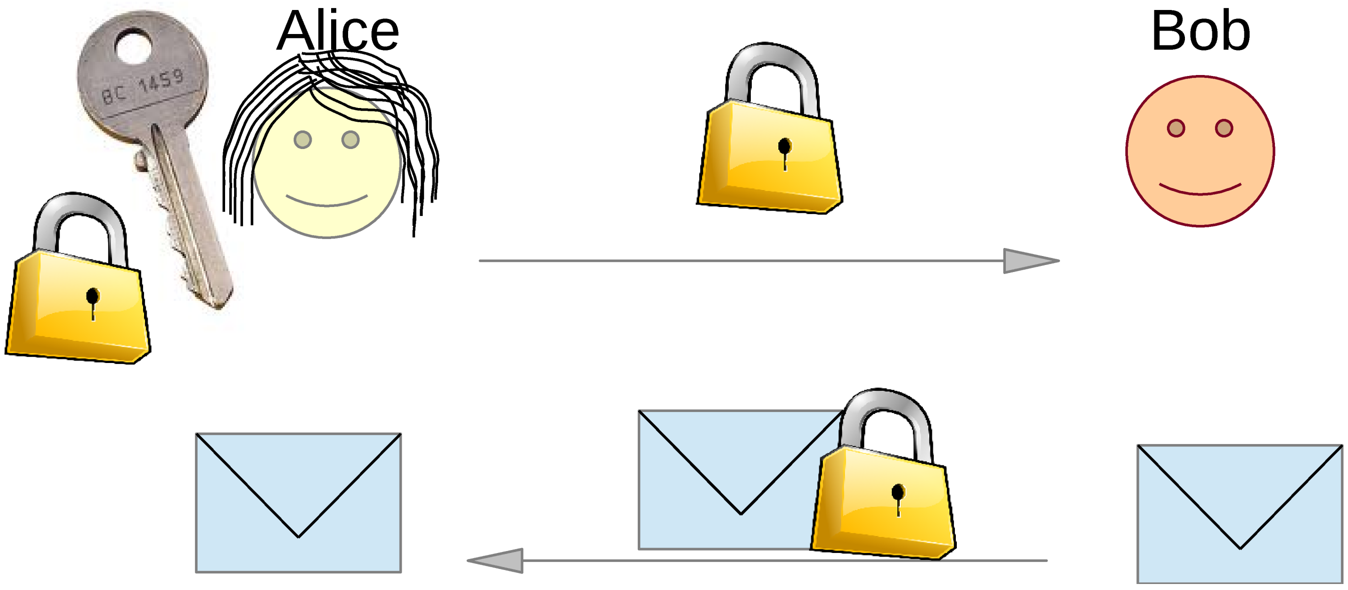


Symmetric Cryptography



Asymmetric Cryptography

Without needing a previous meeting:



Asymmetric vs Symmetric Cryptography

Asymmetric:

- Advantage: No need of key exchange.
- Disadvantage: Computationally costly.

Symmetric:

- Disadvantage: Need of key exchange.
- Advantage: Performant, adapted to constrained environments.

$\Rightarrow$ Use asymmetric for key exchange, and next use symmetric!!.

Security of Encryption Algorithms

Asymmetric (e.g. RSA) *(no key exchange/computationally costly)*

Security based on well-known hard mathematical problems (e.g. factorization).

Symmetric (e.g. AES) *(key exchange needed/efficient)*

Ideal security defined by generic attacks.

Need of continuous security evaluation (cryptanalysis).

Generic Attacks on Ciphers

- Security provided by an ideal cipher defined by the best generic attack:
  exhaustive search for the key in $2^{|K|}$.

- Recovering the key from a secure cipher must be infeasible:

  $\Rightarrow$ typical key sizes $|K| = 128$ to 256 bits.

Cryptanalysis

In general:

A primitive is considered secure as long as no attack better than generic attacks on it is found.

Cryptanalysis: looking for these other attacks.

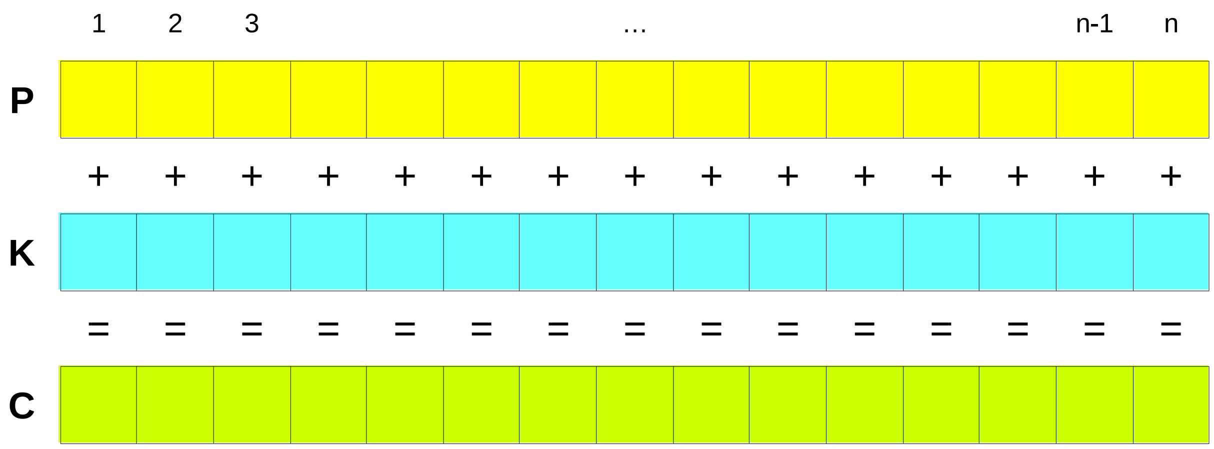
(we will see more about this later)

One Time Pad & Stream Ciphers

One Time Pad

- One Time Pad: provides perfect secrecy.

With a completly random key K



$\Rightarrow$ all C are equally likely,

but needs a secret key as long as the message!!

OTP with shorter keys?

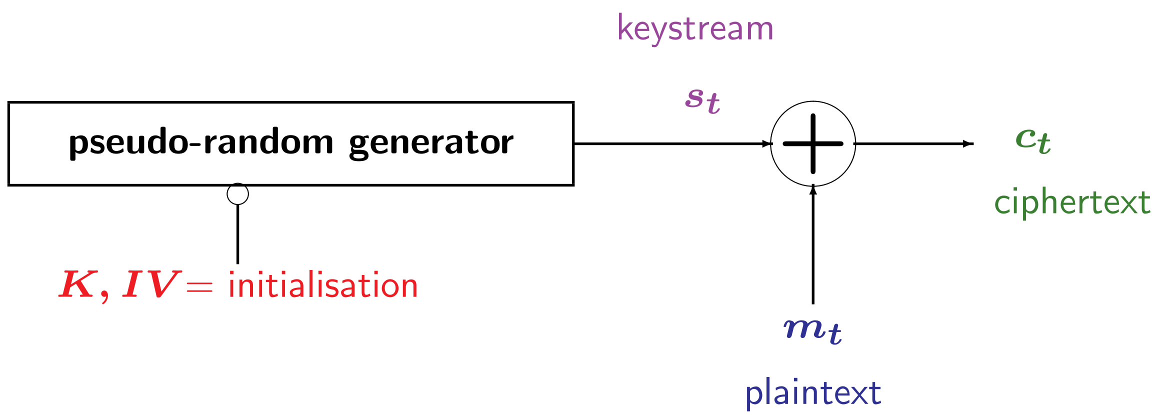
Solution:

- From a shorter secret seed k, generate a "long" sequence (keystream) indistinguishable from random if we don't have the seed k

Stream Ciphers

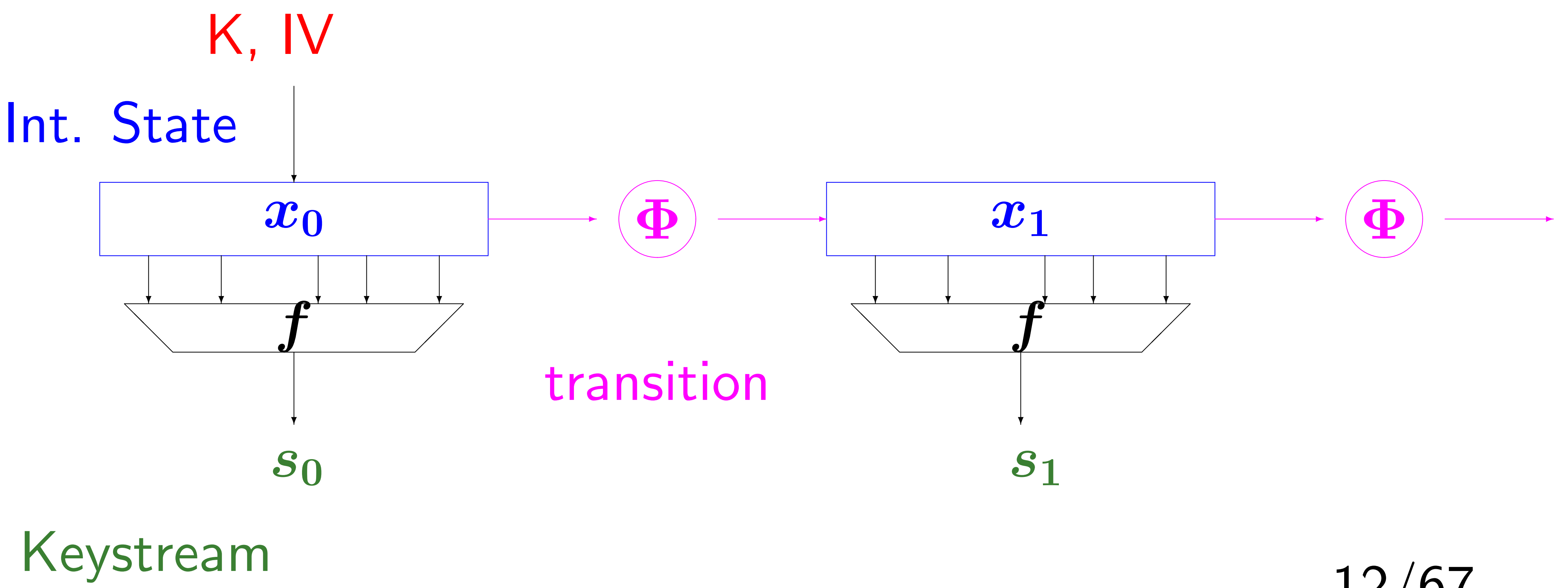
In practice: the keystream is obtained from pseudo-random generators.

Additive stream cipher:

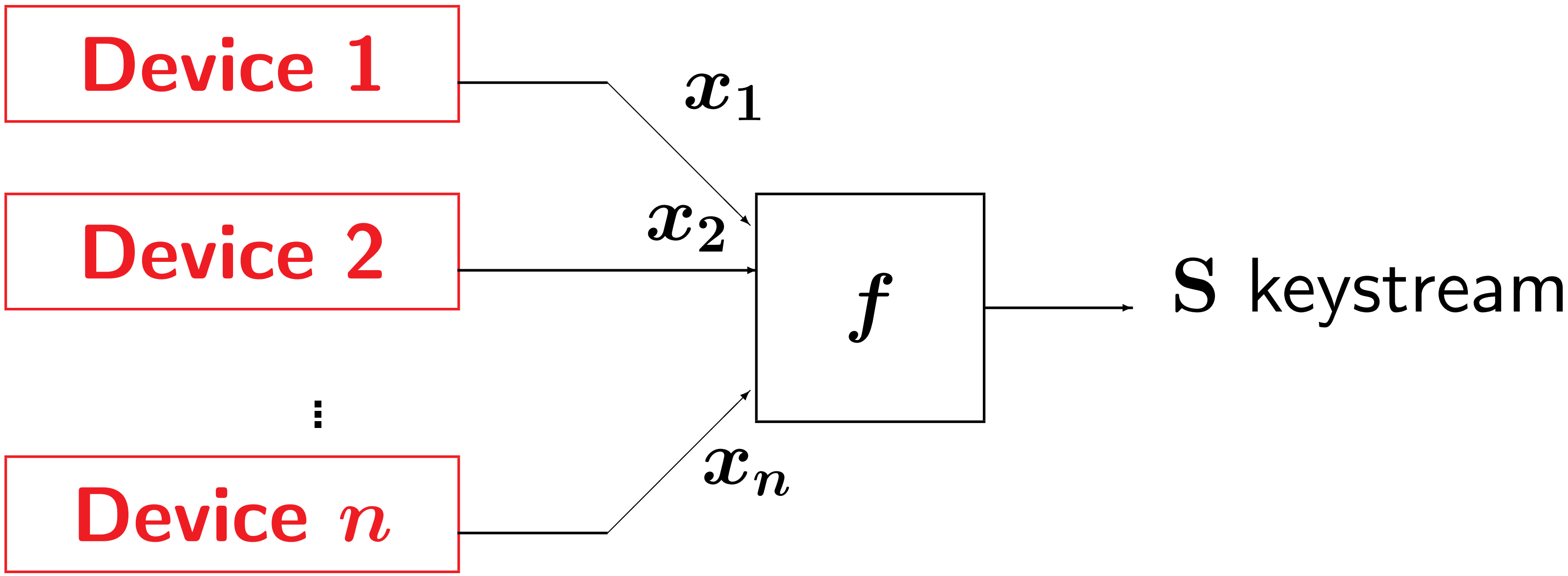


Stream Ciphers

Initialisation, transition, extraction:



Ex: Combination generators



where each x_i has period T_i.

eSTREAM project

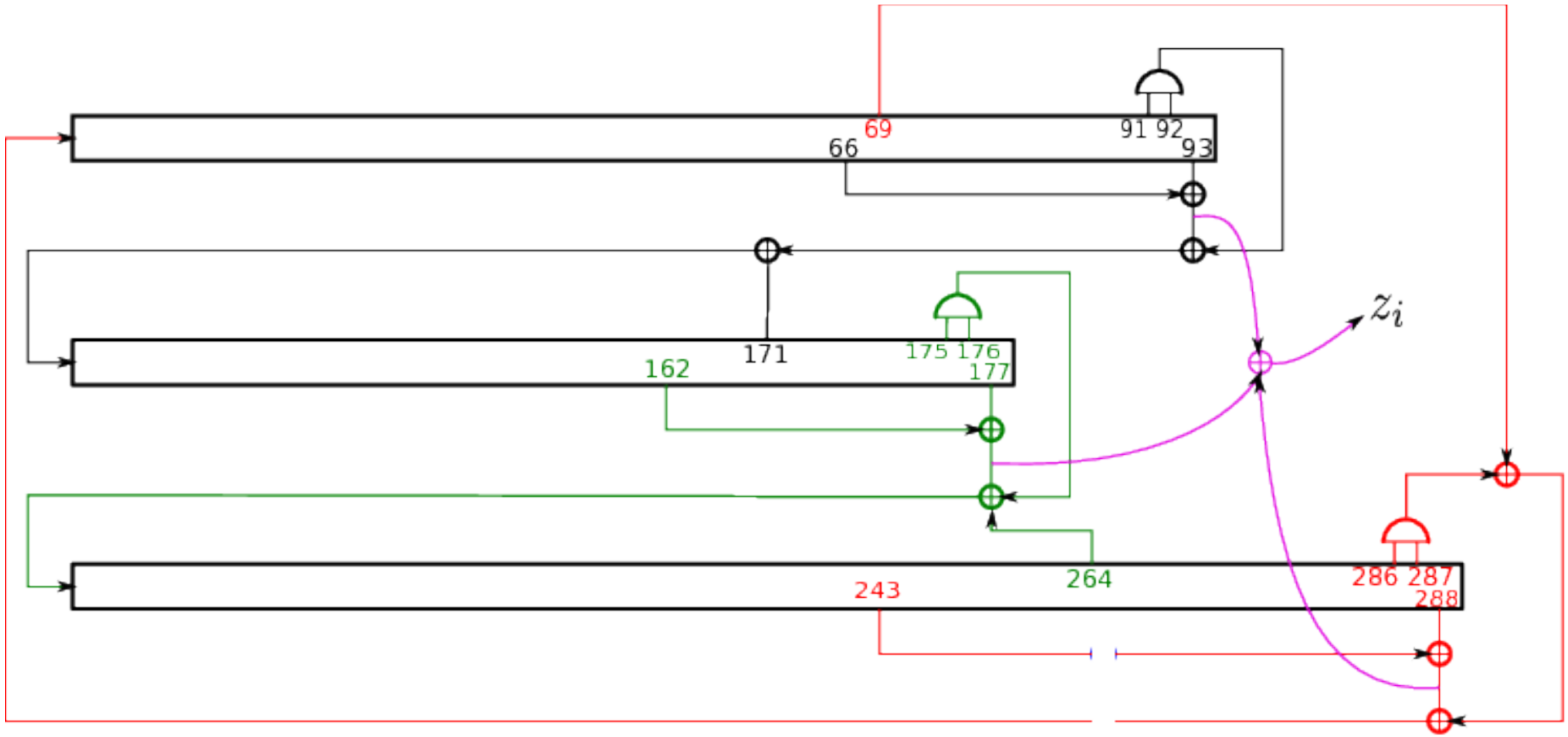
After Nessie's failure:

- Launched by European network ECRYPT 2005-08
- Conception of new dedicated stream ciphers
- 37 submitted algorithms
- 8 in final portfolio, only 6 unbroken now...

Seems difficult - how could it be easier? $\Rightarrow$ Block ciphers

Ex. Trivium (eSTREAM portfolio)

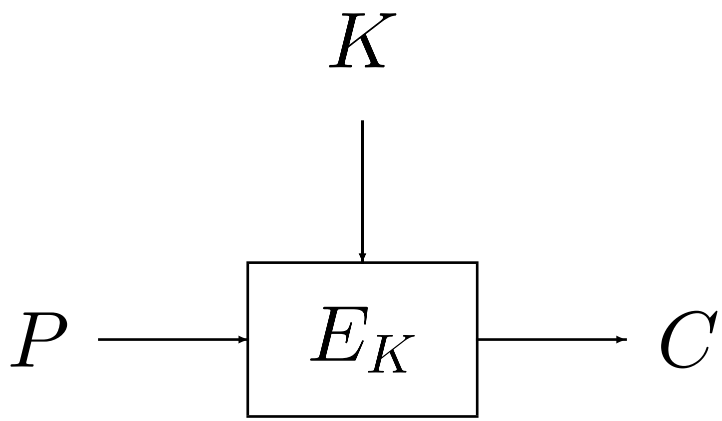
80 bit key and IV, 288 bit state [DC-P'06].



Block Ciphers

Block ciphers

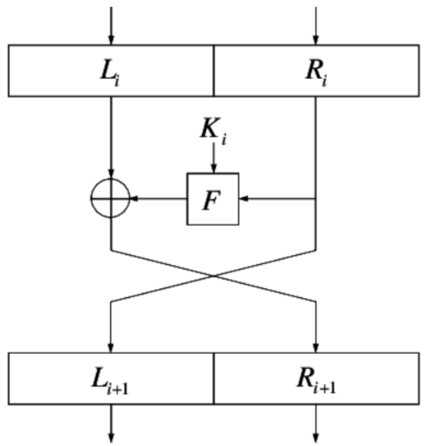
Message decomposed into blocks, each transformed by the same function E_K.



E_K is composed of a round transform repeated through several similar rounds.

Block ciphers - Two main families

- Feistel constructions:



- SPN constructions: transform the whole state:
  - Substitution layer (S-boxes, non-linear)
  - Permutation layer typically $\oplus$ and/or rotations.
  - Subkey addition.

Block ciphers

- Key schedule: generates subkeys for each round from the secret key.

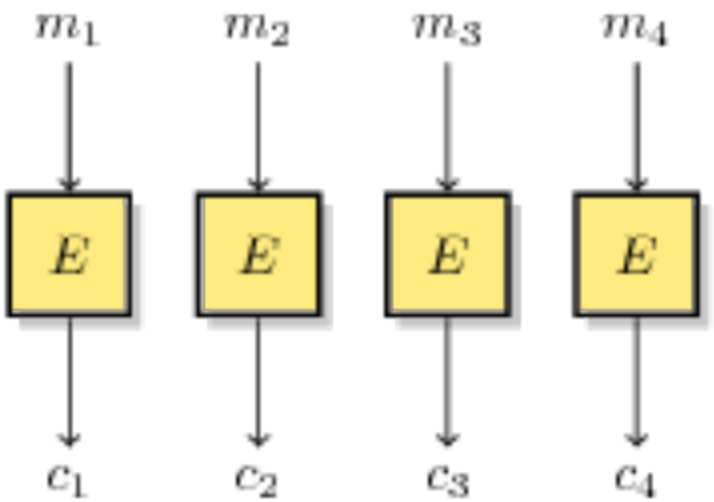
- A block cipher is a family of permutations parametrized by the key.

What to do when:

- Longer messages than a block?
- Several messages?

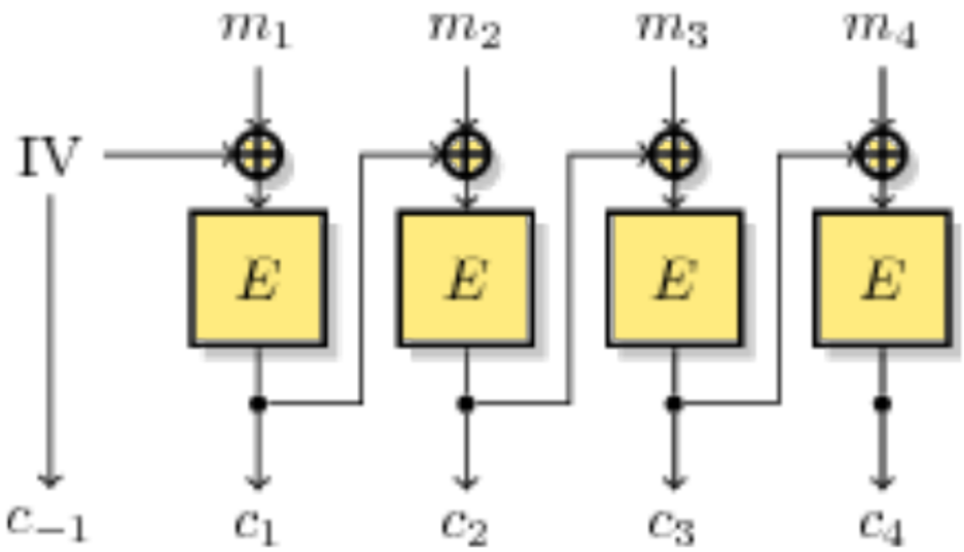
$\Rightarrow$ Operation modes

Operation Modes: ECB



- Problem: equal Ptxts generate equal Ctxts

Operation Modes: CBC [EMST'76]



- Proven secure if the block cipher is secure and if the key is changed after $\ll 2^{n/2}$ encryptions.

Interlude: birthday paradox

Birthday Paradox

- "In a room with 23 people, there is a 50% chance of having two colliding dates of birthday".

Intuitive explanation:
23 people $\Rightarrow \frac{23 \times 22}{2}$ pairs.

With $2^{n/2}$ elements we can build about 2^n pairs (so we have a good chance of finding a collision).

Back to modes

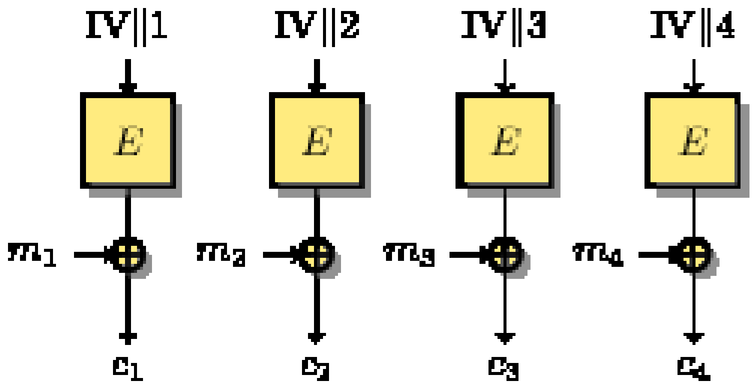
CBC: Careful with Recommendations

Sweet-32 attack [BL'16], based on finding a collision in the internal state:

For ciphers of 64 bits, we can find a collision in about 2^{32} encrypted blocks, and recover the plaintext.

Possible because the security recommendations were not respected.

Operation Modes: CTR[DH' 79]



- Proven secure if the block cipher is secure and if the key is changed after $\ll 2^{n/2}$ encryptions (missing difference attack otherwise [LS18]).

AES

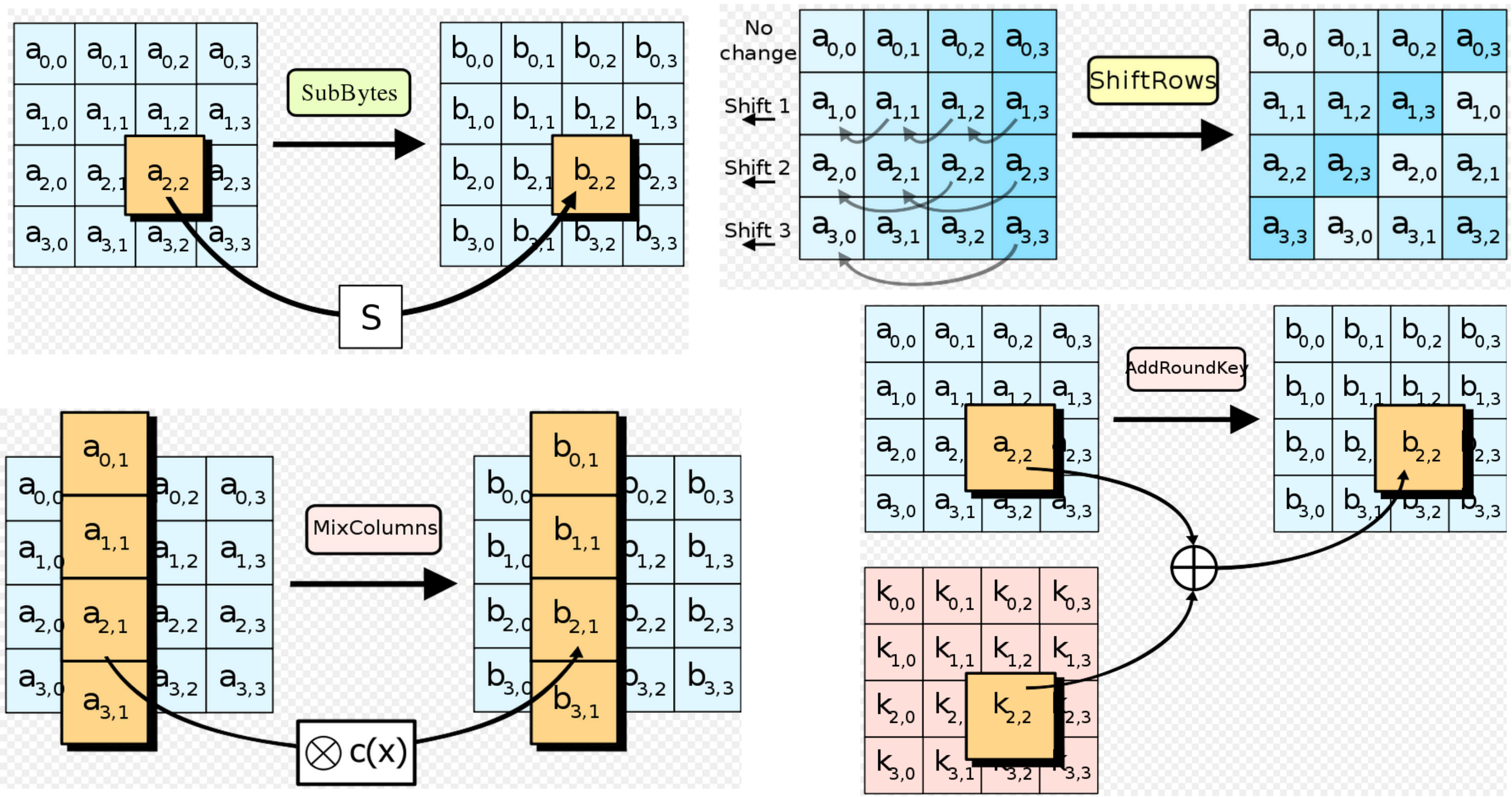
AES Competition and Winner

Launched by NIST to find a succesor of DES 97-00.
15 submissions, 1 winner: Rijndael [Daemen-Rijmen 97]

AES:

- SPN cipher.
- 10/12/14 rounds for 128/192/256-bit keys.
- Block of 128 bits.

AES Round Function



Images from http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Authenticated Encryption

AE

In order to provide confidentiality and authenticity:

- Authenticated encryption:
- Caesar competition finished this year.
- See next talk by Thomas Shrimpton

Hash Functions

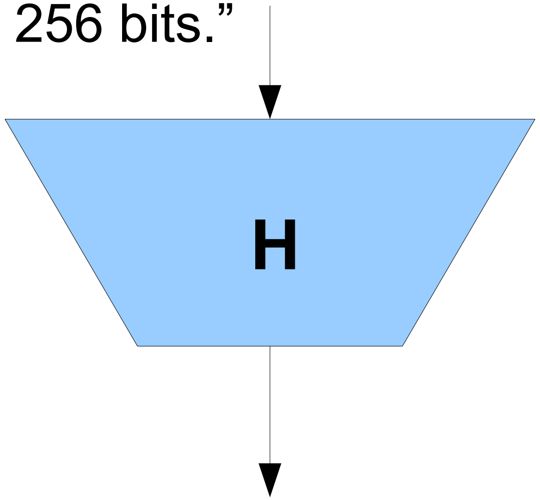
Cryptographic Hash Functions

$$\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^{\ell_h}$$

- Given a message of arbitrary length returns a short 'random-looking' value of fixed length.
- Many applications: MAC's (authentification), digital signatures, integrity check of executables, pseudorandom generation...

Cryptographic Hash Functions

"Here we introduce any message that we want to hash. We will then obtain a fingerprint of the message, a random looking value that will identify it. In this case, 256 bits."

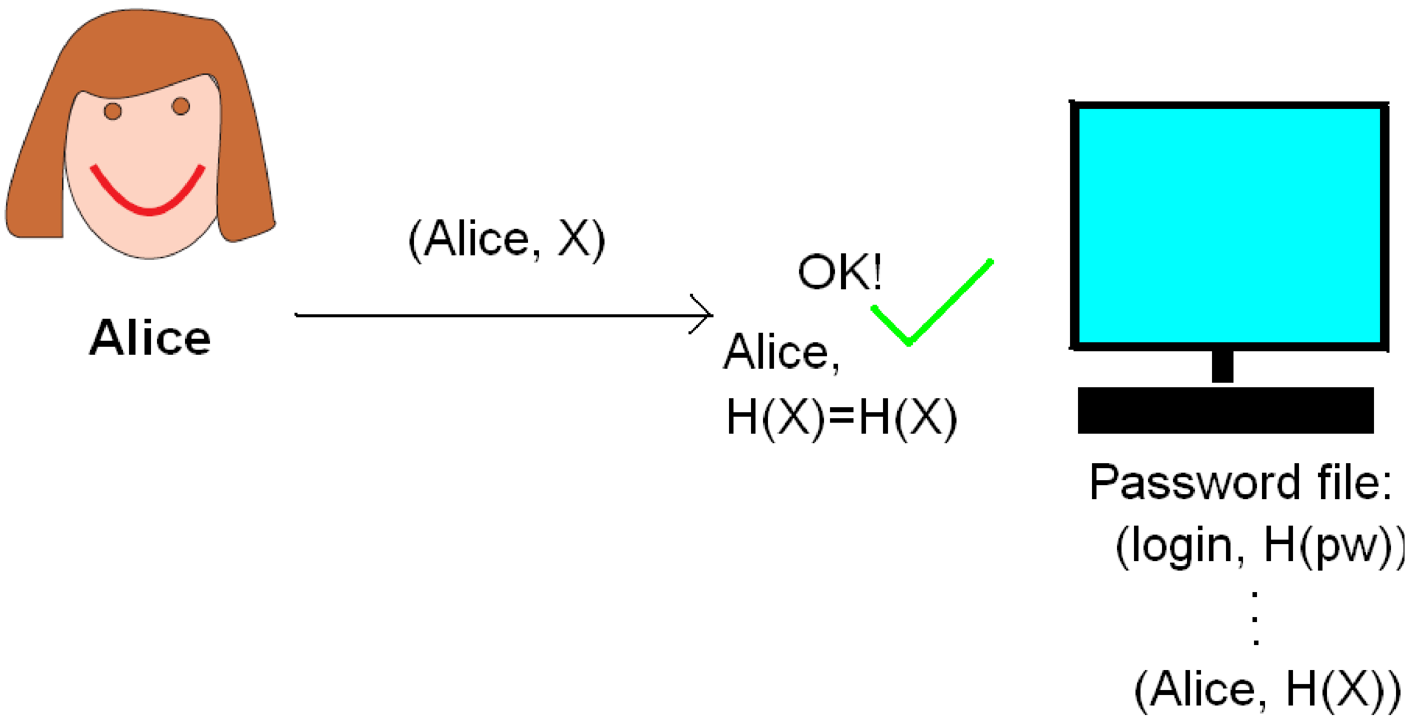


H is easy to compute

**"A4F567BCA61234FA
987DF45F6C7A3B22
BA5BCD6784857DBF
46F5D4A8CD327345"**

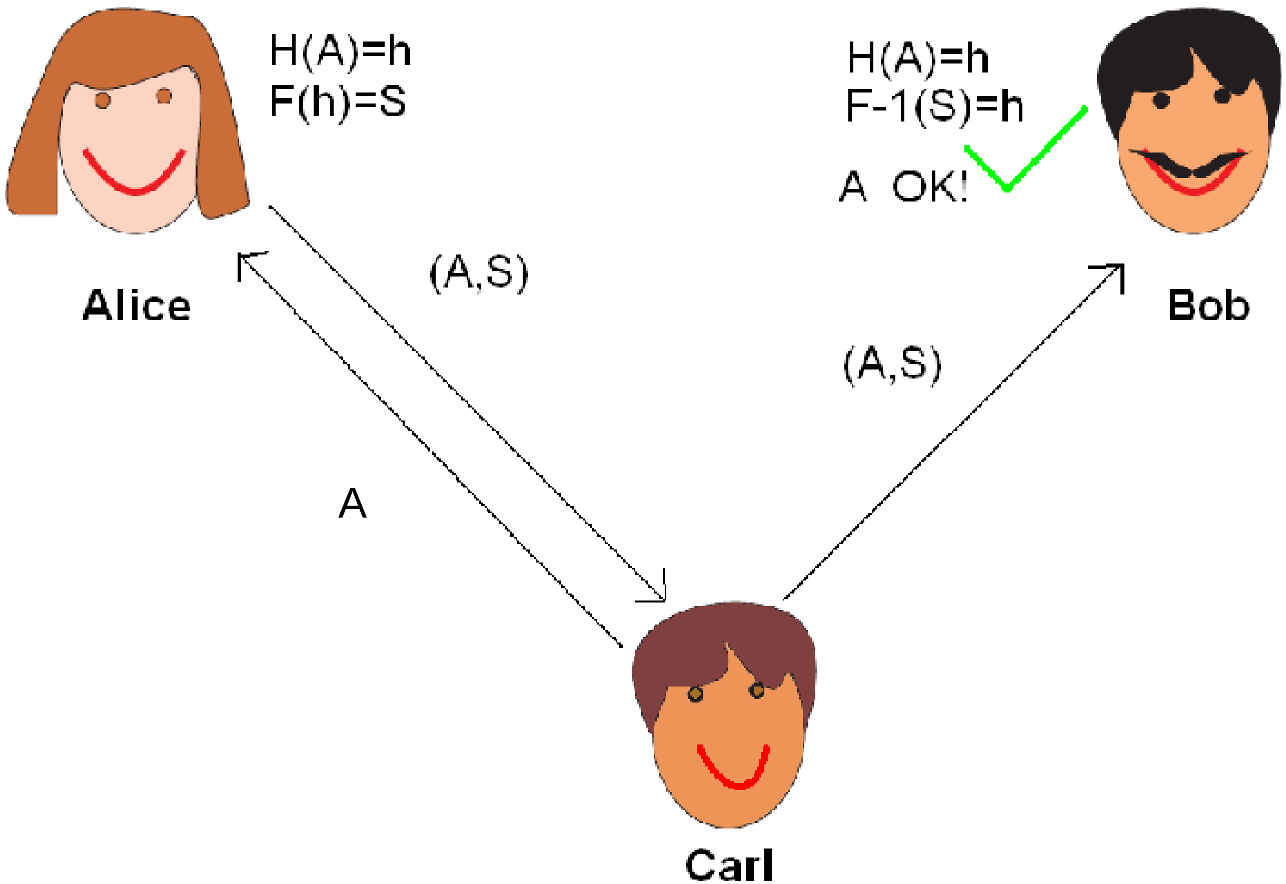
Hash Functions applications

Autentication:



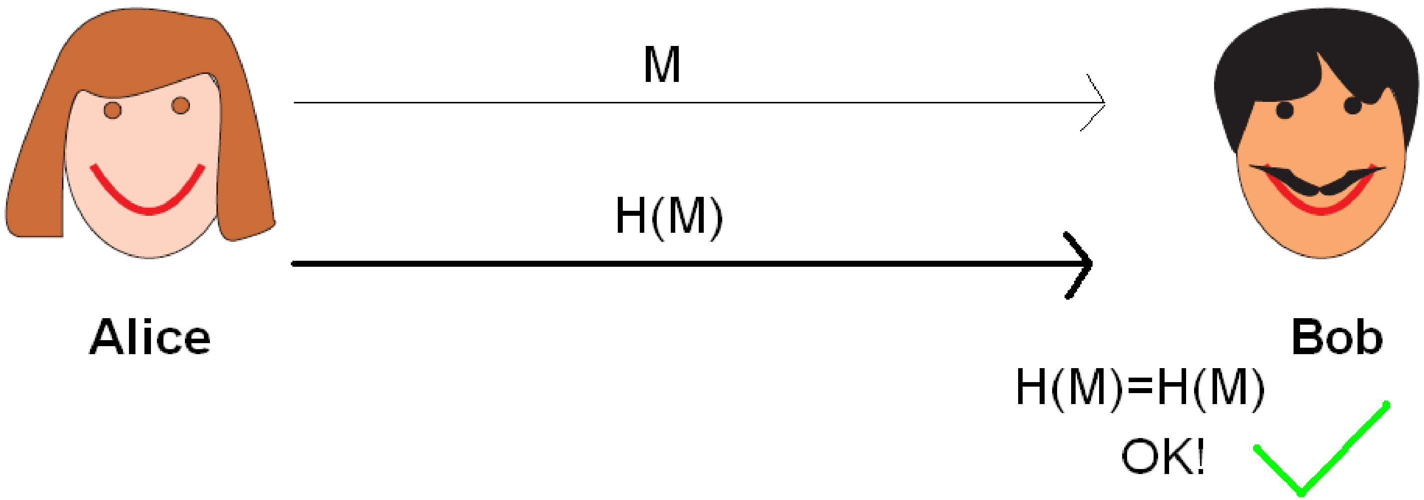
Hash Functions applications

Digital signature:



Hash Functions applications

Verifying the integrity:



Security requirements of hash functions

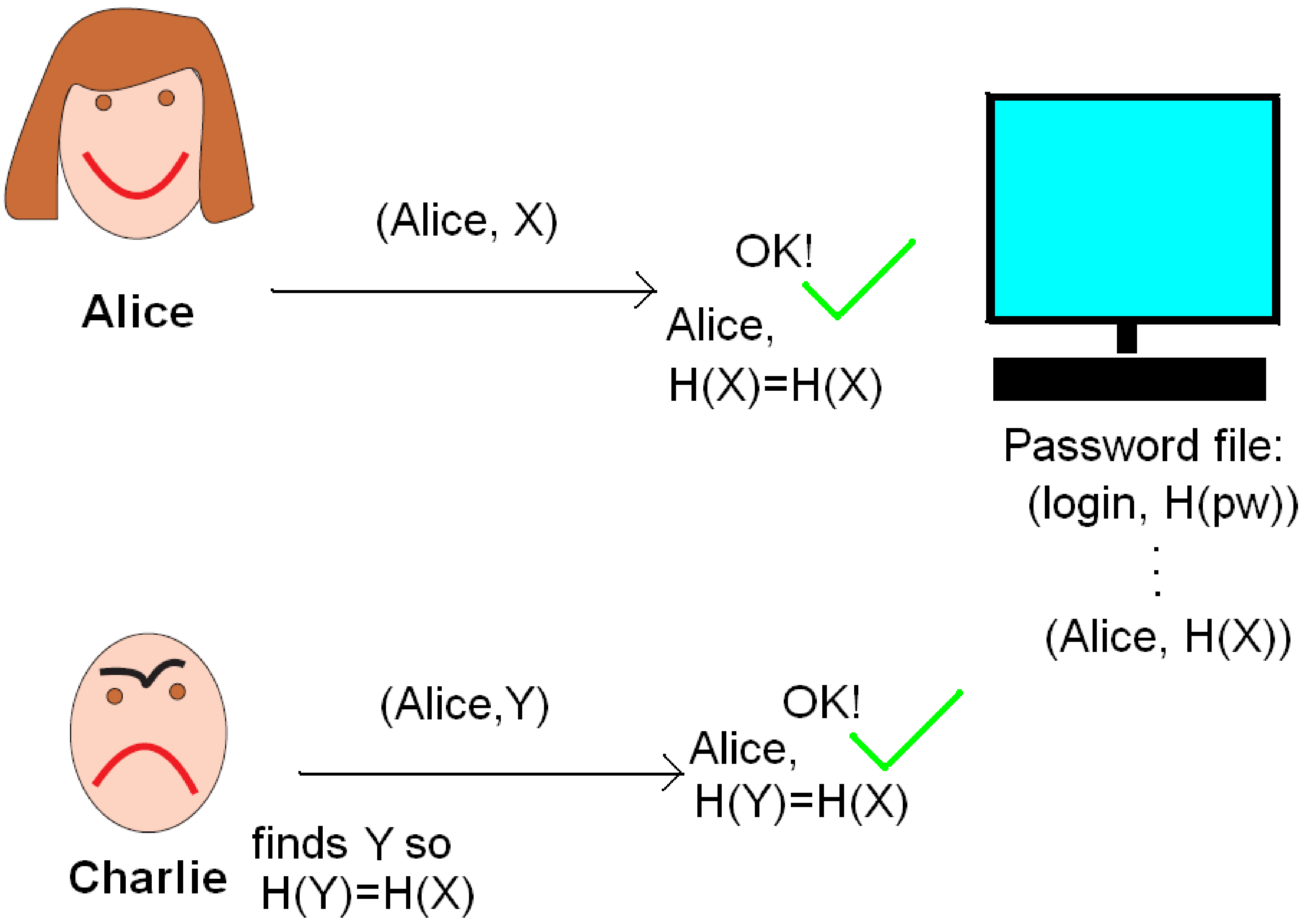
- Collision resistance
  Finding two messages $\mathcal{M}$ and $\mathcal{M}'$ so that $\mathcal{H}(\mathcal{M}) = \mathcal{H}(\mathcal{M}')$ must be "hard".
- Second preimage resistance
  Given a message $\mathcal{M}$ and $\mathcal{H}(\mathcal{M})$, finding another message $\mathcal{M}'$ so that $\mathcal{H}(\mathcal{M}) = \mathcal{H}(\mathcal{M}')$ must be "hard".
- Preimage resistance
  Given a hash $\mathcal{H}$, finding a message $\mathcal{M}$ so that $\mathcal{H}(\mathcal{M}) = \mathcal{H}$ must be "hard".

Security requirements of hash functions?

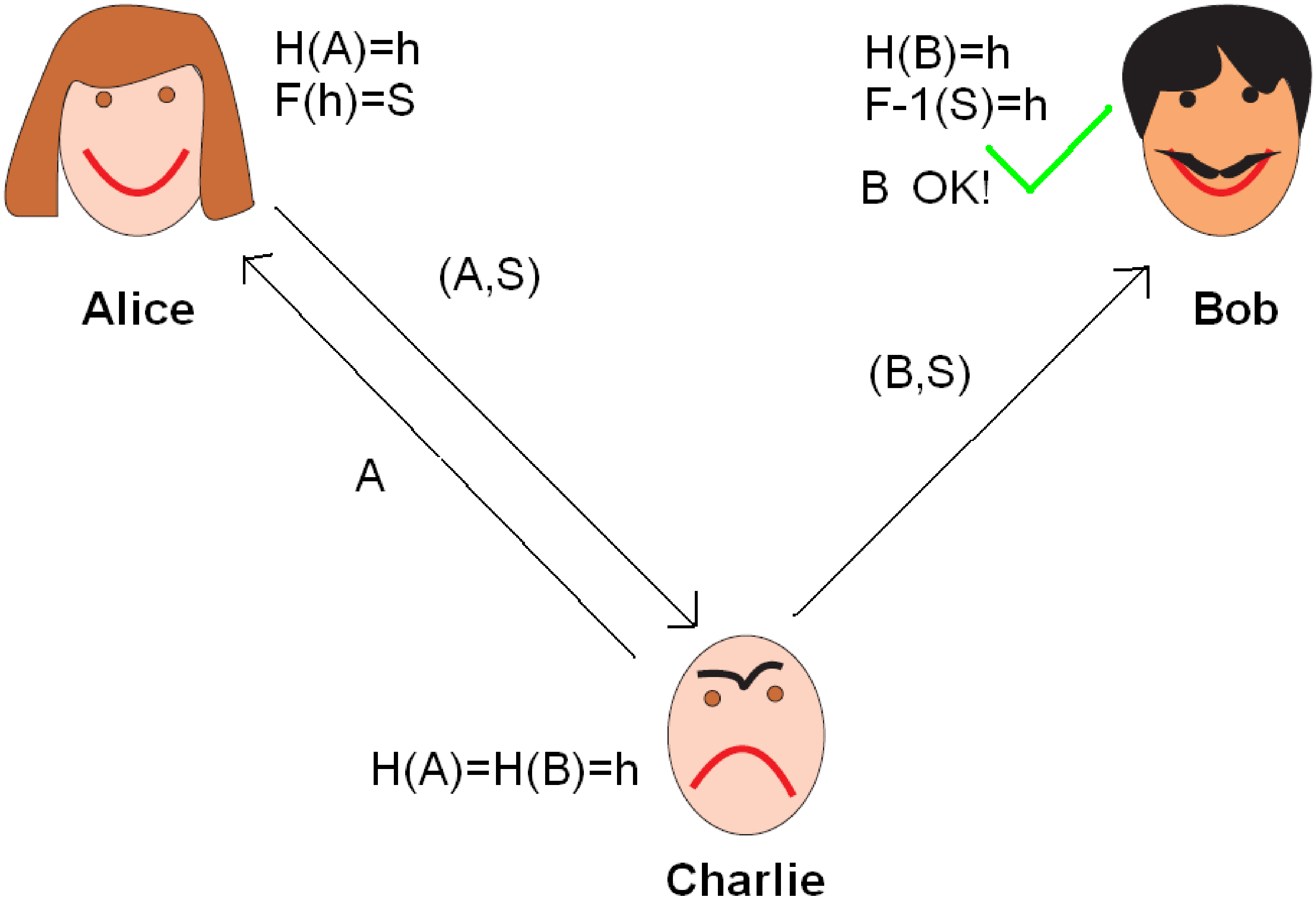
A strict definition of "hard":

- Collision resistance
  - Generic attack needs $2^{\ell_h/2}$ hash function calls $\Rightarrow$ any attack requires at least as many hash function calls as the generic attack.
- Second preimage resistance and preimage resistance
  - Generic attack needs 2^{ℓ_h} hash function calls $\Rightarrow$ any attack requires at least as many hash function calls as the generic attack.

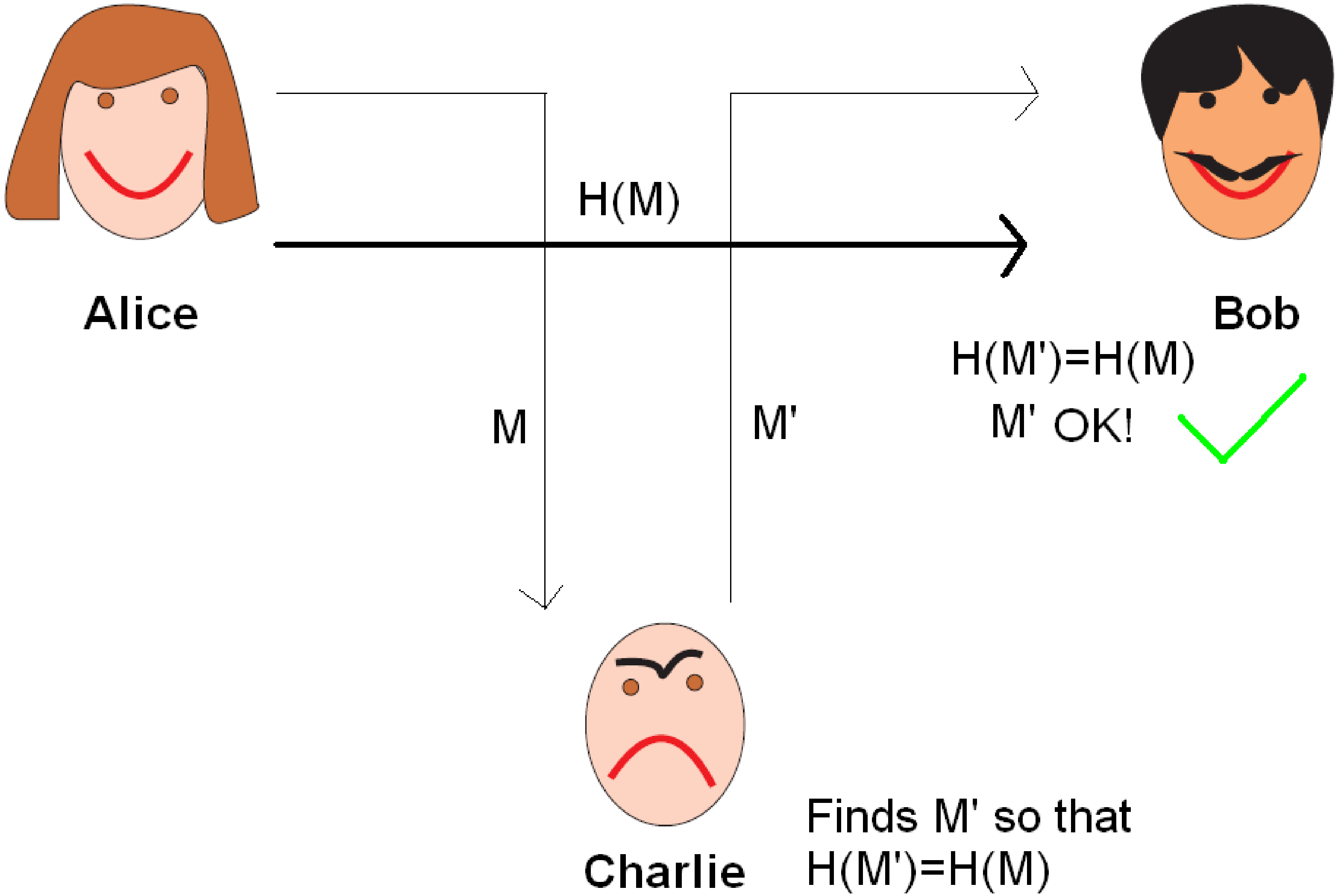
Why Preimage Resistance? Example



Why Collision Resistance? Example



Why 2nd Preimage Resistance? Example



Iterative Hashing

- Difficulty to create algorithms with an arbitrary length input: concept of iterative hashing.

- The message is split into blocks. Typically, an iterative hash function can be defined by:
  - a compression function, that takes a chaining value and a message block and generates a new chaining value.
  - an construction, that defines how to iterate the applications of the compression function.

Padding the message

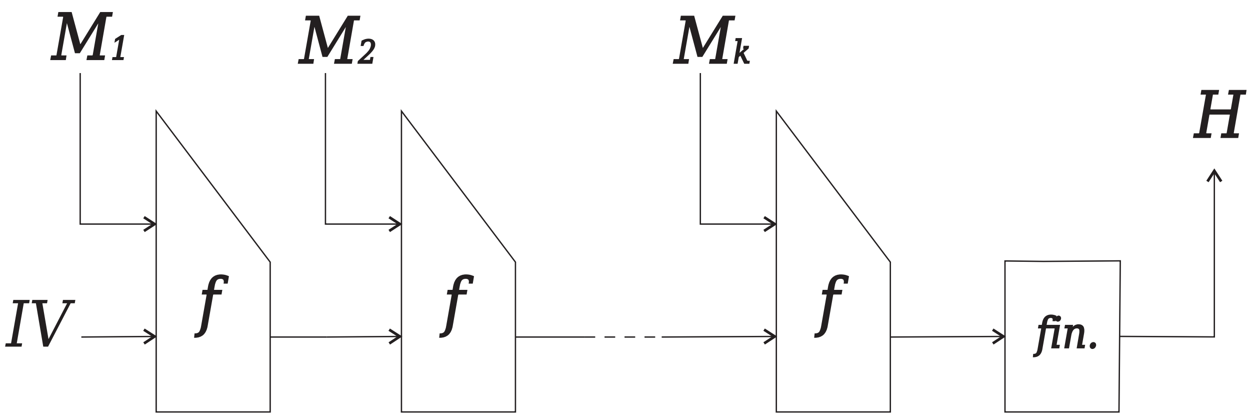
- Cut the message in blocks of fixed length.
- If the length of the message is not a multiple of the size of the block?
  - we can not just complete it with zeroes:
  - 00010 and 0001000 can produce a collision.

- Ex. of sound padding: Add '1' in the end, next add '0's until completing the block.

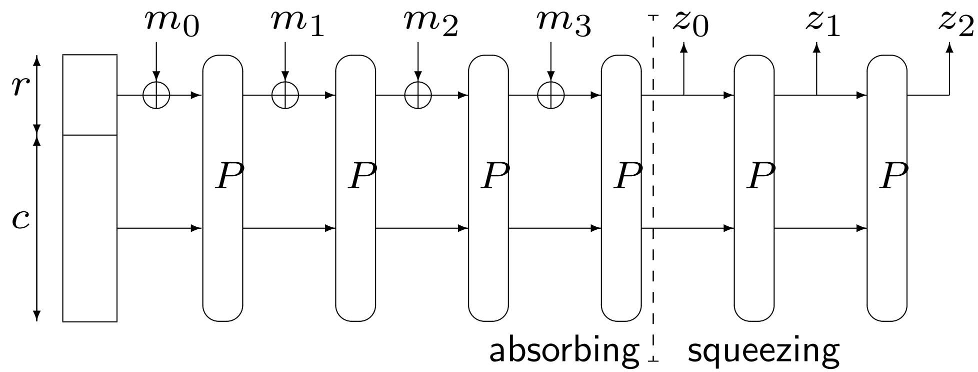
- Strengthened padding: includes the message length.

Construction: Merkle-Damgård [MD'79]

- Apply iteratively a compression function f
- Collision-resistance proof: if f is collision resistant, then the hash function is collision resistant.



Construction: Sponge [Bertoni et al. 08]



- Based on a permutation P.
- Sponge proof of indifferentiability: if P is a random permutation, then the hash function is indifferentiable from a random oracle.

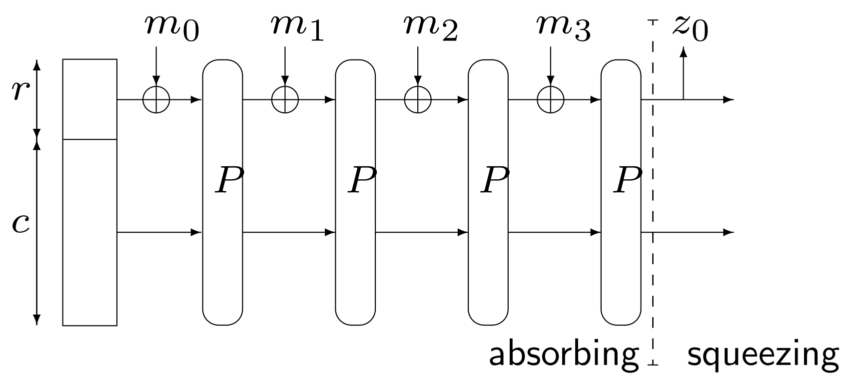
SHA-3 Competition

A NIST competition for looking for a hash standard replacement of SHA-1.

- From 2008 to 2012.
- 64 initial submissions
- 1 winner: KECCAK

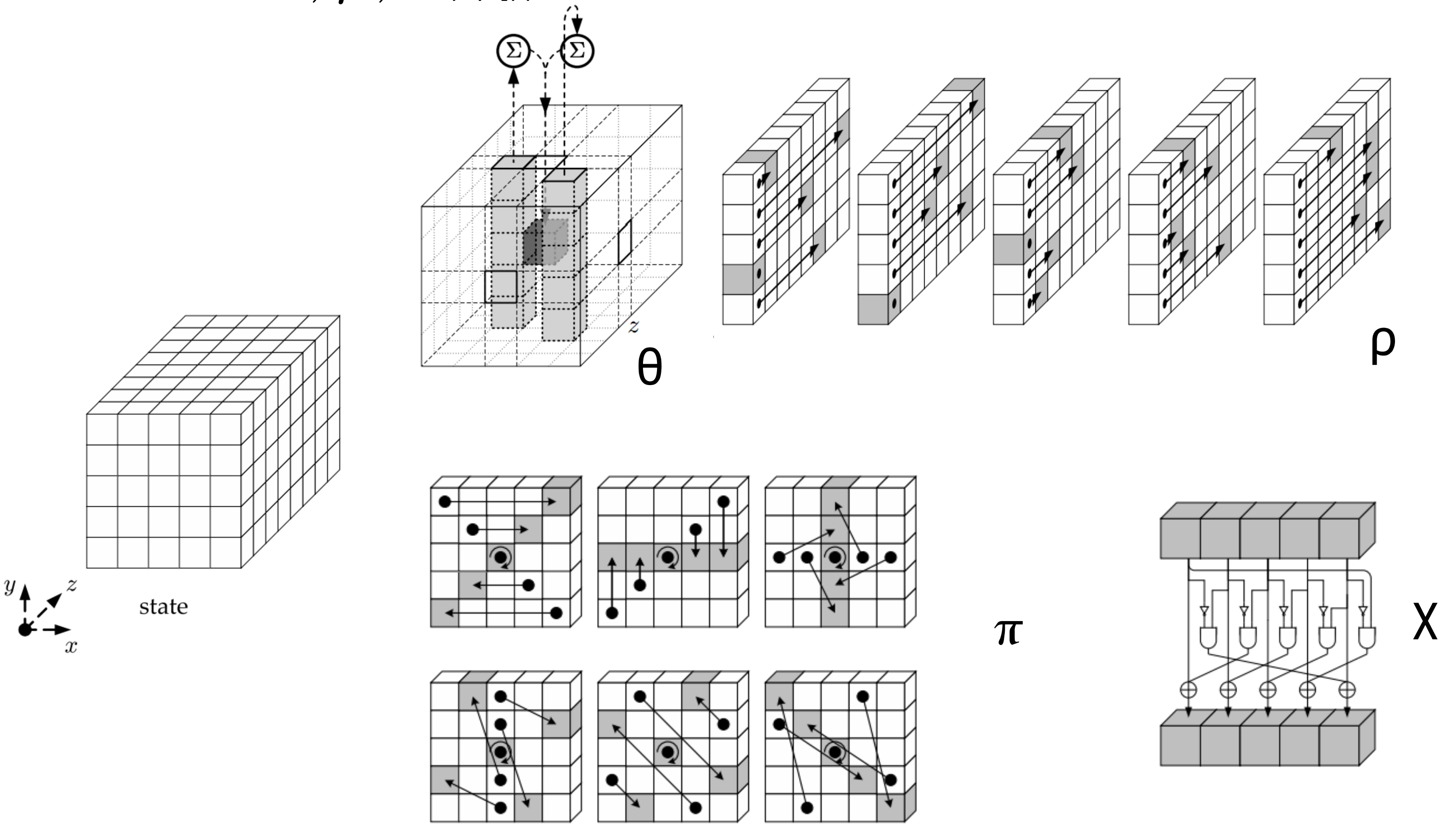
Keccak [Bertoni et al. 08]

- $|State| = 1600$ bits
- $|M| = 1024$ bits (256) or 512 bits(512).



Keccak: Internal Permutation

24 rounds of $\theta, \rho, \pi, \chi, \iota$:



Images from http://keccak.noekeon.org/Keccak-reference-3.0.pdf

Cryptanalysis

Cryptanalysis: Foundation of Confidence

> Any attack better than the generic one
> is considered a "break".

- Proofs on symmetric primitives need to make unrealistic assumptions.
- We are often left with an empirical measure of the security: cryptanalysis.

Cryptanalysis

Studies the security of cryptographic primitives.

AKA: Trying to break the primitives, to find attacks:

Empirical measure of security.

Cryptanalysis and Confidence

Security by knowledge and not by obscurity $\rightarrow$ only good way to go.

- Primitives are known to the general public $\Rightarrow$ their best existing cryptanalysis should also be known,
- implying a great need for public cryptanalysis (the nice guys).

Current scenario

- Competitions (AES, SHA-3, eSTREAM, CAESAR).
- New needs: lightweight, FHE-friendly, easy-masking.
  $\Rightarrow$ Many good proposals/candidates.

- How to choose?

- How to be ahead of possible weaknesses?

- How to keep on trusting the chosen ones?

Cryptanalysis: Foundation of Confidence

When can we consider a primitive as secure?

- A primitive is secure as far as no attack on it is known.
- The more we analyze a primitive without finding any weaknesses, the more reliable it is.

Design new attacks + improvement of existing ones:

- essential to keep on trusting the primitives,
- or to stop using the insecure ones!

What can an attacker do?

We can consider the attacker to have access to:

- Known Ciphertexts (KPA)
- Known Plaintexts (KCA)
- Chosen Plaintexts (CPA)
- Chosen Ciphertexts (CCA)
- Adaptative-Chosen Plaintexts...(ACPA)

In general: we expect the primitives to resist attacks in the strongest possible non trivial setting.

On weakened versions

If no attack is found on a given cipher, what can we say about its robustness, security margin?

The security of a cipher is not a 1-bit information:

- Round-reduced attacks.
- Analysis of components.

$\Rightarrow$ determine and adapt the security margin.

Ex.: Advanced Encryption Standart

Winner: AES-128, 10 rounds.

- 1998: best internal attack: 6 rounds.
- 2001: new attack on 7 rounds.
- 2001 to 2018: more than 30 new attacks, improving complexity.
- 2018: best known attack is still on 7 rounds. Best complexity: 2^{97} data, 2^{99} time and 2^{98} memory [DFJ12].

"The hard problem here is to break AES" (*Anne Canteaut*)

On high complexities

When considering large keys, sometimes attacks breaking the ciphers might have a very high complexity far from practical *e.g.*. 2^{120} for a key of 128 bits.

Still dangerous because:

- Weak properties not expected by the designers.
- Experience shows us that attacks only get better.
- Other existing ciphers without the "ugly" properties.

On very high complexities

Attack complexity reduced by one or two bits regarding generic attack:

- When determining the security margin: find the highest number of rounds reached.
- Security redefinition when a new generic attack is found (e.g. accelerated key search with bicliques [BKR 12]).

On weaker scenarios

Key recovery, state recovery, plaintext recovery vs ...

Distinguishers are dangerous: *e.g.* to decide between only two possible plaintexts.

Related-keys might be dangerous, depending on the use of the cipher (if used in hash functions, these properties should be known).

On weaker scenarios

Collision, preimage, second-preimage vs ...

Distinguishers, compression function collisions, semi-free start collisions... (might invalidate proof assumptions).

In general, most of the cases might be seen as non-expected "ugly" properties. Better to consider other existing ciphers without the "ugly" properties.

Cryptanalysis Warnings

Recommendations should be respected. For example:

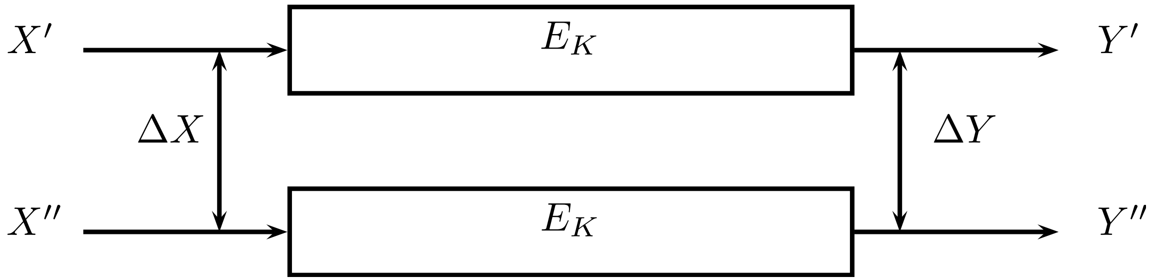
- Flame [2012]: collisions on MD5[WFL2004].
- Attaque sur TLS[ABP..13]: Bias of RC4[FMS01].
- Sloth[BL16]: collisions on MD5[WLF2004].

Problems that were predicted !!

Differential Cryptanalysis

Differential Cryptanalysis [BS'90]

Given an input difference between two plaintexts, some output differences occur more often than others.

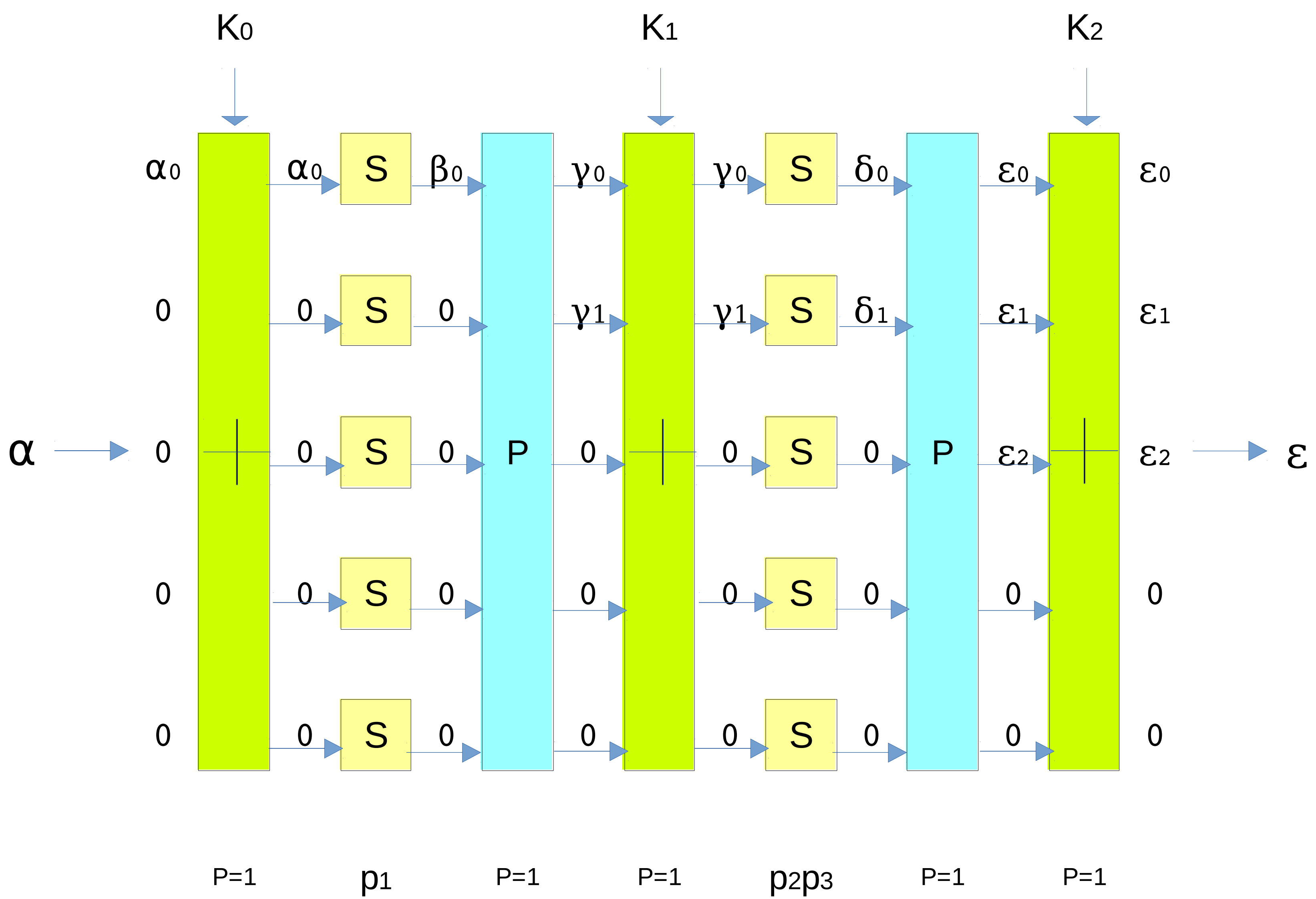


Differential: input and output difference $(\Delta X, \Delta Y)$.
Differential probability:
$P_{X,K}[E_K(X) \oplus E_K(X + \Delta X) = \Delta Y]$ (vs 2^{-n}).
Chosen Plaintext Attacks. Provides a distinguisher.

Differential paths

- Differential path $=$ configuration of differences in the internal state through rounds.
- Each differential path has a probability of being verified.
- Easier to compute a priori: hypothesis of stochastic equivalence: consider the rounds independent: compute the differential probability of a path by multiplying the probability of each round.
- The S-box DDT provides, for all α and β:
  $DDT[\alpha, \beta] = \#\{x | S(x + \alpha) + S(x) = \beta$
- DP of linear layer is 1.

Differential path: example



Differential Cryptanalysis [BS'90]

Probability of differential: sum of all the differential paths. Hard to determine. Try to approximate by the highest probability ones...

Many hypothesis: actually, rounds are not independent, for some keys it (not always) behaves like a random key...

$\Rightarrow$ Importance of implementing attacks (or reduced-round attacks) in order to verify theoretical assumptions.

Last round attacks: key recovery

R-round differential$(\Delta X, \Delta Y)$ of high probability

$$\Downarrow$$

attack $R + n$ rounds of the cipher.

1. Find many pairs with input difference ΔX.

2. Encrypt each of them for $R + n$ rounds of the cipher.

If the **partial decryption** of the last n rounds leads to a difference ΔY frequently enough, then the key bits involved are the correct ones with **high probability**.

Differential Cryptanalysis

Many improvements, related techniques:

- Truncated differentials
- Neutral bits
- Conditional differentials
- Impossible differentials
- Rebound attacks...

Linear Cryptanalysis

Linear cryptanalysis [MY'92]

- The dual of differential cryptanalysis:
- Exploit the existence of (highly) biased affine relations between some plaintext and ciphertext bits.
- This bias can be used to mount a distinguisher or even to recover some keybits.

Linear cryptanalysis [MY'92]

This expression

$$\bigoplus_{i \in S_p} P_i \;\oplus\; \bigoplus_{j \in S_K} K_j \;=\; \bigoplus_{k \in S_C} C_k$$

is verified with high bias $2^{-\varepsilon}$:

$$Pb = \frac{1}{2}(1 \pm 2^{-\varepsilon}),$$

with about $2^{2\varepsilon}$ data we can detect the bias. Known plaintext attacks.

Improvements Linear cryptanalysis

- Big number of (very) technical improvements.

- Many variants: last-round, multiple, multidimensional, zero correlation,...

We are always looking at how to improve the complexities, how to reach more rounds...

Important/Future Directions

Important/Future Directions

- Permutaton-based primitives (sponge family)
- Lightweight primitives $\Rightarrow$ new NIST competition
- New needs: FHE, masking..
- Post-quantum security?

Conclusion

Conclusion

- Many new needs/ scenarios
- Cyptanalysis: new techniques, improvements, families. A never ending task.
- Better safe than sorry!
- To be continued on Friday with Lightweight Primitives and Cryptanalysis.