Chapitre 2
Cryptography Basics

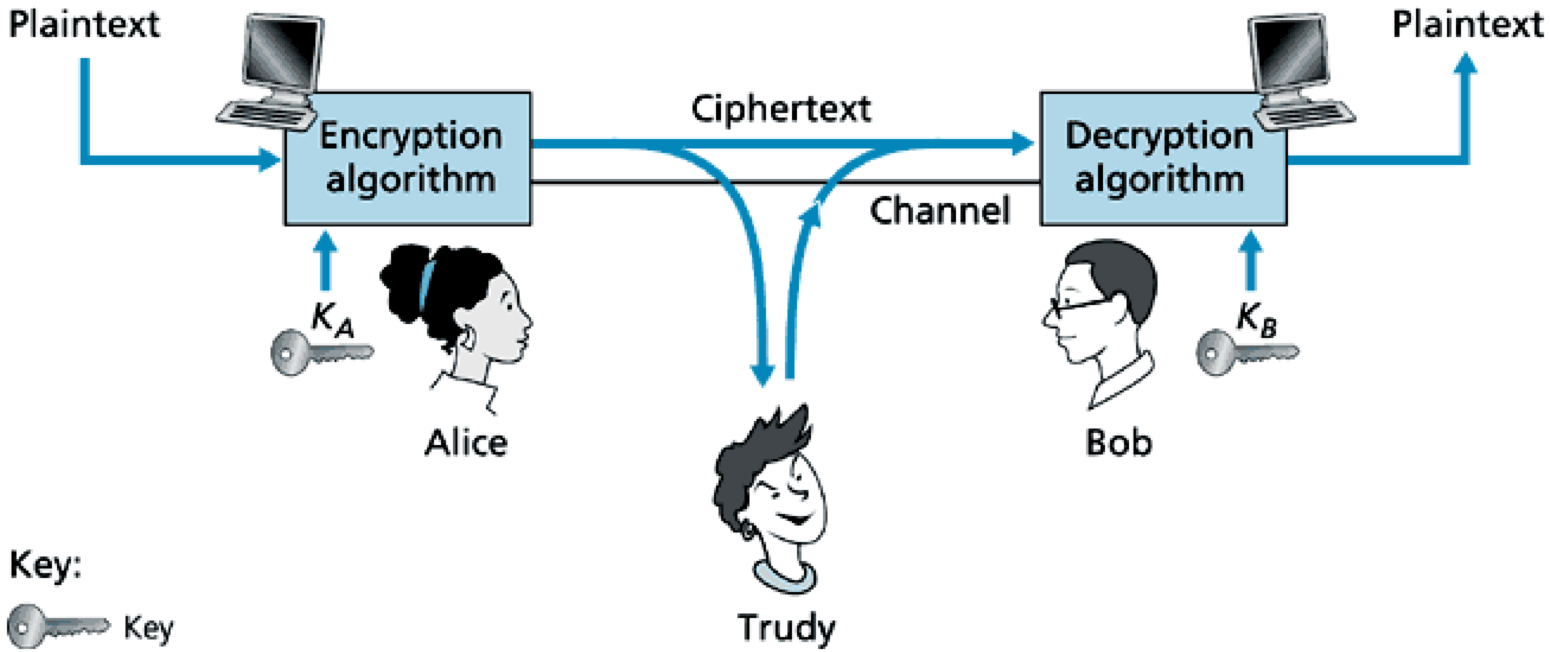
Laurent Schumacher (UNamur)
Dernière mise-à-jour : 05 janvier 2019

UNIVERSITÉ DE NAMUR

Outline

- Introduction
- Cryptography
  - Symmetric key encryption: DES, AES
  - Public key encryption: RSA, ECC
  - Hash functions
- Key distribution and certification

Introduction
Cryptography – From plaintext to cipher, and back



Introduction
Computational security

- Encryption scheme is said “*computationally secure*” if
  - The cost of breaking the cipher exceeds the value of the encrypted information,
  - The time required to break the cipher exceeds the useful lifetime of the information.
- One approach to crack an encryption is a brute-force approach (*exhaustive search*)
- On average, half of all possible keys have to be tried to find the needed key.
- No *security by obfuscation*
  - Use algorithms allowing public scrutiny
  - Don't design your own algorithm

Introduction
Where is the secret?

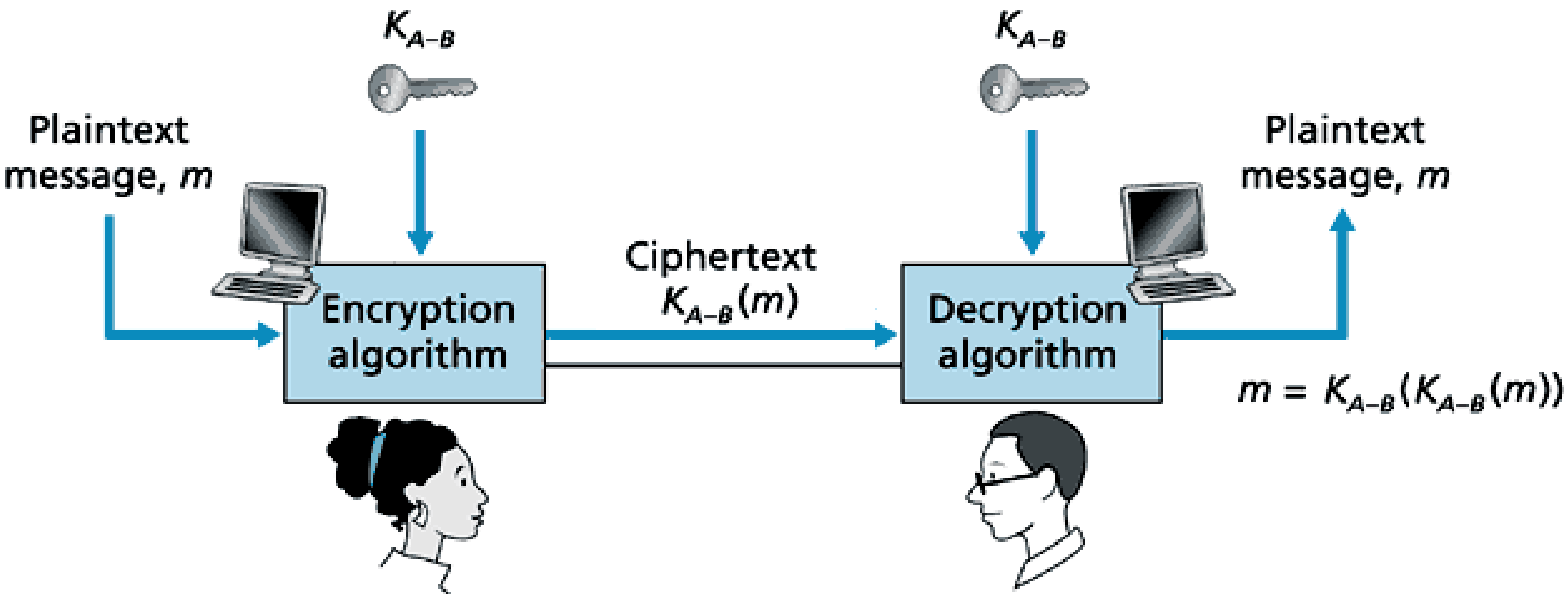
- Symmetric key cryptography
  - Sender and receiver’s keys are identical
  - They share a secret
- Public key cryptography
  - Encryption key public, decryption key secret (private)
  - Secret on single side

Outline

- Introduction
- Cryptography
  - Symmetric key encryption: DES, AES
  - Public key encryption: RSA, ECC
  - Hash functions
- Key distribution and certification

Symmetric key cryptography

- Alice and Bob share know same (symmetric) key
- Key $K_{A\text{-}B}$ is (seed for) substitution pattern
- How do Bob and Alice agree on key value?



Symmetric key cryptography
Basic examples (1/3)

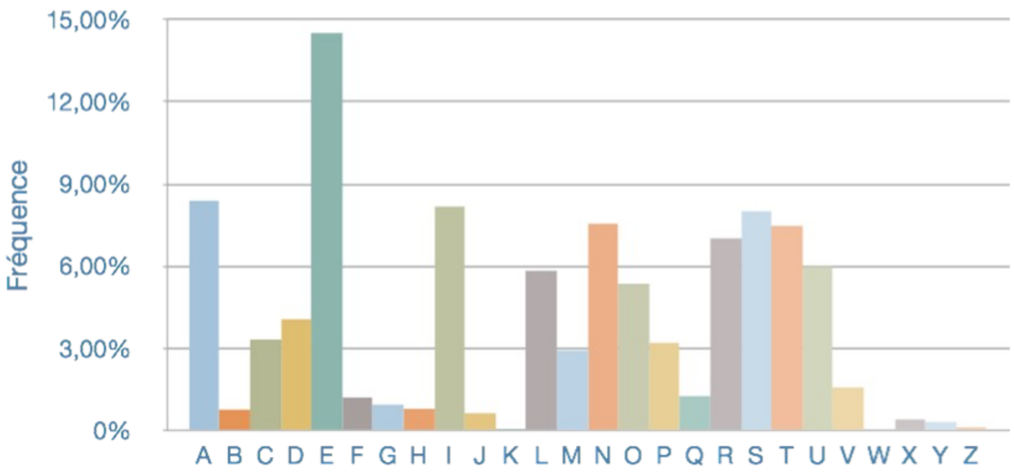
- Principle: substitute plaintext with ciphertext
- Caesar cipher *k*
  - Substitute a letter by another one, *k* positions later
  - bob, i love you. alice → ere, l oryh brx. dolfh
  - 25 key values
- Monoalphabetic cipher
  - No regular pattern

Plaintext letter:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext letter:	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

  - bob, i love you. alice → nkn, s gktc wky. mgsbc
  - 26! = 403 10^{24} key values

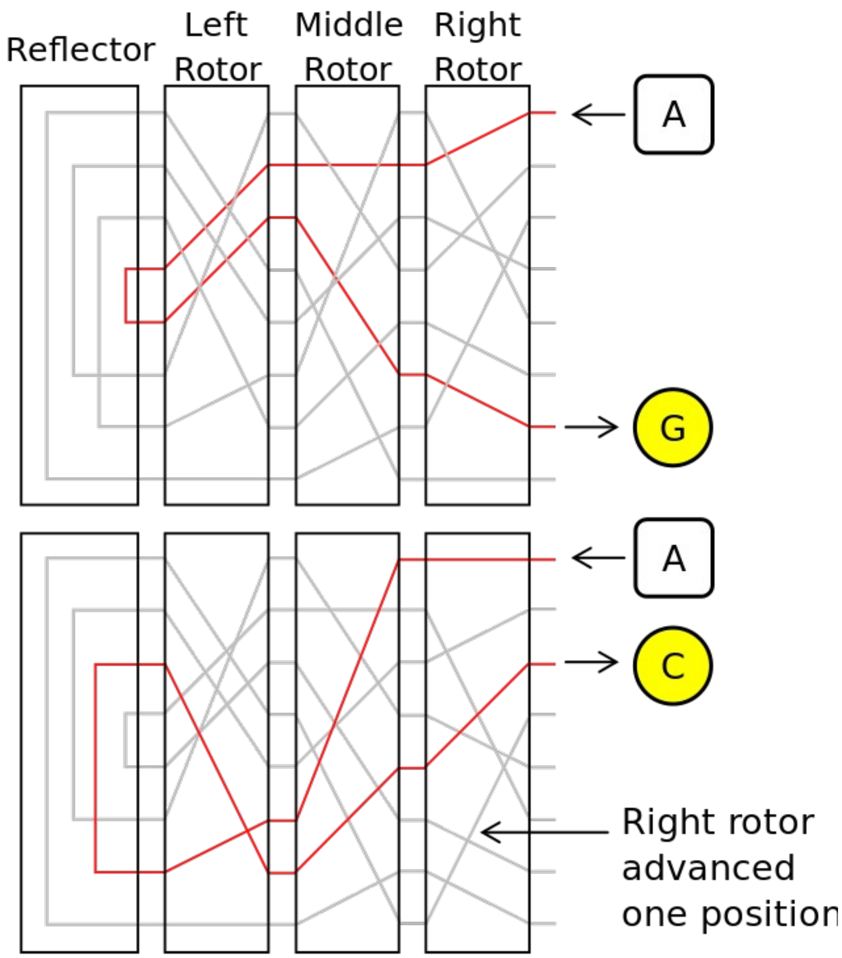
Symmetric key cryptography
Basic examples (2/3)

- Monoalphabetic cipher (con't)
  - Attack eased by statistical analysis of content



- Polyalphabetic encryption
  - Multiple, successive encodings
  - Example: Caesar ciphers k = {5, 19} + $C_1C_2C_2C_1C_2$
  - bob, i love you. alice → wvi, d sjcl tvp. hsdjz

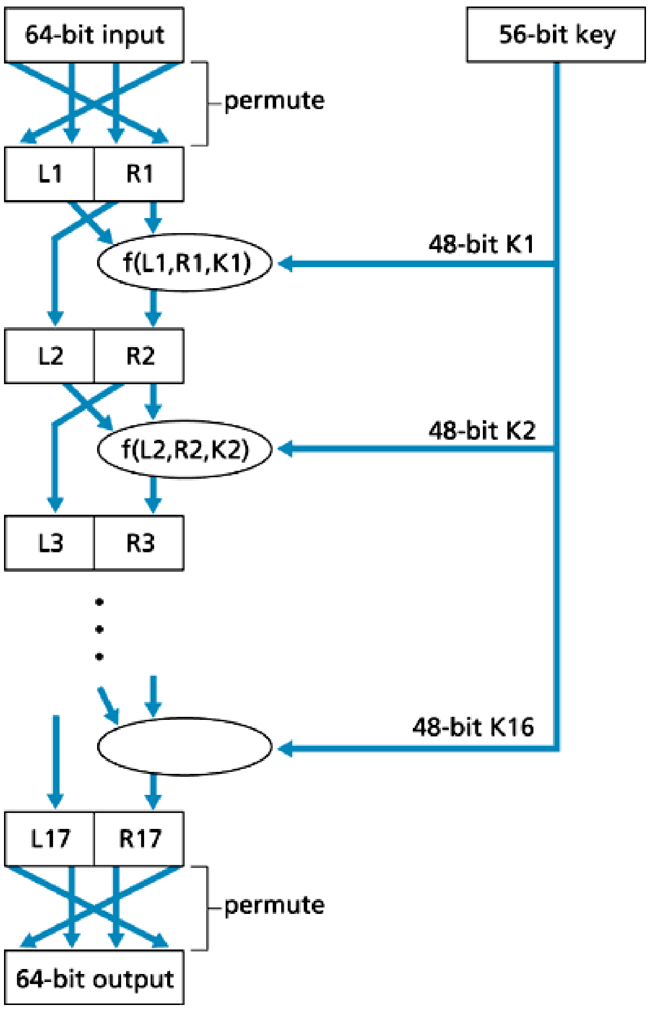
Symmetric key cryptography
Basic examples (3/3)



Source : Wikipedia

- Polyalphabetic encryption (con't)
  - Enigma (World War II)
  - Substitution scheme changed for each character
  - Secret: rotors + initial position of rotors and cables
  - Brute-force: 158,962,555,217,826, 360,000 settings

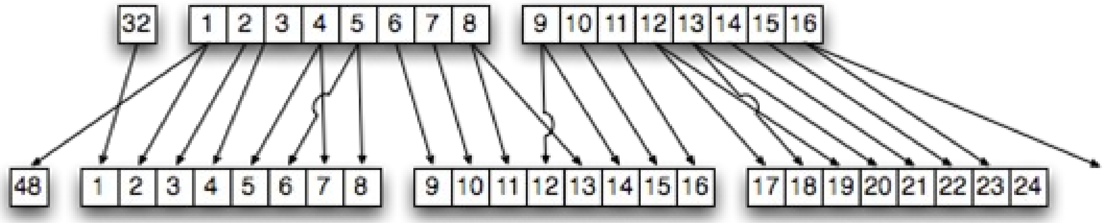
Symmetric key cryptography
Data Encryption Standard (DES, 1/3)



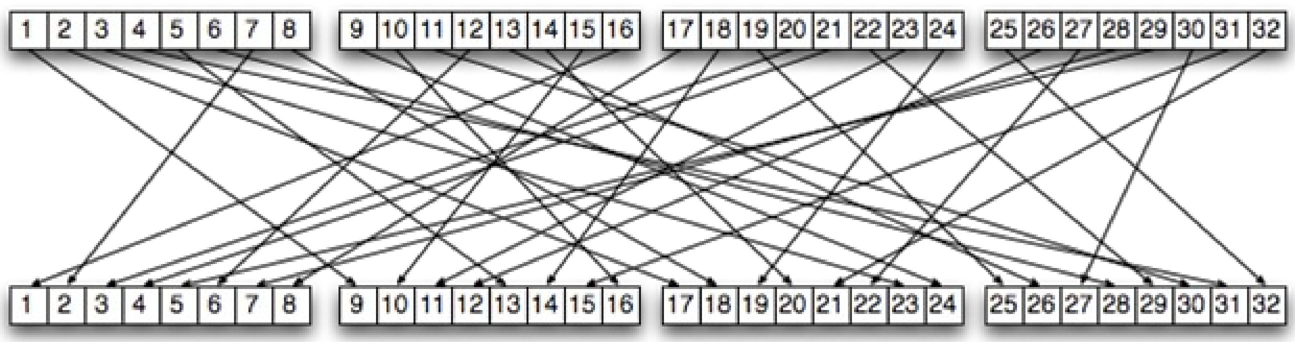
- Chunks of 64-bit plaintext input encrypted with 56-bit symmetric key
- Encryption
  - Initial permutation
  - 16 identical “rounds” of
    - Swap left-right
    - Function application, each using a different 48-bit key derived from the 56-bit key
    - Function combines expansion, substitution and XOR
  - Final permutation
- Decryption: reverse operations

Symmetric key cryptography
Data Encryption Standard (DES, 2/3)

- Expansion (E-Box)



- Permutation (P-Box)



- Actual connections depend on the secret

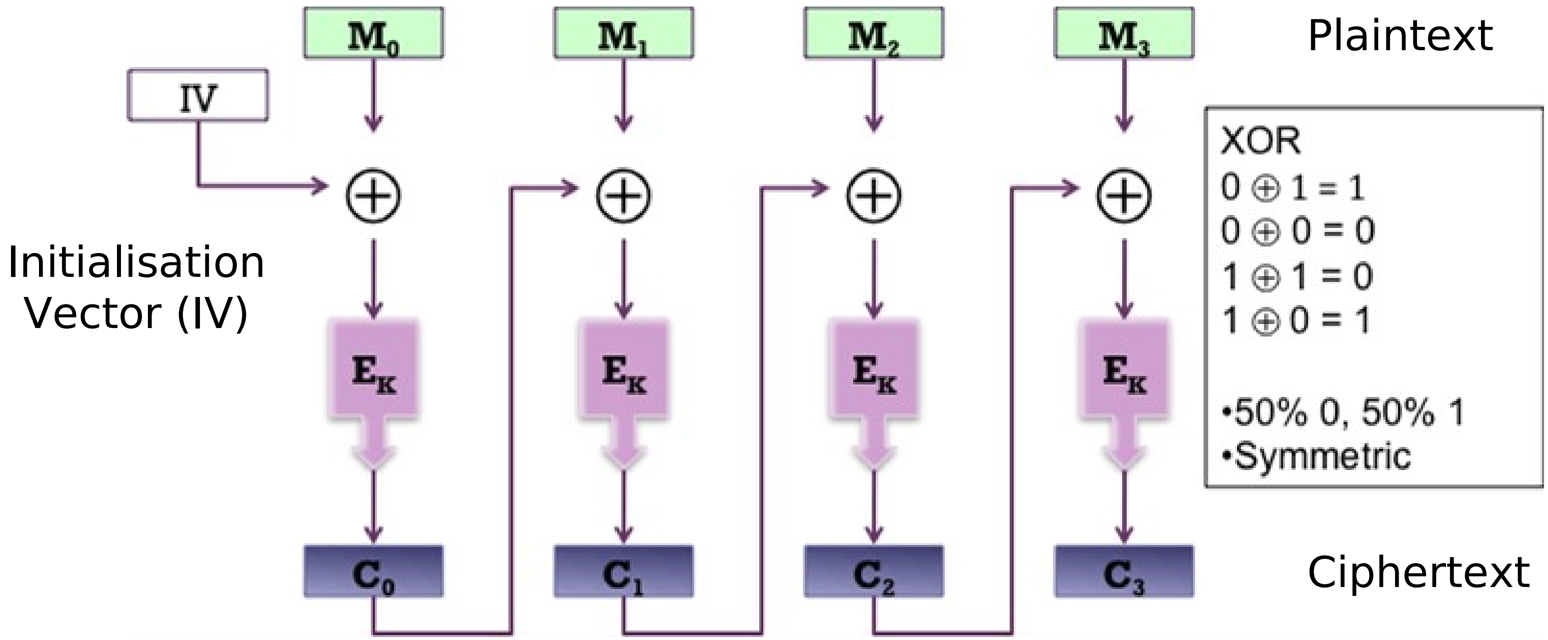
Symmetric key cryptography
Data Encryption Standard (DES, 3/3)

- Goal: no correlation between ciphertext and either original data or encryption key
- No known “backdoor” decryption approach, only exhaustive search
- How secure is DES? DES Challenge

Challenge I	January 1997	96 days
Challenge II-1	February 1998	41 days
Challenge II-2	July 1998	56 hours
Challenge III	January 1999	22,25 hours

Symmetric key cryptography
3-DES – CBC

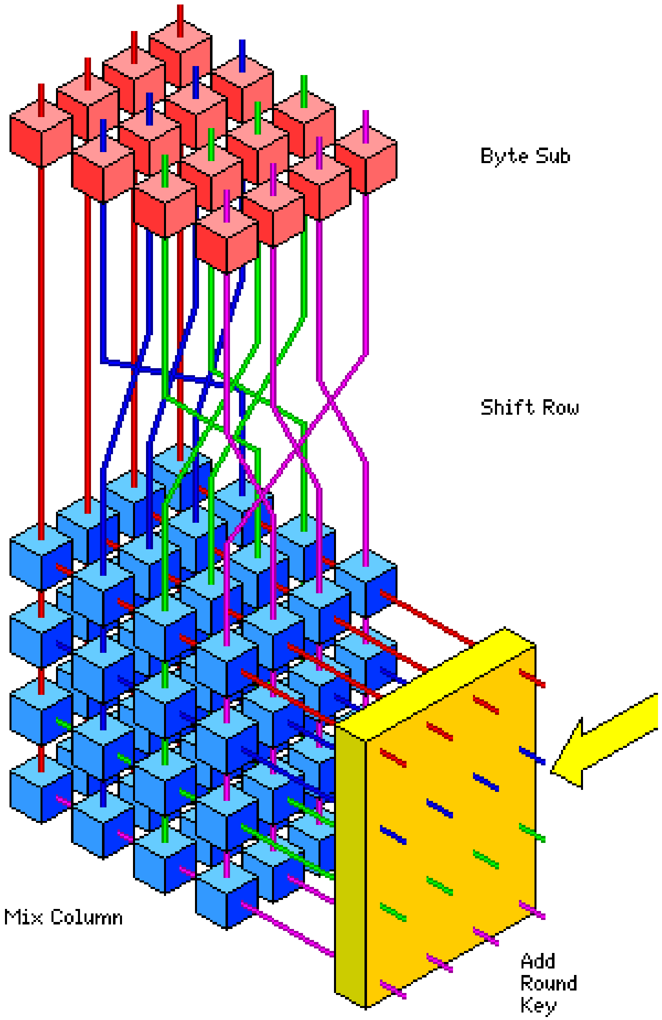
- 3-DES: sequence of three DES encodings
- Cipher-Block Chaining (CBC)
  - XOR($K(m^j)$, $m^{(j+1)}$) before cyphering $m^{(j+1)}$
  - IEEE 802.11i



Symmetric key cryptography
Advanced Encryption Standard (AES, 1/2)

- Originally proposed in 2000 as Rijndael by Vincent Rijmen (KULeuven) and Joan Daemen (Proton World International)
- Adopted by NIST as DES replacement
- A machine able to crack 56-bit DES in one second would need 149 10^{12} years to crack AES
- No academic attack based on cryptanalysis found yet. Repeated attempts to shorten sequence of operations.

Symmetric key cryptography
Advanced Encryption Standard (AES, 2/2)



- Chunks of 128 bits using 128-, 192- and 256-bit keys
- 10/12/14 identical “rounds” of
  - Byte substitution
  - Row shifting
  - Column mixing
  - Addition with key + rounding
- Illustration

Source: http://home.ecn.ab.ca/~jsavard/crypto/images/rijnab.gif

Symmetric key cryptography
AES cryptanalysis

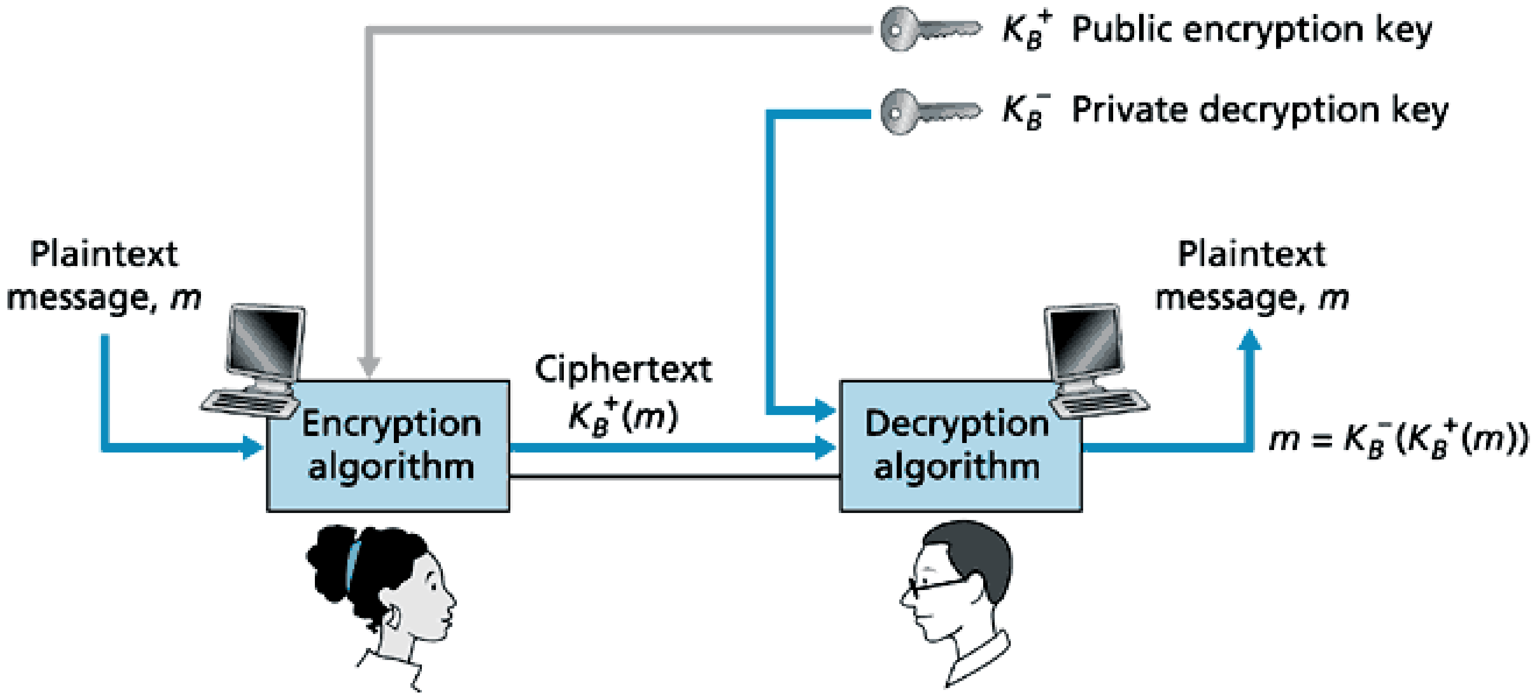
- Repeated attempts to shorten the sequence
- First meaningful succes in august 2011
- Equivalent to a 2-bit reduction of the key length (gain = 4)
- Number of steps for brute-force attacks remains $8*10^{37}$
- With 10^{12} devices testing 10^{9} keys per second, still $2*10^{9}$ years requested to find a given key

Symmetric key cryptography
Other algorithms

- Non exhaustive list
  - Ron Rivest’s Ciphers RC2, RC4 (Prohibited, RFC 7465), RC5, RC6 (AES candidate)
  - International Data Encryption Algorithm (IDEA)
  - CAST (C Adams, S. Tavares)
- Similar method
  - Split plaintext in *N*-bit words
  - Repeat basic secret key-dependent operations
  - Substitution, permutation, XOR, etc

Public key cryptography

- Alice fetches Bob's public key K_{B^+} and encrypts message using K_{B^+} and a standardised algorithm
- Bob decrypts ciphertext using his own private key K_{B^-}



Public key cryptography
Requirements

- Pretty simple
- Two requirements
  - $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$
  - Given the public key K_{B^+}, it should be impossible to derive the private key K_{B^-}
- Drawbacks
  - Enables chosen plaintext attacks, since anyone can send a ciphered message
  - Key ownership not longer authenticates sender
- Algorithms
  - 1^{st} generation: RSA (Rivest, Shamir and Adleman)
  - 2^{nd} generation: ECC (Elliptic Curve Cryptography)

Public key cryptography
RSA – Key generation

- Using modulo-n arithmetic
  - Choose two large prime numbers p, q (e.g. 1,024 bits each)
  - Compute $n = pq$, $z = (p-1)(q-1)$
  - Choose e (with $e < n$) that has no common factors with z (e, z are "relatively prime")
  - Choose d such that $ed-1$ is exactly divisible by z (in other words: $ed \bmod z = 1$)
  - Public key is $K_B^+ = (n,e)$, private key is $K_B^- = (n,d)$
- Example
  - $p = 5$, $q = 7$
  - $n = 35$, $z = 24$, $e = 5 < 35$, $d = 29$

Public key cryptography
RSA – Key generation with OpenSSL

- Generate a pair of keys

```
openssl genrsa -out my.key -aes128 1024
```

- Check the keys (actually *p*, *q*, *n*, *e*, *d*)

```
openssl rsa -in my.key -noout -text
```

- Extract public key K^+

```
openssl rsa -in my.key -out mypub.key -pubout
```

Public key cryptography
RSA – Cyphering and decyphering

- Given (n, e) and (n, d) as computed above
  - To encrypt bit pattern m, compute $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)
  - To decrypt bit pattern m, compute $c^d \bmod n$ (i.e., remainder when c^d is divided by n)

$$
\begin{aligned}
c^d \bmod n &= \left(m^e\right)^d \bmod n \\
&\overset{\Delta}{=} m^{\left[ed \bmod (p-1)(q-1)\right]} \bmod n \\
&= m^1 \bmod n
\end{aligned}
$$

- Security relies on the absence (so far) of algorithms for quickly factoring n into p and q
- Important property: $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$

Public key cryptography
RSA – Example

Plaintext Letter	m: numeric representation	m^e	ciphertext $c = m^e \bmod n$
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Ciphertext c	c^d	$m = c^d \bmod n$	Plaintext Letter
17	481968572106750915091411825223071697	12	l
15	12783403948858939111232757568359375	15	o
22	851643319086537701956194499721106030592	22	v
10	100000000000000000000000000000	5	e

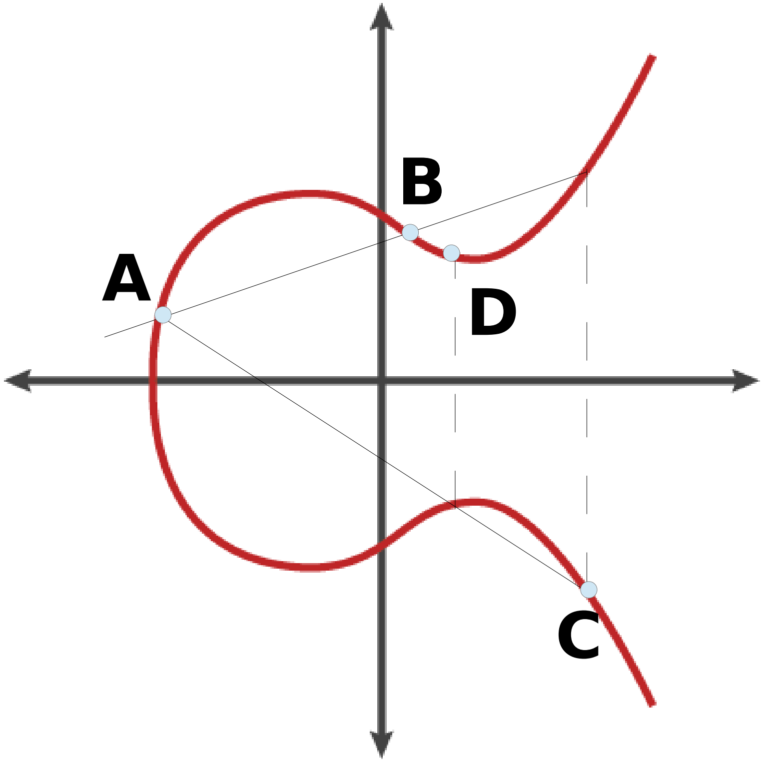
- RSA generates extremely large numbers
- Exponentiation is computationally demanding

Public key cryptography
RSA – Limits

- Moore's Law forces the increase of the size of the prime numbers
- Two issues
  - Smartphones and similar devices miss CPU and battery
  - Factorisation approximation algorithms (Quadratic Sieve, General Number Field Sieve) compute better results for large numbers
- Cryptography base on prime number factorisation is doomed
- Five years according to MIT Technology Review (august 2013)

Public key cryptography

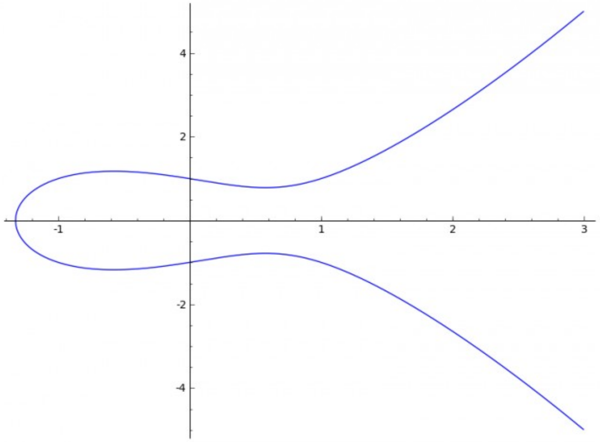
ECC – Principle (1/2)



- $y^2 = x^3 + ax + b$
- Symmetry w.r.t. x-axis
- Three section points per line
- Kind of snooker game
- Knowing the curve, point A and final point, how many runs to reach it ?

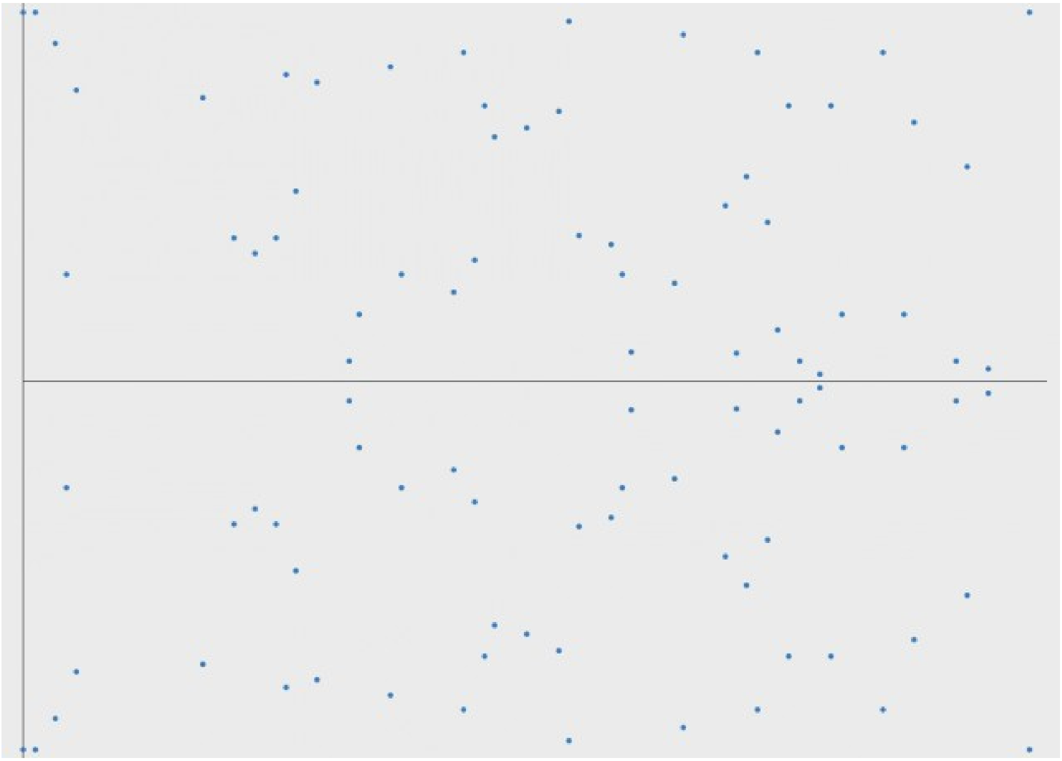
Nick Sullivan, « *A (relatively easy to understand) primer on elliptic curve cryptography* », Ars Technica, october 2013

Public key cryptography
ECC – Principle (2/2)



$$y^2 = x^3 - x + 1$$

Modulo 97
Integers
only



Public key cryptography
ECC vs. RSA

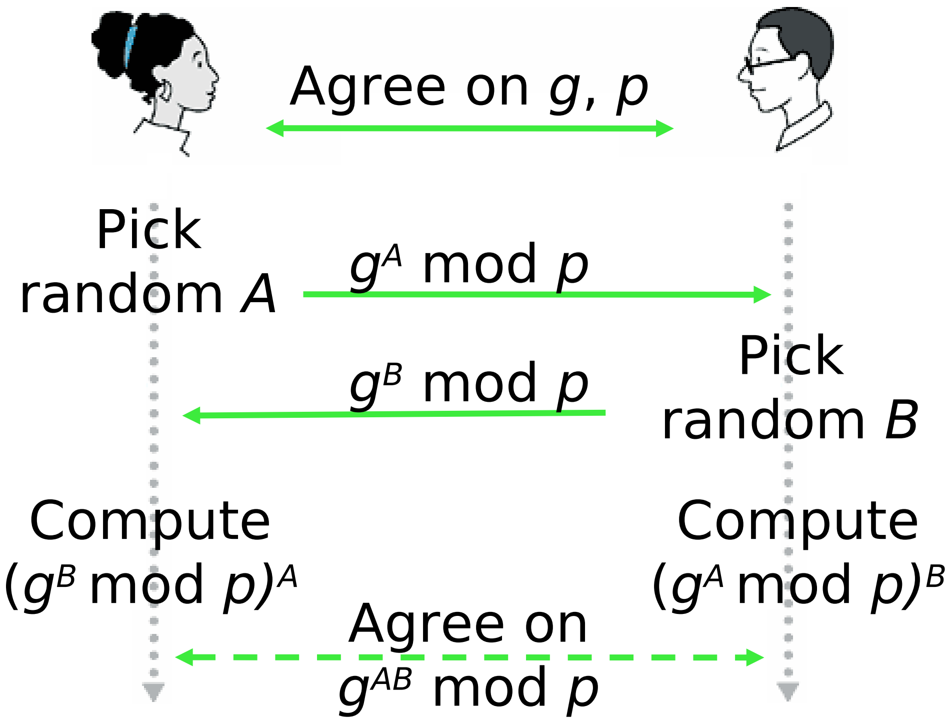
- Assuming
  - A prime number as modulo
  - An elliptic curve
  - A starting point A
- The pair of keys is
  - K^-: a number n
  - K^+: the point reached on the elliptic curve after n bounds
- Mathematical foundation : elliptic curve discrete logarithm
- Currently unsolvable but by brute force
- Same protection as RSA, but with shorter keys

Cryptography
Symmetric key vs. public key

Symmetric key	Public key
• Requires sender and receiver know shared secret key • How to agree on key in first place (particularly if never “met”)?	• Radically different approach • Sender and receiver do not share secret key • Public encryption key K^+ known to all • Private decryption key K^- known only to receiver
Based on substitutions and permutations	Based on mathematical functions

Cryptography
Session key – Diffie-Hellman or mixed scheme

- Diffie-Hellman

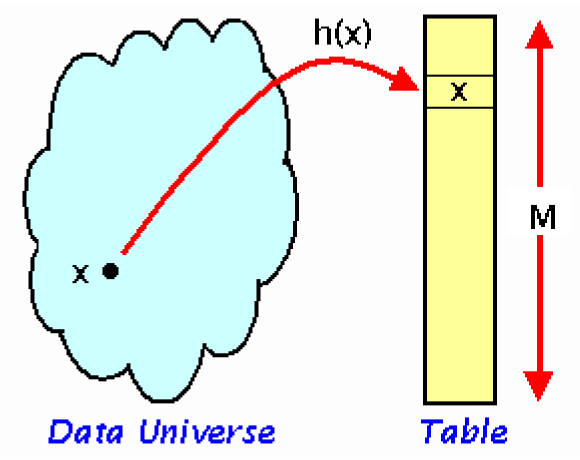


Given g, p and $g^X \bmod p$, guess X is tricky

- Mixed scheme
  - Alice generates an AES key K_S
  - Alice encrypts it using Bob's public key K_B^+
  - Bob receives the RSA-encrypted key and decrypts it with its private key K_B^- to obtain the AES session key K_S

Cryptography
Hash functions (1/2)

- Many-to-1 mapping
- Produces fixed-size outcome
- Examples: checksum, CRC



Message				ASCII Representation				
I	O	U	1	49	4F	55	31	
0	0	.	9	30	30	2E	39	
9	B	O	B	39	42	4F	42	
				B2	C1	D2	AC	Checksum

Message				ASCII Representation				
I	O	U	9	49	4F	55	39	
0	0	.	1	30	30	2E	31	
9	B	O	B	39	42	4F	42	
				B2	C1	D2	AC	Checksum

Poor hash: easy to find two messages with same checksum

Source: http://cs.engr.uky.edu/~lewis/essays/algorithms/hashing/hashing.html

Cryptography
Hash functions (2/2)

- Property of a strong hash function: it is computationally infeasible to find two messages x and y such that $H(x) = H(y)$
- Strategies
  - Multiplicative: $H(x) = 1 + (x/k)*M$ if $0 < x < k$
  - Modular: $H(x) = 1 + x \bmod M$
  - Hybrid: $H(x) = 1 + \alpha x \bmod M$
- Rules of thumb
  - M = prime number
  - $\alpha = 0.618033$

Cryptography
Hash functions – Examples

- MD2/4/5 hash functions (RFC 1321)

```
openssl dgst -md5 plaintext.txt
```

- Secure Hash Algorithm : SHA-0/1/2/3 (a.k.a. Keccak)
- RIPEMD-128/160/256/320

Cryptography
Summary

Symmetric key	Public key	Hash functions
AES	ECC	MD2/4/5
ChaCha20	RSA	SHA-0†/1†/2/3
3-DES		RIPEMD-128/160
RC2, RC4†, RC5		Poly1305
IDEA		
CAST		

Cryptography
Post-Snowden era


ars technica UK

MAIN MENU MY STORIES: 25 FORUMS

RISK ASSESSMENT / SECURITY & HACKTIVISM

How the NSA can break trillions of encrypted Web and VPN connections

Researchers show how mass decryption is well within the NSA's £7 billion budget.

by **Dan Goodin** (US) - Oct 15, 2015 9:15pm CEST

Share Tweet 97

- No hint that mathematical one-way functions have been inverted
- But hints that implementations have been deliberately weakened

Cryptography
Summary

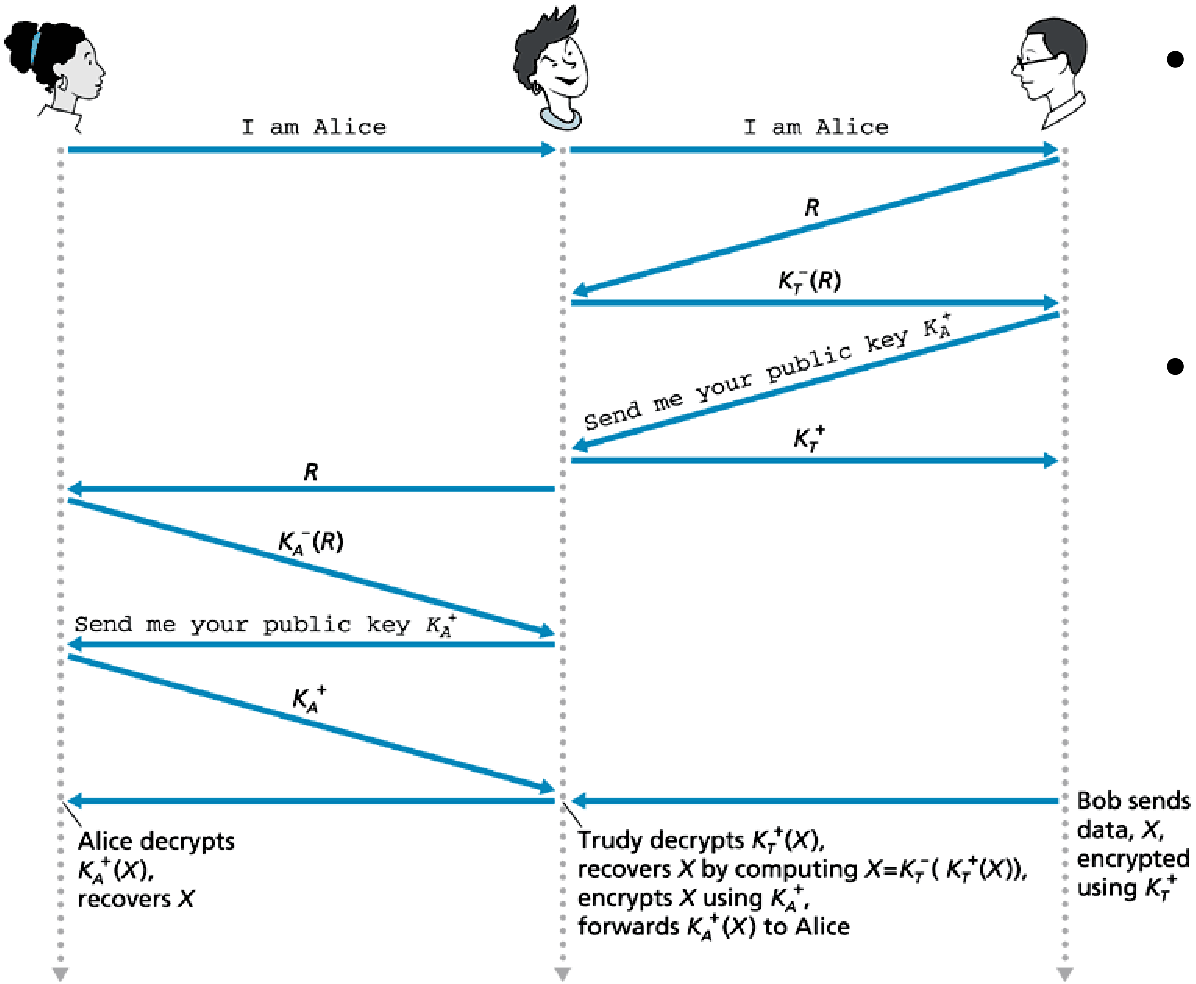
Cryptography		Until 2020	From 2020
Symmetric (AES)	Keys	100 bits	128 bits
	Words	64 bits	128 bits
Factorisation (RSA)	Modulo	2,048 bits	4,096 bits
	Exponants	$\geq 2^{16}$	$\geq 2^{16}$
ECC	Modulo	200 bits	256 bits
Hash	Digest	200 bits	256 bits

Source : Agence nationale de la sécurité des systèmes d'information, « Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques », january 2010.

Outline

- Introduction
- Cryptography
  - Symmetric key encryption: DES, AES
  - Public key encryption: RSA, ECC
  - Hash functions
- Key distribution and certification

Key distribution and certification
Person-in-the-middle attack



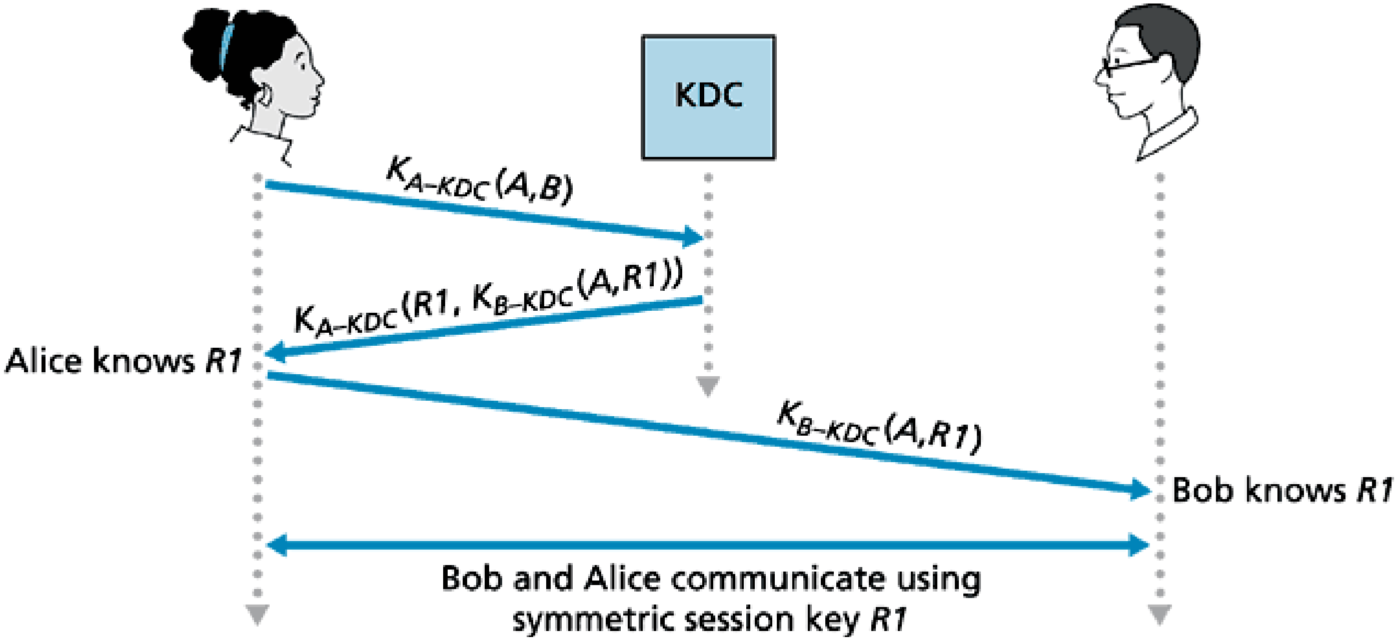
- Trudy can impersonate Alice with her own pair of keys
- Eventually, Alice and Bob will find out

Key distribution and certification
Trusted Third Party (TTP)

	Symmetric key	Public key
Issue	How to share a secret key?	How to securely collect a public key?
Solution: TTP	Key Distribution Centre (KDC)	Certification Authority (CA)

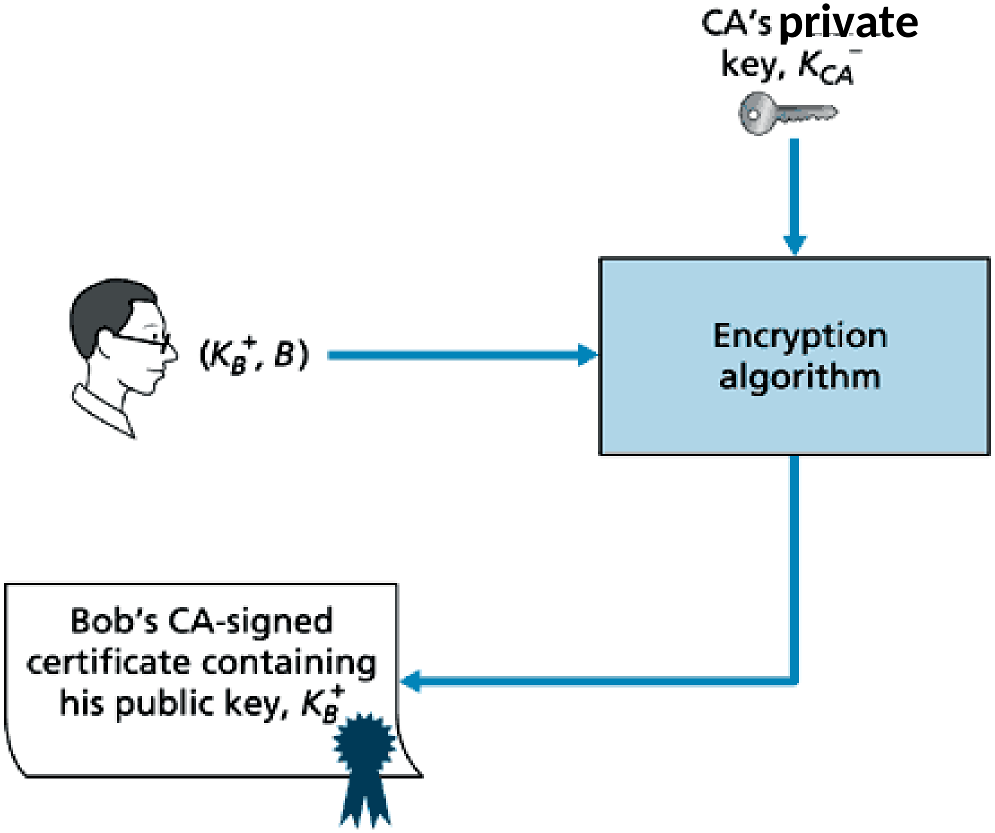
Key distribution and certification
Key Distribution Center (KDC)

- The KDC shares a secret key with each user
- A given user only knows the key it shares with the KDC
- The KDC helps its registered users to share a one-time session key (Needham-Schroeder protocol)



Key distribution and certification

Certification Authority (CA) – X.509 standard

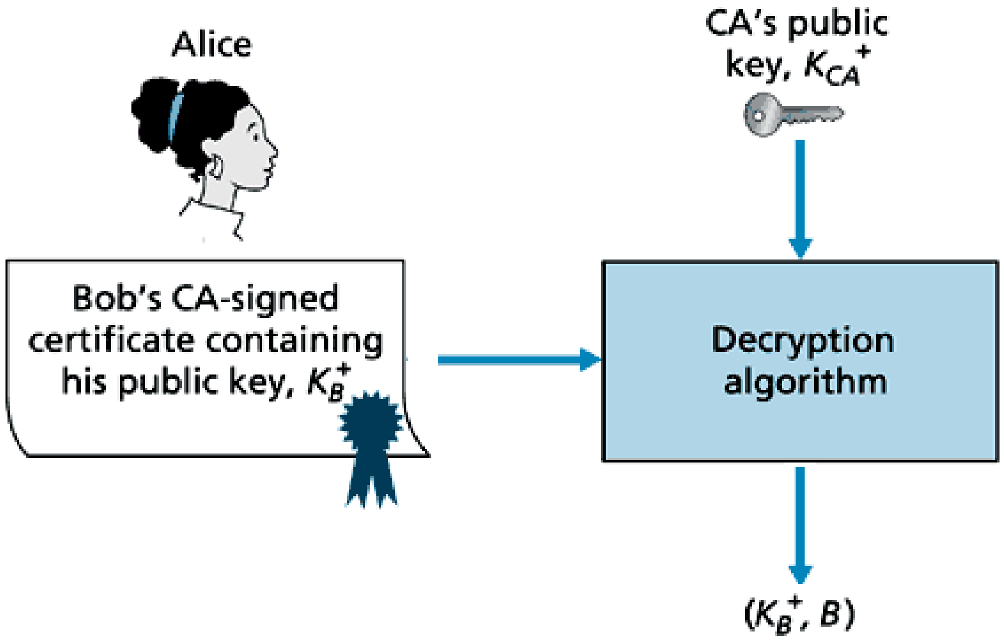


- B provides "proof of identity" to CA.
- CA creates certificate binding B to K_B^+
- Certificate containing B's public key digitally signed by CA
- CA says "this is B's public key"

Key distribution and certification

Certification Authority (CA) – X.509 standard

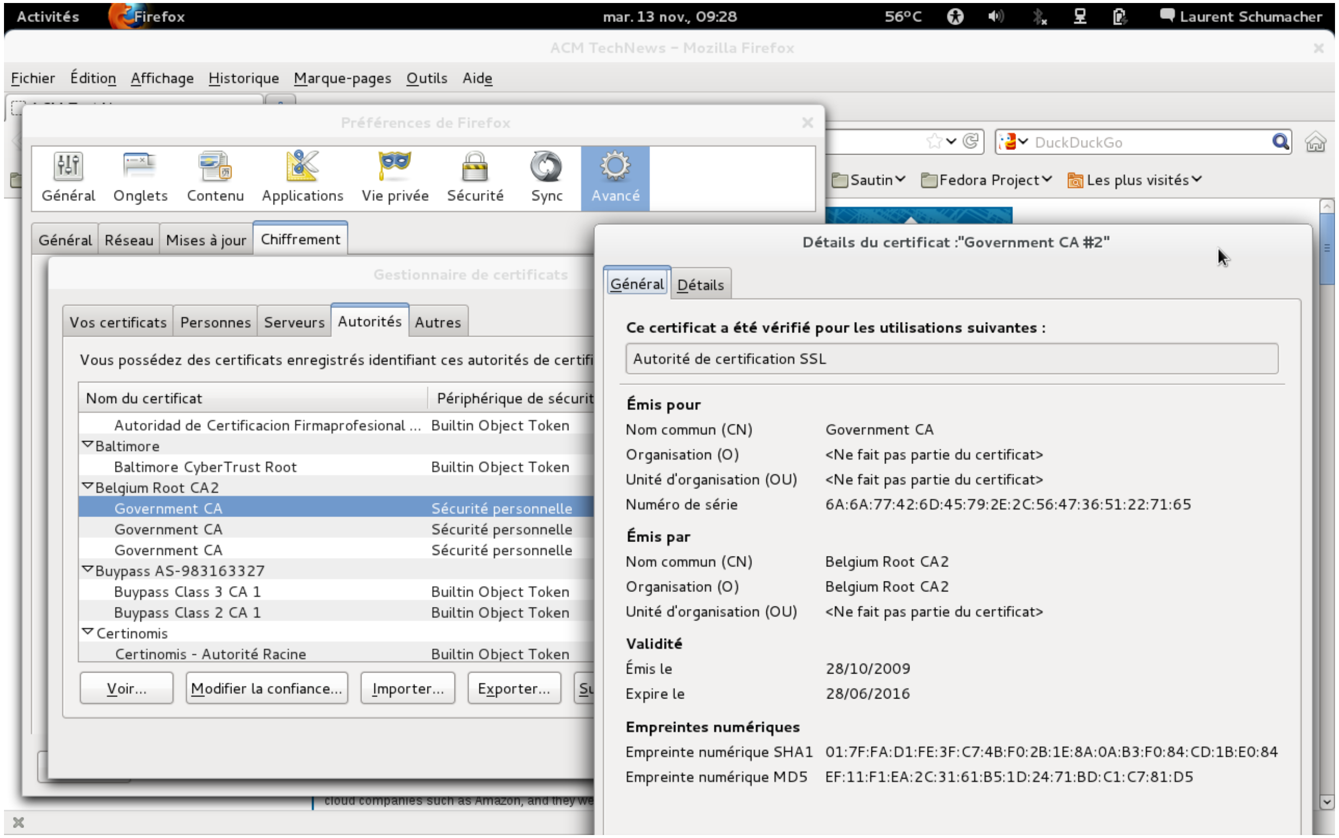
- When Alice wants Bob's public key:
  - She gets Bob's certificate, either directly from Bob or elsewhere
  - She applies CA's public key to Bob's certificate to extract Bob's valid public key



Key distribution and certification

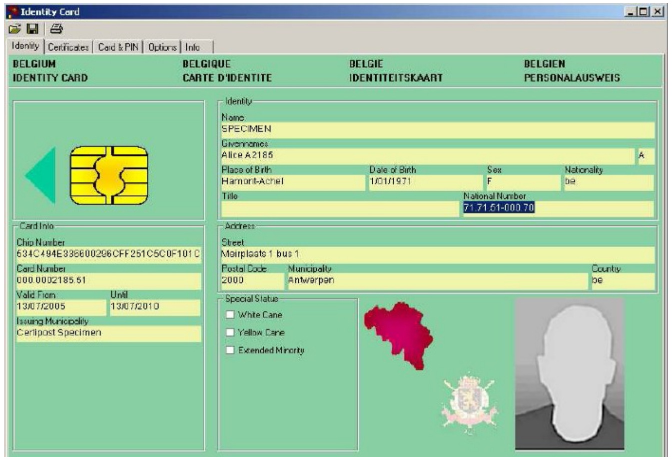
Certification Authority (CA) – X.509 standard

- X.509 = ITU-T standard for Public Key Infrastructure (PKI)



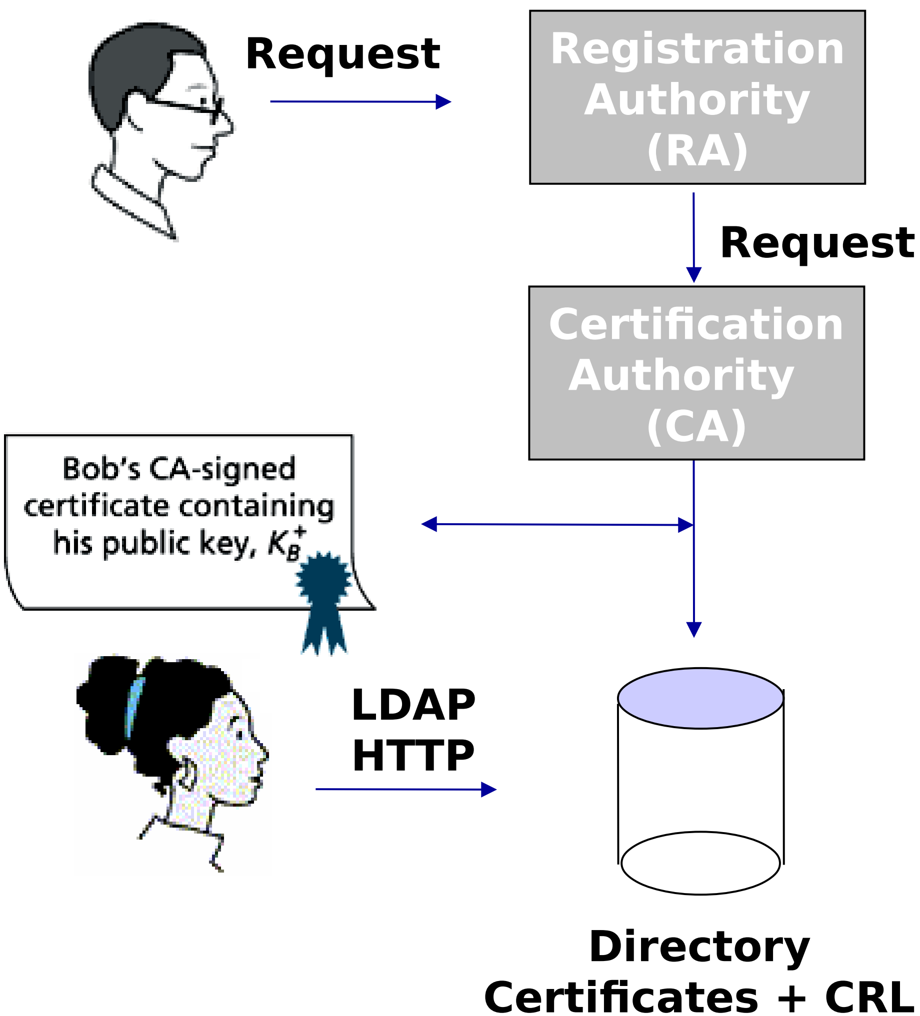
Key distribution and certification
Belgian eID Card

- Only for authentification and signature, not cyphering
- X.509 certificates stored on chip



	Private Key (Java Object)	Public Key (Java Object)	X.509 Certificates (Transparent file)
Basic	PrK#1		
Authentication	PrK#2	In Cert#2	Cert#2
Non-repudiation	PrK#3	In Cert#3	Cert#3
Certification Authority (CA)		In Cert#4	Cert#4
Commune		PuK#5	
Root			Cert#6
CA Role		PuK#7	
RN			Cert#8

Key distribution and certification
Public Key Infrastructure (PKI)



- PKI = authorities + directory + procedures (creation, update, revocation)
- Belgian eID
  - Publication of Certificate Revocation Lists (CRL)
  - Online Certificate Status Protocol (OCSP) service

Summary

- Computationally secure
- Cryptography
  - Symmetric key encryption: DES, AES
  - Public key encryption: RSA, ECC
  - Hash functions
- Key distribution and certification