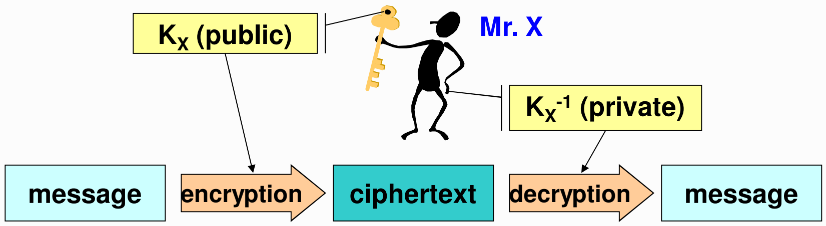
Asymmetric cryptography

Asymmetric (block) ciphers

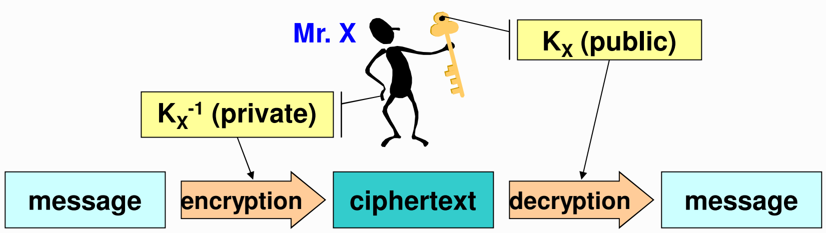
- Use key pairs
  - One private key (personal, not transmittable)
  - One public key
- Allow
  - Confidentiality without any previous exchange of secrets
  - Authentication
    - Of contents (data integrity)
    - Of origin (source authentication, or digital signature)
- Disadvantages
  - Performance (usually very inefficient and memory consuming)
- Advantages
  - N peers requiring pairwise, secret interaction ⇒ N key pairs
- Problems
  - Distribution of public keys
  - Lifetime of key pairs

Confidentiality



- Only the key pair of the recipient is involved
  - $C = E(K, P)$ $P = D(K^{-1}, C)$
  - To send something with confidentiality to X is only required to know X's public key (K_X)
- There is no source authentication
  - X has no means to know who produced the ciphertext
  - If K_X is really public, then everybody can do it

Source authentication



- Only the key pair of the originator is involved
  - $C = E(K^{-1}, P) P = D(K, C);$
  - Only X knows K_X^{-1} that produced C
- There is no confidentiality
  - Anyone knowing the public key of the originator (K_X) can decrypt C
  - If K_X is really public, then everybody can do it

Asymmetric (block) ciphers

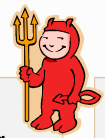
- Approaches: complex mathematic problems
  - Discrete logarithms of large numbers
  - Integer factorization of large numbers
  - Knapsack problems
- Most common algorithms
  - RSA
  - ElGamal
  - Elliptic curves (ECC)
- Other techniques with asymmetric key pairs
  - Diffie-Hellman (key agreement)

Diffie-Hellman key agreement

q (large prime)
α (primitive root mod q)

$a \leftarrow$ random		$b \leftarrow$ random
$Y_a = \alpha^a \bmod q$		$Y_b = \alpha^b \bmod q$
send / publish Y_a	$Y_a \rightarrow$	send / publish Y_b
receive Y_b	$\leftarrow Y_b$	receive Y_a
$K_{ab} = Y_b^{\ a} \bmod q$		$K_{ab} = Y_a^{\ b} \bmod q$

Diffie-Hellman key agreement: Man-in-the-Middle (MitM) attack



Alice	Attacker	Bob
$a \leftarrow$ random	$c \leftarrow$ random	$b \leftarrow$ random
$Y_a = \alpha^a \ mod\ q$	$Y_c = \alpha^c \ mod\ q$	$Y_b = \alpha^b \ mod\ q$
send Y_a	send Y_c	send Y_b
receive Y_c	receive Y_a, Y_b	receive Y_c
$K_{ac} = Y_c^{\,a} \ mod\ q$	$K_{ac} = Y_a^{\,c} \ mod\ q$	
	$K_{cb} = Y_b^{\,c} \ mod\ q$	$K_{cb} = Y_c^{\,b} \ mod\ q$

$Y_a \rightarrow$ $\quad$ $\leftarrow Y_c$ $\quad$ $\leftarrow Y_b$ $\quad$ $Y_c \rightarrow$

RSA (Rivest, Shamir, Adelman)

- Published in 1978
- Computational complexity
  - Discrete logarithm
  - Integer factoring
- Operations and keys
  - $K \equiv (e, n)$
  - $K^{-1} \equiv (d, n)$
  - $C = P^e \bmod n \quad P = C^d \bmod n$
  - $C = P^d \bmod n \quad P = C^e \bmod n$
- Key selection
  - Large n (hundreds or thousands of bits)
  - $n = p \times q$ p and q being large (secret) prime numbers
  -
  - Chose an e co-prime with $(p-1)\times(q-1)$
  - Compute d such that $e \times d \equiv 1 \bmod (p-1)\times(q-1)$
  - Discard p and q
  - The value of d cannot be computed out of e and n
    - Only from p and q

RSA: example

- $p = 5$ $q = 11$ (small primes)
  - $n = p \times q = 55$
  - $(p-1) \times (q-1) = 40$
- $e = 3$
  - Co-prime with 40
- $d = 27$
  - $e \times d \equiv 1 \bmod 40$
- $P = 26$ (note that $P, C \in [0, n-1]$)
  - $C = P^e \bmod n = 26^3 \bmod 55 = 31$
  - $P = C^d \bmod n = 31^{27} \bmod 55 = 26$

ElGamal

- Published by El Gamal in 1984
- Similar to RSA
  - But using only the discrete logarithm complexity
- A variant is used for digital signatures
  - DSA (Digital Signature Algorithm)
  - US Digital Signature Standard (DSS)
- Operations and keys (for signature handling)
  - $\beta = \alpha^x \bmod p$ $K = (\beta, \alpha, p)$ $K^{-1} = (x, \alpha, p)$
  - k random, $k \cdot k^{-1} \equiv 1 \bmod (p-1)$
  - Signature of M: (γ,δ) $\gamma = \alpha^k \bmod p$ $\delta = k^{-1} (M - x\gamma) \bmod (p-1)$
  - Validation of signature over M: $\beta^\gamma \gamma^\delta \equiv \alpha^M \pmod p$
- Problem
  - Knowing k reveals x out of δ
  - k must be randomly generated and remain secret

Elliptic curve

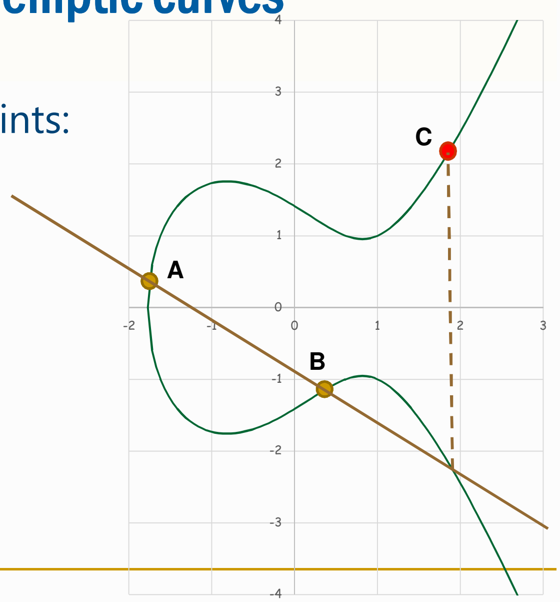
- ▷ A curve described by an equation

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

- ▷ Curves of this kind are symmetric to the X axis
  - And don't have solution for all x values

Operations on elliptic curves

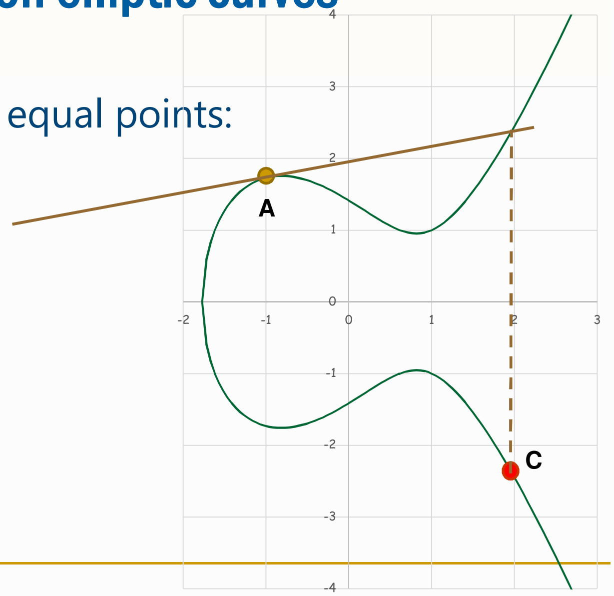
- ▷ Sum of two points:
  - C = A + B



Operations on elliptic curves

▷ Sum of two equal points:

- C = 2A



EC over finite fields

▷ A set of points satisfying the equation

$$y^2 = x^3 + ax + b \ (mod\ q)$$

- The curve also includes a point O at infinity

▷ All x and y values must belong to $[0, q-1]$

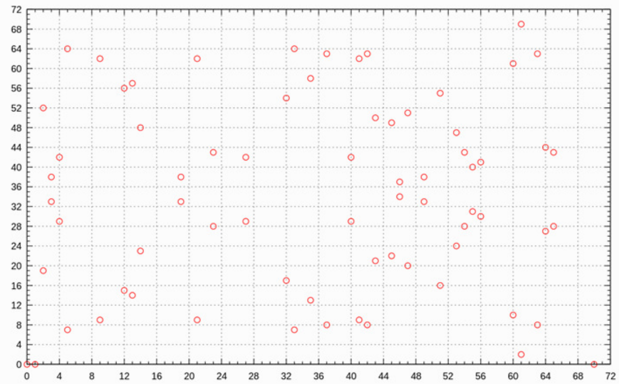
▷ q must be equal to

- p^k, for a prime p (prime finite field $\mathbb{F}_{p^k}$)
- 2^m, for a prime m (binary finite field $\mathbb{F}_{2^m}$)

▷ The elliptic curve is denominated $E(\mathbb{F}_q)$

EC over finite fields: example

$$y^2 = x^3 - x \ (mod\ 71)$$



From https://en.wikipedia.org/wiki/Elliptic_curve

EC discrete logarithm problem

▷ Given an elliptic curve $E(\mathbb{F}_p)$,
a point G on that curve,
a point P which is an integer multiple of G,

find the integer x such that
xG = P

▷ For cryptographic operations, x will be the private key and P the public key

EC cryptography (ECC): curves' definition

- Prime p → (p, a, b, G, n, h)
  - Constants a and b of the EC equation
  - A generator point (or base point) G
  - The order n of G
    - Normally prime
  - A (small) co-factor h
    - Given by $\frac{1}{n}\#E(\mathbb{F}_p)$

EC Diffie-Hellman (ECDH)

- Alice and Bob agree on EC curve
  - (p, a, b, G, n, h)
- Alice chooses a random α
  - And publishes $\mathrm{A} = \alpha\mathrm{G}$
- Bob chooses a random β
  - And publishes $\mathrm{B} = \beta\mathrm{G}$
- Both Alice and Bob compute K
  - $\mathrm{K} = \alpha\mathrm{B}$ $\quad$ $\mathrm{K} = \beta\mathrm{A}$ $\quad$ $\mathrm{K} = \alpha\beta\mathrm{G}$

Public key encryption with EC

▷ DH-based, not like RSA
- Different from RSA

▷ Hybrid encryption
- Target public DH value: T — T = τG
- Source new private DH value: σ — S = σG
- K = σ T
- Encrypt message with K (symmetric encryption)
- Send source public DH value S along w/ message
- Target computes K as K = τS

Recommended curves

Length of n (bits)	p (bits)	m (bits)
161 - 223	192	163
224 - 255	224	233
256 - 383	256	283
384 - 511	384	409
≥ 512	521	571

▷ NIST, 1999
- 5 P curves over prime fields $\mathbb{F}_p$
  - $y^2 = x^3 - 3x + b$
- 5 B curves over binary fields $\mathbb{F}_{2^m}$
  - $y^2 + xy = x^3 + x^2 + b$
- b randomly generated
  - SHA-1 hash of a seed
- 5 K (Koblitz) curves over binary fields $\mathbb{F}_{2^m}$
  - $y^2 + xy = x^3 + ax^2 + 1$

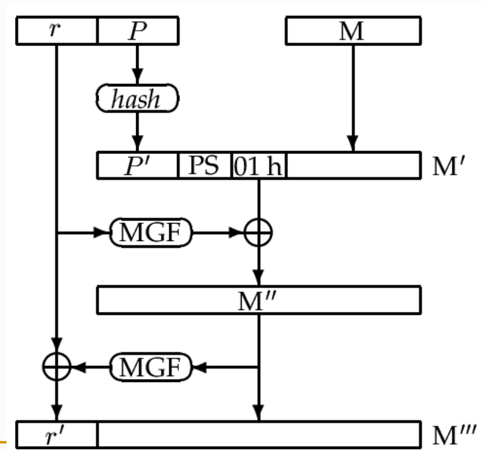
Recommended curves

- IETF
  - Daniel Bernstein's Curve25519
    - $y^2 = x^3 + 486662\,x^2 + x \ (mod\ q)$
    - $q = 2^{255} - 19$
  - Curve448
    - $y^2 = x^3 + 15632\,x^2 + x \ (mod\ q)$
    - $q = 2^{448} - 2^{224} - 1$

Randomization of asymmetric encryptions

- Non-deterministic (unpredictable) result of asymmetric encryptions
  - N encryptions of the same value, with the same key, should yield N different results
  - Goal: prevent trial & error discovery of encrypted values
- Technics
  - Concatenation of values to encrypt with two values
    - A fixed one (for integrity control)
    - A random one (for randomization)
  - PKCS #1
  - OAEP (Optimal Asymmetric Encryption Padding)

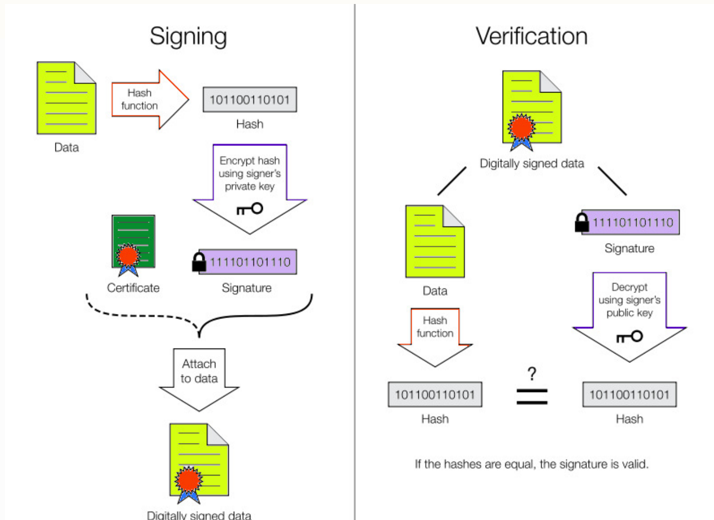
Randomization of asymmetric encryptions: OAEP (Optimal Asymmetric Encryption Padding)



Digital signatures

- Goal
  - Authenticate the contents of a document
    - Ensure its integrity
  - Authenticate its author
    - Ensure the identity of the creator/originator
  - Prevent origin repudiation
    - Genuine authors cannot deny authorship
- Approaches
  - Asymmetric encryption
  - Digest functions (only for performance)
- Algorithms
  - Signing: $A_x(doc) = info + E(K_x^{-1}, digest(doc+info))$
  - Verification: $info \rightarrow K_x$
    $D(K_x, A_x(doc)) \equiv digest(doc + info)$

Signing / verification diagrams



wikipedia, http://en.wikipedia.org/wiki/Digital_signature

Digital signature on a mail:
Multipart content, signature w/ certificate

```
From - Fri Oct 02 15:37:14 2009
[…]
Date: Fri, 02 Oct 2009 15:35:55 +0100
From: =?ISO-8859-1?Q?Andr=E9_Z=FAquete?= <andre.zuquete@ua.pt>
Reply-To: andre.zuquete@ua.pt
Organization: IEETA / UA
MIME-Version: 1.0
To: =?ISO-8859-1?Q?Andr=E9_Z=FAquete?= <andre.zuquete@ua.pt>
Subject: Teste
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature"; micalg=sha1; boundary="------------ms050405070101010502050101"

This is a cryptographically signed message in MIME format.

--------------ms050405070101010502050101
Content-Type: multipart/mixed;
 boundary="------------060802050708070409030504"

This is a multi-part message in MIME format.
--------------060802050708070409030504
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

Corpo do mail

--------------060802050708070409030504-
--------------ms050405070101010502050101
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: S/MIME Cryptographic Signature

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAQAAoIIamTCC
BUkwggSyoAMCAQICBAcnIaEwDQYJKoZIhvcNAQEFBQAwdTELMAkGA1UEBhMCVVMxGDAWBgNV
[…]
KoZIhvcNAQEBBQAEgYCofks852BV77NVuww53vSxO1XtI2JhC1CDlu+tcTPoMD1wq5dc5v40
Tgsaw0N8dqgVLk8aC/CdGMbRBu+J1LKrcVZa+khnjjtB66HhDRLrjmEGDNttrEjbqvpd2QO2
vxB3iPTlU+vCGXo47e6GyRydqTpbq0r49Zqmx+IJ6Z7iigAAAAAAAA==
--------------ms050405070101010502050101--
```

Blind signatures

- Signatures made by a "blinded" signer
  - Signer cannot observe the contents it signs
  - Similar to a handwritten signature on an envelope containing a document and a carbon-copy sheet
- Useful for ensuring anonymity of the signed information holder, while the signed information provides some extra functionality
  - Signer X knows who requires a signature (Y)
  - X signs T_1, but Y afterwards transforms it into a signature over T_2
    - Not any T_2, a specific one linked to T_1
  - Requester Y can present T_2 signed by X
    - But it cannot change T_2
    - X cannot link T_2 to the T_1 that it observed when signing

Chaum Blind Signatures

- Implementation using RSA
  - Blinding
    - Random blinding factor K
    - $k \times k^{-1} \equiv 1 \pmod N$
    - $m' = k^e \times m \bmod N$
  - Ordinary signature (encryption w/ private key)
    - $A_x(m') = (m')^d \bmod N$
  - Unblinding
    - $A_x(m) = k^{-1} \times A_x(m')$ mod