Systems' Security | *Segurança de Sistemas*

Asymmetric Cryptography

Miguel Frade

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Overview

Learning Objectives

After this chapter, you should be able to:

1. Present an overview of the basic principles of public-key cryptosystems
2. Explain the two distinct uses of public-key cryptosystems
3. List and explain the requirements for a public-key cryptosystem.

Introduction

Misconceptions about public-key encryption

- public-key encryption is more secure than is symmetric encryption from cryptanalysis

Misconceptions about public-key encryption

- public-key encryption is more secure than is symmetric encryption from cryptanalysis $\leftarrow$ **wrong**
  - the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher

Misconceptions about public-key encryption

- public-key encryption is more secure than is symmetric encryption from cryptanalysis ← **wrong**
  - the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher
- public-key encryption is a general-purpose technique that has made symmetric encryption obsolete

Misconceptions about public-key encryption

- public-key encryption is more secure than is symmetric encryption from cryptanalysis ← **wrong**
  - the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher
- public-key encryption is a general-purpose technique that has made symmetric encryption obsolete ← **wrong**
  - symmetric encryption is faster and public-key cryptography are best suited for key management and digital signatures

Misconceptions about public-key encryption

- public-key encryption is more secure than is symmetric encryption from cryptanalysis $\leftarrow$ **wrong**
  - the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher
- public-key encryption is a general-purpose technique that has made symmetric encryption obsolete $\leftarrow$ **wrong**
  - symmetric encryption is faster and public-key cryptography are best suited for key management and digital signatures
- public key distribution is trivial

Misconceptions about public-key encryption

- public-key encryption is more secure than is symmetric encryption from cryptanalysis $\leftarrow$ **wrong**
  - the security of any encryption scheme depends on the length of the key and the computational work involved in breaking a cipher
- public-key encryption is a general-purpose technique that has made symmetric encryption obsolete $\leftarrow$ **wrong**
  - symmetric encryption is faster and public-key cryptography are best suited for key management and digital signatures
- public key distribution is trivial $\leftarrow$ **wrong**
  - distribution of public keys does not require confidentiality, however to assure the authenticity of the public keys some form of protocol is needed and it is complex

Genesis of asymmetric cryptography

Address two of the most difficult problems associated with symmetric encryption:

1. key distribution
    - why develop impenetrable cryptosystems, if their users were forced to share their keys with a centralized system that could be compromised

Genesis of asymmetric cryptography

Address two of the most difficult problems associated with symmetric encryption:

1. key distribution
    - why develop impenetrable cryptosystems, if their users were forced to share their keys with a centralized system that could be compromised
2. digital signatures
    - for widespread use on commercial and private purposes, electronic messages and documents would need the equivalent of signatures used in paper documents

Genesis of asymmetric cryptography

In 1976, Whitfield Diffie and Martin Hellman created the first public-key cryptographic algorithm

- Diffie-Hellman – for symmetric key exchange

Requirements that public-key algorithm must fulfill

1. it is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)

Requirements that public-key algorithm must fulfill

1. it is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)
2. it is computationally easy for a sender A to encrypt a message M: $C = E_{PU_b}(M)$

Requirements that public-key algorithm must fulfill

1. it is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)
2. it is computationally easy for a sender A to encrypt a message M: $C = E_{PU_b}(M)$
3. it is computationally easy for the receiver B to decrypt the ciphertext using the private key to recover the original message: $M = D_{PR_b}(C) \Leftrightarrow M = D_{PR_b}(E_{PU_b}(M))$

Requirements that public-key algorithm must fulfill

1. it is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)
2. it is computationally easy for a sender A to encrypt a message M: $C = E_{PU_b}(M)$
3. it is computationally easy for the receiver B to decrypt the ciphertext using the private key to recover the original message: $M = D_{PR_b}(C) \Leftrightarrow M = D_{PR_b}(E_{PU_b}(M))$
4. it is computationally infeasible to derive the private key PR_b from the public key PU_b

Requirements that public-key algorithm must fulfill

1. it is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)
2. it is computationally easy for a sender A to encrypt a message M: $C = E_{PU_b}(M)$
3. it is computationally easy for the receiver B to decrypt the ciphertext using the private key to recover the original message: $M = D_{PR_b}(C) \Leftrightarrow M = D_{PR_b}(E_{PU_b}(M))$
4. it is computationally infeasible to derive the private key PR_b from the public key PU_b
5. it is computationally infeasible to to recover the original message M knowing the public key PU_b, and a ciphertext C

Public-Key Cryptanalysis

- public-key encryption algorithms are also vulnerable to a brute-force attack
    - the countermeasure is increase key size
    - but these algorithms depend on an invertible mathematical function whose complexity increases exponentially with the size of the key
    - the key size must be large enough to make brute-force attack impractical but small enough for practical encryption and decryption

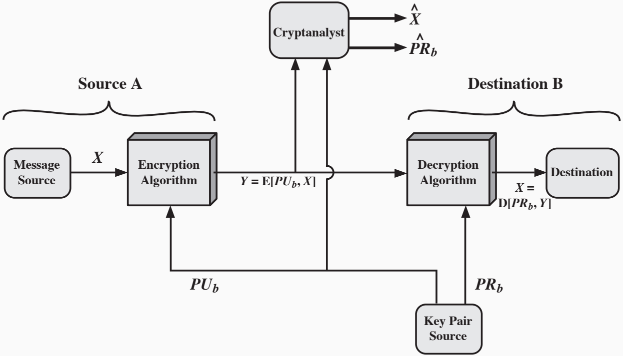
Public-Key Cryptanalysis

- public-key encryption algorithms are also vulnerable to a brute-force attack
  - the countermeasure is increase key size
  - but these algorithms depend on an invertible mathematical function whose complexity increases exponentially with the size of the key
  - the key size must be large enough to make brute-force attack impractical but small enough for practical encryption and decryption
- find a way to compute the private key from the public key
  - to date, it has not been mathematically proven that this form of attack is infeasible
  - the history of cryptanalysis shows that, given the time, a problem that seems insoluble can be found to have a solution

Security Services that can be achieved with public-key algorithms

- confidentiality
- non-repudiation
  - includes integrity and authenticity
- both confidentiality and non-repudiation

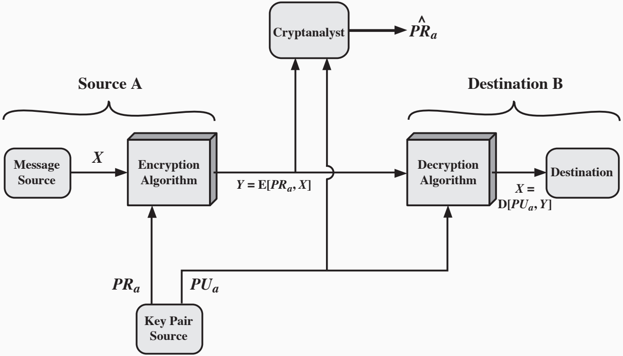
Model of asymmetric encryption – **confidentiality**



Short representation

$$A \rightarrow B: \quad E_{PU_B}(M)$$

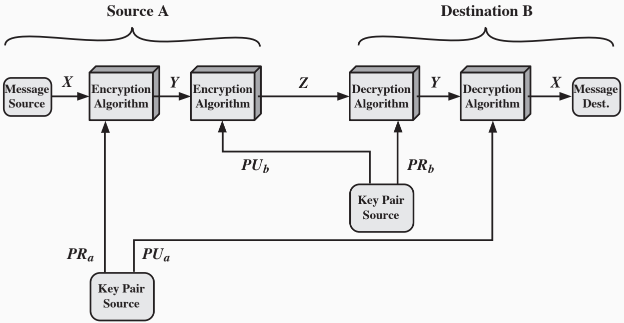
Model of asymmetric encryption – **non-repudiation**



Short representation

$$A \rightarrow B : \quad E_{PR_A}(M)$$

Model of asymmetric encryption – **confidentiality and non-repudiation**



Short representation

$A \rightarrow B: \quad E_{PU_B}(E_{PR_A}(M))$

Public-key Algorithms

Public-key algorithms can be used for:

Key exchange two entities cooperate to exchange a session key (secret key for symmetric encryption generated for a session and valid for a short period of time)

Digital signature the sender “signs” a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message

Encryption/decryption the sender encrypts a message with the recipient’s public-key, and the recipient decrypts the message with the recipient’s private-key.

Applications for Public-Key Algorithms

Algorithm	Encryption	Signature	Key exchange
Diffie–Hellman	—	—	Yes
DSA	—	Yes	—
RSA	Yes	Yes	Yes
ElGamal	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes

2. **Asymmetric algorithms** can be dived into:

    2.1 Prime factorization

    - large key sizes are required, minimum recommended is 2 048 bits

2. **Asymmetric algorithms** can be dived into:

   2.1 Prime factorization
   - large key sizes are required, minimum recommended is 2 048 bits

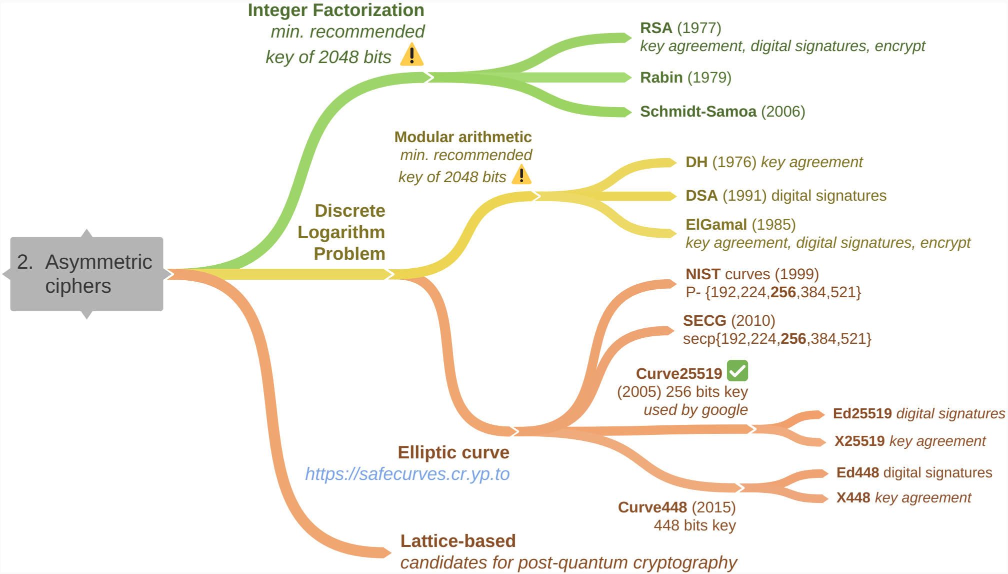
   2.2 Discrete logarithm
   - Modular Arithmetic
     - large key sizes are required, minimum recommended is 2 048 bits
   - Elliptic Curve
     - shorter key lengths, minimum recommended is 256 bits
     - faster than Modular Arithmetic and Prime factorization, except to verify digital signatures

2. **Asymmetric algorithms** can be dived into:

   2.1 Prime factorization
   - large key sizes are required, minimum recommended is 2 048 bits

   2.2 Discrete logarithm
   - Modular Arithmetic
     - large key sizes are required, minimum recommended is 2 048 bits
   - Elliptic Curve
     - shorter key lengths, minimum recommended is 256 bits
     - faster than Modular Arithmetic and Prime factorization, except to verify digital signatures

   2.3 Lattice-based
   - candidates for post-quantum cryptography


2. Asymmetric ciphers
Integer Factorization
min. recommended
key of 2048 bits
RSA (1977)
key agreement, digital signatures, encrypt
Rabin (1979)
Schmidt-Samoa (2006)
Discrete Logarithm Problem
Modular arithmetic
min. recommended
key of 2048 bits
DH (1976) key agreement
DSA (1991) digital signatures
ElGamal (1985)
key agreement, digital signatures, encrypt
NIST curves (1999)
P- {192,224,256,384,521}
SECG (2010)
secp{192,224,256,384,521}
Curve25519
(2005) 256 bits key
used by google
Ed25519 digital signatures
X25519 key agreement
Ed448 digital signatures
X448 key agreement
Curve448 (2015)
448 bits key
Elliptic curve
https://safecurves.cr.yp.to
Lattice-based
candidates for post-quantum cryptography

Tools that use asymmetric algorithms

- digital signatures:
  - Portuguese eID software, aCCinaPDF, Acrobat Reader DC, ...
  - email clients, *i. e.* Thunderbird, Outlook, ...
- symmetric key distribution:
  - TLS, IPSec, SSH, ...
- symmetric key encapsulation (for encryption):
  - minilock.org, email clients with encryption (S/MIME, OpenPG), ...

Comparing different public-key algorithms

- never use key size to compare different algorithms
- computational effort to brute-force an algorithm
- measured in Millions of Instructions Per Second over a year (MIPS-Year)
  - 1 MIPS-year $= 10^6$ instructions/second $\times$ 86 400 seconds/day $\times$ 365 days/year $= 3.15 \times 10^{13}$ instructions

Comparing different public-key algorithms

- never use key size to compare different algorithms
- computational effort to brute-force an algorithm
- measured in Millions of Instructions Per Second over a year (MIPS-Year)
  - 1 MIPS-year $= 10^6$ instructions/second $\times$ 86 400 seconds/day $\times$ 365 days/year $= 3.15 \times 10^{13}$ instructions

Time to break (MIPS-Year)	RSA Key size (bits)	ECC key size (bits)
10^4	512	106
10^8	768	132
10^{11}	1 024	160
10^{20}	2 048	210
10^{78}	21 000	600

Comparable strengths to resist a brute-force attack

Bits	Symmetric	RSA/DH/DSA	ECC
80	2DES	1 024	160 – 223
112	3DES	2 048	224 – 255
128	AES-128	3 072	256 – 383
192	AES-192	7 680	384 – 511
256	AES-256	15 360	$\geqslant$512

Source: NIST SP 800-57 Pt. 1 Rev. 4 (`https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final`)

Diffie-Hellman Algorithm

Key exchange

See Youtube Video

Key exchange

- the parties agree on two numbers, a prime number p and a primitive root of that prime a
- public values in blue, and secret/private values in red
- for example $\quad p = 23 \quad$ and $\quad a = 5$

Key exchange

- the parties agree on two numbers, a prime number p and a primitive root of that prime a
- public values in blue, and secret/private values in red
- for example $p = 23$ and $a = 5$

Alice (A)

- private key is a random integer $0 \leqslant PR_A < p$
- public key is $PU_A = a^{PR_A} \bmod p$
  - $PR_A = 4$
  - $PU_A = 5^4 \bmod 23 = 4$

Bob (B)

- private key is a random integer $0 \leqslant PR_B < p$
- public key is $PU_B = a^{PR_B} \bmod p$
  - $PR_B = 3$
  - $PU_B = 5^3 \bmod 23 = 10$

Key exchange

- the parties agree on two numbers, a prime number p and a primitive root of that prime a
- public values in blue, and secret/private values in red
- for example $p = 23$ and $a = 5$

Alice (A)

- private key is a random integer $0 \leqslant PR_A < p$
- public key is $PU_A = a^{PR_A} \bmod p$
  - $PR_A = 4$
  - $PU_A = 5^4 \bmod 23 = 4$
- secret key
  - $k = {PU_B}^{PR_A} \bmod p$
  - $k = 10^4 \bmod 23 = 18$

Bob (B)

- private key is a random integer $0 \leqslant PR_B < p$
- public key is $PU_B = a^{PR_B} \bmod p$
  - $PR_B = 3$
  - $PU_B = 5^3 \bmod 23 = 10$
- secret key
  - $k = {PU_A}^{PR_B} \bmod p$
  - $k = 4^3 \bmod 23 = 18$

Key exchange attack

- an adversary has access to p, a, PU_A, and PU_B
- in order to get the key k he has to calculate the private key of A or B
  - $PR_A = dlog_{a,p}(PU_A)$, or
  - $PR_B = dlog_{a,p}(PU_B)$
- and then calculate the k like the parties A, or B do

Key exchange attack

- an adversary has access to p, a, PU_A, and PU_B
- in order to get the key k he has to calculate the private key of A or B
  - $PR_A = dlog_{a,p}(PU_A)$, or
  - $PR_B = dlog_{a,p}(PU_B)$
- and then calculate the k like the parties A, or B do

Discrete Logarithm (*dlog*)

The discrete logarithm (*dlog*) is computationally infeasible to calculate for very large numbers

Terminology

Asymmetric Keys two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Asymmetric Keys two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Public Key Certificate A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key.

Asymmetric Keys two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Public Key Certificate A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key.

Asymmetric Cryptographic Algorithm A cryptographic algorithm that uses two related keys, a public key and a private key. Deriving the private key from the public key is computationally infeasible.

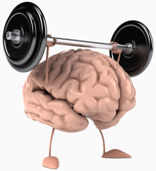
Asymmetric Keys two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Public Key Certificate A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key.

Asymmetric Cryptographic Algorithm A cryptographic algorithm that uses two related keys, a public key and a private key. Deriving the private key from the public key is computationally infeasible.

Public Key Infrastructure (PKI) A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.

Exercises

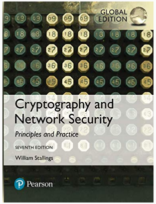


1. Consider the following expression: $A \rightarrow B: \quad E_{PR_B}(E_{PU_A}(M)) = C$
   Is it possible to compute C? Justify.

2. Consider the following DH parameters: $\quad p = 353 \quad$ and $\quad a = 3$.
   A chooses $PR_A = 97$ as his private key and B chooses $PR_B = 233$.
   Calculate the secret key k using the DH algorithm.

   Tip: use `https://www.wolframalpha.com/` for the calculations.

Questions?



Chapters 9 of

William Stallings, Cryptography and Network Security: Principles and Practice, Global Edition, 2016