RSA

Gaspare FERRARO
CyberSecNatLab
Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

This presentation is licensed under the Creative Commons BY-NC License

To view a copy of the license, visit:

http://creativecommons.org/licenses/by-nc/3.0/legalcode

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

Goal

- Give the definition of public-key encryption
- Present the RSA public-key cryptosystem
- Show how to choose the parameters

Prerequisites

- Lectures:
  - *CR_0.1 - Number Theory and modular arithmetic*
  - *CR_1.1 – Introduction to cryptography and classical ciphers*
  - *CR_2.1 – Key Exchange & DH*

Outline

- Introduction to public-key cryptography
- RSA textbook scheme
- RSA-CRT
- Choose parameters

Outline

- Introduction to public-key cryptography
- RSA textbook scheme
- RSA-CRT
- Choose parameters

Recall: Cryptography

- Two main methods:
  - *Symmetric key* - Single key
  - *Public key* - Double key

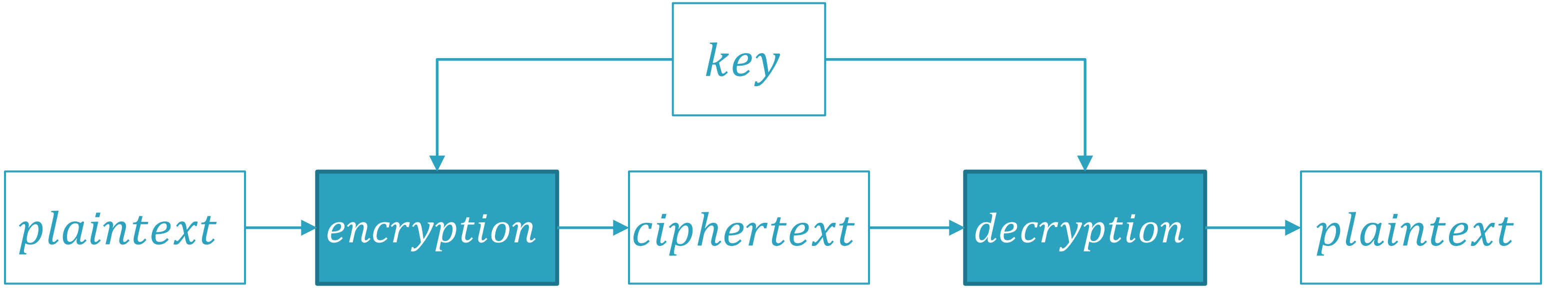
Recall: Cryptography

- Two main methods:
  - *Symmetric key* - Single key
  - *Public key* - Double key

Seen in:
- *CR_1.2 – XOR cipher*
- *CR_1.3 – Block ciphers*
- *CR_1.4 – Stream ciphers*

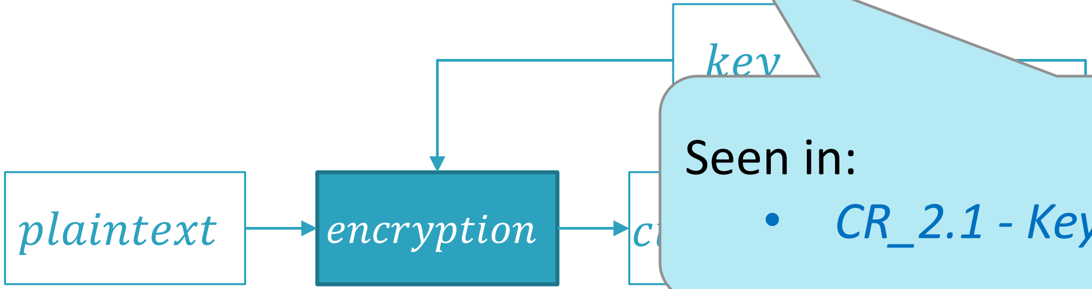
Recall: Symmetric key cryptography

- Requires that both sender and recipient know the same key
- An issue is how they do share it without meeting



Recall: Symmetric key cryptography

- Requires that both sender and recipient know the same key
- An issue is how they do share it without meeting



Seen in:

- *CR_2.1 - Key Exchange & DH*

Recall: public-key Encryption

- Require the use of two keys:
  - a public key, which can be known by anyone and can be used to encrypt messages and verify signatures
  - a corresponding private key, known only to the recipient, used to decrypt messages and to sign them
- Rely an asymmetric exchange: whoever encodes messages or verifies signatures cannot decode messages or create signatures
- Also called {double key, asymmetric} encryption

Asymmetric Encryption

- Developed to address two important issues:
  - key distribution: ensuring secure communications with a personal key without depending on a key distribution center and without trusting the behavior of others
  - digital signatures: verify that a message comes intact from the declared sender
- Complements rather than replaces symmetric encryption
- Is based on properties guaranteed by number theory rather than on the use of permutations and substitutions

Asymmetric vs symmetric encryption

- Symmetric encryption: same algorithm used to encrypt and decrypt with the secret same key. Key and algorithm are shared by sender and receiver
  - Almost impossible to decrypt a message if only the algorithm and the cipher text are known
- Asymmetric encryption: same algorithm used to encrypt and decrypt, but two keys are used: one to encrypt, the other to decrypt. Sender and receiver must each have a key that pairs with the other (not the same)
  - Almost impossible to decrypt a message if only the algorithm, the cipher text, and one of the keys are known

Principles of asymmetric cryptography

- A distinction is made between the keys of the subjects:
  - public key: publicly disclosed by the subject
  - private key: kept secret by the subject
- It must be computationally difficult to derive the decryption key knowing the algorithm and the encryption key
- The two keys can be (complementarily) used for encryption/decryption
- Encryption with public key guarantees confidentiality
- Encryption with private key guarantees authentication
- With an appropriate mix we can also guarantee messages integrity

Principles of asymmetric cryptography

- A distinction is made between the keys of the subjects:
  - public key: publicly disclosed by the subject
  - private key: kept secret by the subject
- It must be computationally difficult to derive the decryption key knowing the algorithm a... the encryption key
- The two keys can be (c... ...rily) used for encryption/decryption
- Encryption w...
- Encryption w...
- With an appropriate mix we can also guarantee messages integrity

"trap-door one-way functions", seen in:

- *CR_0.1 – Number theory and modular arithmetic*

Outline

- Introduction to public-key cryptography
- RSA textbook scheme
- RSA-CRT
- Choose parameters

RSA

- The most famous asymmetric encryption algorithm is RSA, from the authors **R**ivest, **S**hamir, **A**dleman, developed in 1977 and still used in practice today
  - Based on exponentiation of integers modulo n. Very large integers are used (typically 2048 bits)
  - Encryption and decryption are single modular exponentiation operation: exponentiation is easy (requires $O((\log n)^3)$ operations)
  - Security is guaranteed by the cost of factoring large numbers: factoring is difficult (requires $O\left(e^{(\log n)(\log \log n)}\right)$operations)

RSA Key Generation

- A user generates a pair of public/private keys as follows:
  - Randomly chooses two prime numbers: p, q
  - Computes $n = p \times q$ and $\phi(n) = (p-1) \times (q-1)$ Euler's totient
  - Randomly chooses the public key e such that:
    - $1 < e < \phi(n)$ with e and $\phi(n)$ coprime ($\gcd(e, \phi(n)) = 1$)
  - Determines the private key d by solving the equation:
    - $(e \times d)\ mod\ \phi(n) = 1$ with $0 \leq d \leq n$ ($d = e^{-1}\ mod\ \phi(n)$)
  - Share public key $k_{pub} = \{e, n\}$ and keeps private key $k_{priv} = \{d, n\}$

RSA encryption and decryption

- **Public Key** - $k_{pub} = \{e, n\}$ - **Private Key** - key $k_{priv} = \{d, n\}$
- To encrypt a message M, the sender:
  - Gets the recipient's public key $k_{pub} = \{e, n\}$
  - Computes $C = M^e \; mod \; n$, with $0 \leq M < n$
- To decipher the ciphertext C, the recipient:
  - Uses his private key $k_{priv} = \{d, n\}$
  - Computes $M = C^d \; mod \; n$
- The "magic" is due to the fact that $(M^e)^d \; (mod \; n) = M$

Why RSA Works

- Euler's theorem:
  - $a^{\phi(n)}\ mod\ n = 1$ if $\gcd(a, n) = 1$
- In RSA we have:
  - $n = p \times q$ and $\phi(n) = (p-1) \times (q-1)$
- The keys in the pair (e, d) are inverses $mod\ \phi(n)$
  - $e \times d = k \times \phi(n) + 1$ for some integer value of k

Why RSA Works

- Euler's theorem:
  - $a^{\phi(n)}\ mod\ n = 1$ if $\gcd(a, n) = 1$
- Thus, working modulo n, $C^d =$
  - $= M^{e \times d}$ (since $C = M^e$)
  - $= M^{k \times \phi(n)+1}$ since (e, d) are inverses $mod\ \phi(n)$
  - $= M^1 \times \left(M^{\phi(n)}\right)^k$ with simple arithmetic
  - $= M^1 \times 1^k$ for the Euler's theorem
  - $= M^1 = M$

An example in RSA - Key Setup

- Select two primes: $p = 17$ and $q = 11$
- Compute $n = p \times q = 17 \times 11 = 187$
- Compute $\phi(n) = (p-1) \times (q-1) = 16 \times 10 = 160$
- Select e: $GCD(e, 160) = 1; e = 7$
- Determine $d < 160$ such that $(d \times e)\, mod\, 160 = 1$
  - We have $d = 23$ as $23 \times 7 = 161 = 160 + 1$
- Publish public key $k_{pub} = \{e = 7, n = 187\}$
- Keep private key secret $k_{priv} = \{d = 23, n = 187\}$

An example in RSA - En/Decryption

- Public key = $\{e = 7, n = 187\}$
- Private key = $\{d = 23, n = 187\}$
- Given $M = 88$ $(88 < 187)$
- Cipher M:
  - $C = M^e \, mod \, n = 88^7 \, mod \, 187 = 11$
- Decipher C:
  - $M = C^d \, mod \, n = 11^{23} \, mod \, 187 = 88$

Outline

- Introduction to public-key cryptography
- RSA textbook scheme
- RSA-CRT
- Choose parameters

RSA-CRT

- In practice, the public exponent e is chosen to optimize the (encryption) exponentiation operation
- The private exponent d is unfortunately not as convenient as the public one
- We can use the Chinese Remained Theorem (CRT) to optimize also the decryption function

RSA-CRT – Key generation

- Generate all the parameters from RSA textbook schema as seen before (assuming $p > q$)
- Precompute (just once) the following private values:
  - $d_p = d \ (mod \ p - 1)$
  - $d_q = d \ (mod \ q - 1)$
  - $q_{inv} = q^{-1} \ (mod \ p)$

RSA-CRT – Decryption

- To decrypt a ciphertext c we have to execute:
  - $m_1 = c^{d_p} \ (mod\ p)$
  - $m_2 = c^{d_q} \ (mod\ p)$
  - $h = q_{inv} \times (m_1 - m_2) \ (mod\ p)$
  - $m = m_2 + h \times q \ (mod\ n)$
- In practice RSA-CRT is four time more efficiently than the standard decryption operation $m = c^d \ (mod\ n)$
- This thanks to the exponent used (d_p and d_q) which are smaller than d

Outline

- Introduction to public-key cryptography
- RSA textbook scheme
- RSA-CRT
- Choose parameters

Choose parameters

- Choose the correct parameters is essential for the security of RSA
- Some example recommendations for the primes:
  - Primes p and q should be at least 1024 bits
  - Same primes should not be reused for different keys
  - Values of p and q should not be too close to each others

Choose parameters

- The public exponent $e = 65537$ is commonly choose
- This is mainly due to the exponentiation by squaring algorithm used to evaluate $c = m^e \ (mod\ n)$:
- Smaller public exponent (e.g., $e = 3$) are vulnerable

What can go wrong?

- In the next lecture we will present different vulnerabilities, mainly due to some bad choice in the parameters
- For example:
  - How to factorize public modulo
  - Standard attacks in vulnerable scenarios
  - Attacks with RSA oracles

RSA

Gaspare FERRARO
CyberSecNatLab
Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it