RSA cryptosystem (Rivest-Shamir-Adleman)

RSA®

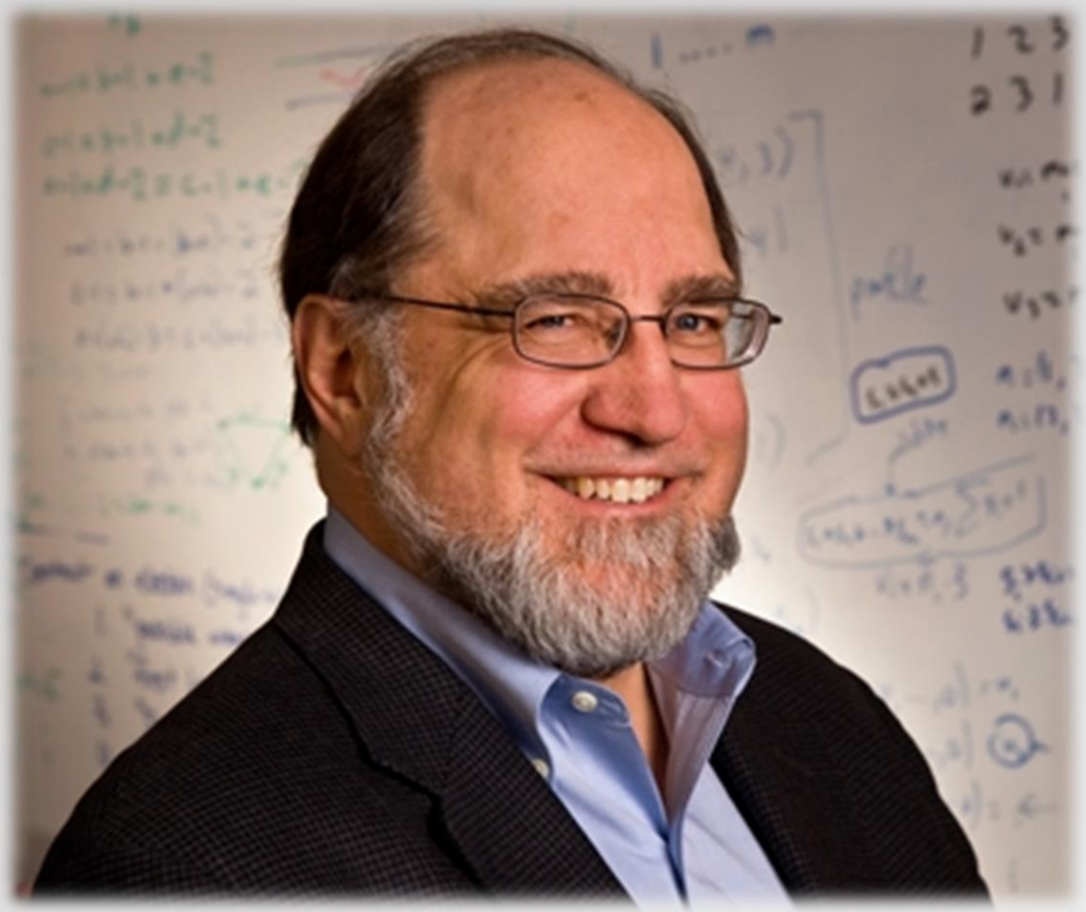
Criptografia 2018/2019 | Prof. Isabel Oitavem
FCT NOVA – 10 de Maio de 2019

Emanuel Carvalho 45466, MIEI
Sara Lobo 45622, MIEB

Clifford Cocks

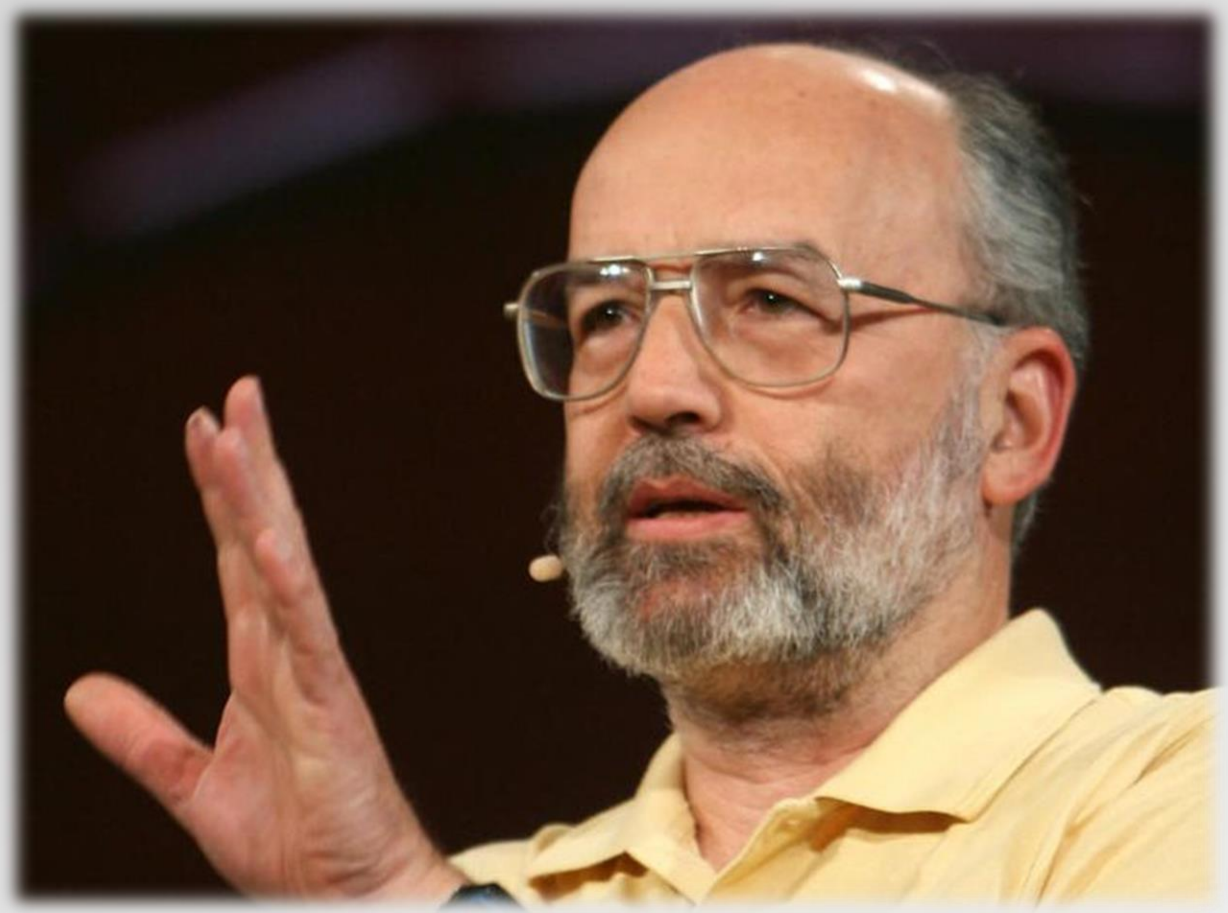
GCHQ

RSA - Ronald Linn Rivest



Massachusetts
Institute of
Technology

RSA - Adi Shamir

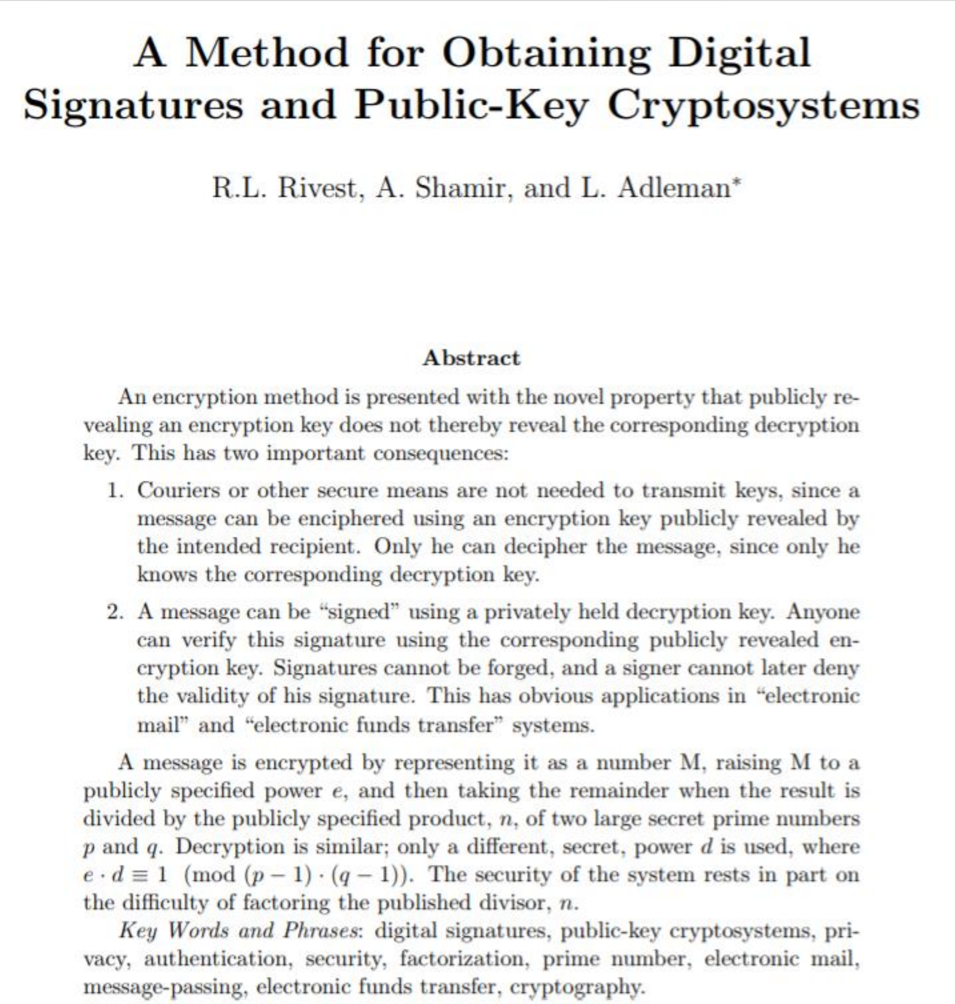


Massachusetts
Institute of
Technology

RSA - Leonard Max Adleman

Massachusetts
Institute of
Technology

Timeline



A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman*

Abstract

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:

1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.
2. A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems.

A message is encrypted by representing it as a number M, raising M to a publicly specified power e, and then taking the remainder when the result is divided by the publicly specified product, n, of two large secret prime numbers p and q. Decryption is similar; only a different, secret, power d is used, where $e \cdot d \equiv 1 \pmod{(p-1)\cdot(q-1)}$. The security of the system rests in part on the difficulty of factoring the published divisor, n.

Key Words and Phrases: digital signatures, public-key cryptosystems, privacy, authentication, security, factorization, prime number, electronic mail, message-passing, electronic funds transfer, cryptography.

≈ 1900 BC
- Non-standard egyptian hieroglyphs

50-60 BC
- Caesar's cipher

1933-45
- Enigma machine

1976
- DES is defined as standard
- Public-key cryptography (Diffie & Hellman)

1977
- Publication of RSA in the September 1977 issue of Scientific American
- NSA objected to the distribution of RSA's full technical report

1978
- Publication of RSA algorithm in the Communications of the ACM

1982
- Comercialization of RSA encryption algorithm

1998
- AES first publication, established by the US NIST in 2001

Before RSA

Modular arithmetic

Euler’s Theorem (generalization of Fermat’s Little Theorem)

Euler’s Totient Funtion (Phi Function)

Chinese Remainder Theorem

Coprime numbers

a and b are **coprime** if they have no factors in common

a and b are **coprime** if $\gcd(a, b) = 1$

<u>Example</u>:

10 = 2 x 5   21 = 3 x 7

Factors of each number

$\begin{matrix} 2 \\ 5 \end{matrix}$!= $\begin{matrix} 3 \\ 7 \end{matrix}$ They have no factors in common $\rightarrow$ 10 and 21 are coprime

Euler's Theorem

Euler's Theorem: Let a and n be $coprime$

$$a^{\varphi(n)} \equiv 1 (mod\, n)$$

$\varphi(n)$: number of positive integers up to n that are coprime to n

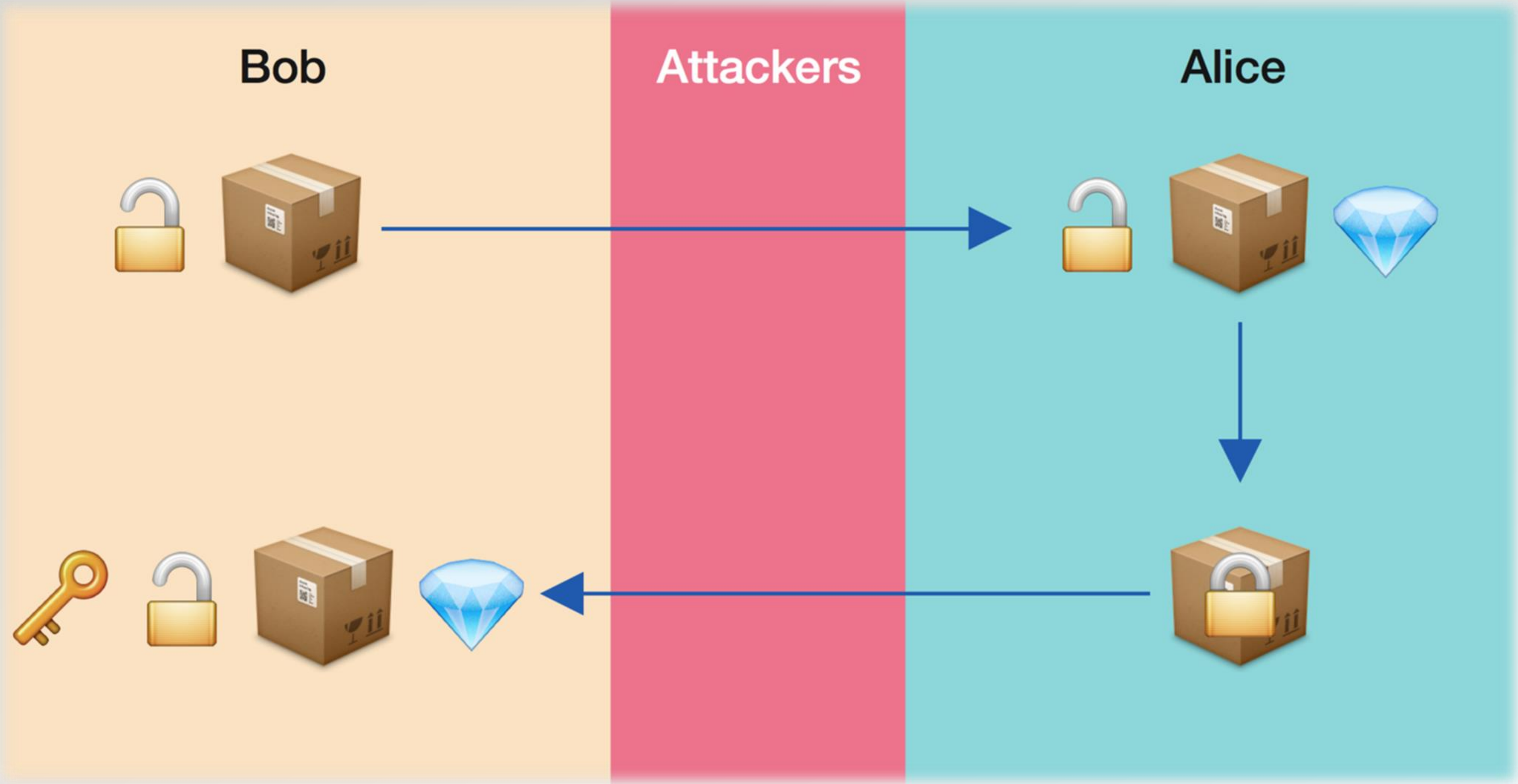
Chinese Remainder Theorem

Chinese Remainder Theorem:

Let p and q be $coprime$

$$x \equiv a(mod\ pq) \Leftrightarrow \begin{cases} x \equiv a(mod\ p) \\ x \equiv a(mod\ q) \end{cases}$$

RSA Algorithm



RSA Algorithm

Key Pair Generation Algorithm

1. Choose prime numbers p and q
2. $n = pq$ and $\varphi(n) = (p-1)(q-1)$
3. Choose e (public exponent)

$$\begin{cases} 1 < e < \varphi(n) \\ \gcd\big(e, \varphi(n)\big) = 1 \end{cases}$$

4. Choose d (private exponent)

$$\begin{cases} 1 < d < \varphi(n) \\ ed \equiv 1 (mod\ \varphi(n)) \end{cases}$$

Secret: $p, q, d, \varphi(n)$

Key Pair

Public Key: (n, e)

Private Key: (n, d)

Cryptographic Algorithm

Encryption: $E(m) = m^e (mod\ n)$

Decryption: $D(m) = m^d (mod\ n)$

m: $1 < m < n$

https://www.di-mgt.com.au/rsa_alg.html

RSA Algorithm (In Practice)

Key Pair Generation Algorithm

1. Select e from $\{3, 5, 17, 257, 65537\}$ → Prime numbers that allow for less expensive computations and optimizations

↓

2. Choose prime numbers p, q each with → Simpler to test if the prime number x respects $\gcd(e, \varphi(n)) = 1$

   $n = pq$ and $\varphi(n) = (p-1)(q-1)$

3. Calculate d using $modular\ inversion$ → Recent standards use the $Charmichael\ function$

   $d = e^{-1}(mod\ \varphi(n))$

   Using Extended Euclidean Algorithm

$$\lambda(n) = lcm(p-1, q-1)$$

or

$$\lambda(n) = \frac{(p-1)(q-1)}{gdc(p-1, q-1)}$$

↓

$$d = e^{-1}(mod\ \lambda(n))$$

Secret: $p, q, d, \varphi(n)$

Example: https://brilliant.org/wiki/rsa-encryption/

RSA Algorithm

Sara	Emanuel
Key Creation	
Choose secret primes p and q. Choose encryption exponent e with $\gcd(e, (p-1)(q-1)) = 1$. Publish $N = pq$ and e.	
Encryption	
	Choose plaintext m. Use Sara's public key (N, e) to compute $c \equiv m^e \pmod{N}$. Send ciphertext c to Sara.
Decryption	
Compute d satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. Compute $m' \equiv c^d \pmod{N}$. Then m' equals the plaintext m.	

RSA Proof

From the RSA Cryptographic algorithm we have

$$\left.\begin{aligned} E(m) &= m^e (mod\, n) \\ D(m) &= m^d (mod\, n) \end{aligned}\right\} D\big(E(m)\big) \equiv (m^e)^d \equiv m^{ed} \equiv m\ (mod\, n)$$

The same applies for $E(D(m))$

Prove $\boxed{m^{ed} \equiv m(mod\, n)}$

RSA Proof (1/4)

Prove: $m^{ed} \equiv m (mod\ n)$

From the Key Pair Generation Algorithm

4. Choose d (private exponent)

   $1 < d < \varphi(n)$

   $ed \equiv 1 (mod\ \varphi(n))$ $\longrightarrow$ d is the inverse of e modulus $\varphi(n)$

$$ed \equiv 1\big(mod\ \varphi(n)\big) \Leftrightarrow ed = k\varphi(n) + 1 \Rightarrow m^{ed} = m^{k\varphi(n)+1}$$

$$m^{ed} \equiv m^{k\varphi(n)+1} \equiv m^{k\varphi(n)} m \equiv \left(m^{\varphi(n)}\right)^k m\ (mod\ n)$$

RSA Proof (2/4)

$$\underbrace{\left(m^{\varphi(n)}\right)^k m \ (mod\ n)}$$

Euler's Theorem: Let a and n be $coprime$

$$a^{\varphi(n)} \equiv 1(mod\ n)$$

Since m can be any value, we are not sure if m and n are coprime

m and n are coprime

m and n are not coprime

RSA Proof – (m, n) coprimes (3/4)

$$\left(m^{\varphi(n)}\right)^k m \equiv 1^k m \equiv m (mod\ n)$$

Euler's Theorem: Let a and n be *coprime*

$$a^{\varphi(n)} \equiv 1 (mod\ n)$$

$$\therefore m^{ed} \equiv m (mod\ n)$$

https://www2.cs.arizona.edu/~collberg/Teaching/466-566/2012/Handouts/Handout-7.pdf
https://www.di-mgt.com.au/rsa_theory.html#modmn

RSA Proof – (m, n) not coprimes (4/4)

(m, n) not coprime $\Rightarrow \begin{cases} p \mid m \Rightarrow (m, q)\ coprimes \\ \text{or} \\ q \mid m \Rightarrow (m, p)\ coprimes \end{cases}$

Proof for (m, q) $coprime$, same for (m, p):

$$\begin{aligned} m^{k\varphi(n)+1} &\equiv m^{k\varphi(p)\varphi(q)} m (mod\ q) \\ &\equiv \left(m^{\varphi(q)}\right)^{k\varphi(p)} m (mod\ q) \\ &\equiv (1)^{k\varphi(p)} m (mod\ q) \\ &\equiv m (mod\ q) \end{aligned}$$

Euler's Theorem

Chinese Remainder Theorem

$(p, q)\ coprimes$

$$x \equiv a(mod\ pq) \Leftrightarrow \begin{cases} x \equiv a(mod\ p) \\ x \equiv a(mod\ q) \end{cases}$$

If we prove both cases, we prove for n

$$\therefore m^{ed} \equiv m(mod\ n)$$

Security, Attacks & Vulnerabilities

RSA Problem: Given c (ciphertext), e (public exponent) and n (modulus).

Find m such that $m^e \equiv c(mod\ n)$

All (mathematical) attacks are equivalent to **factoring $\boldsymbol{n}$**

(With n decomposed we obtain all information)

↓

Factoring $\boldsymbol{n}$ = **Prime Factorization Problem**: Decompose a composite number into a product of its smaller prime numbers.

Solution: Increase key size

Larger Key = Harder Factoring = More Secure

RSA-768 HAS BEEN BROKEN

RSA-2048 AND UP IS RECOMMENDED

Factoring n knowing $\varphi(n)$

By knowing n and $\varphi(n)$, we can obtain p and q $\longrightarrow$ Factoring n is as easy as factoring $\varphi(n)$

$$\begin{aligned}\varphi(n) &= (p-1)(q-1)\\ &= pq - (p+q) + 1\\ &= n - p - \frac{n}{p} + 1\end{aligned}$$

$$\uparrow$$

$$n = pq \Leftrightarrow q = \frac{n}{p}$$

$$\begin{aligned}& p\varphi(n) = p\left(n - p - \frac{n}{p} + 1\right) && \Leftrightarrow\\ \Leftrightarrow\quad & p\varphi(n) = np - p^2 - n + p && \Leftrightarrow\\ \Leftrightarrow\quad & p^2 - np + n - p - p - p\varphi(n) = 0 && \Leftrightarrow\\ \Leftrightarrow\quad & p^2 - p(n - \varphi(n) + 1) + n = 0 &&\end{aligned}$$

$$\boxed{p^2 - p(n - \varphi(n) + 1) + n = 0}$$

Quadratic equation – Two solutions of p

Both solutions are p and q

Factoring n knowing $\varphi(n)$ - Example

$$n = 84773093 \qquad \varphi(n) = 84754668$$

$$\boxed{p^2 - p(n - \varphi(n) + 1) + n = 0}$$

$$\begin{aligned}
& p^2 - p(84773093 - 84754668 + 1) + 84773093 = 0 && \Leftrightarrow \\
\Leftrightarrow \quad & p^2 - 18426p + 84773093 = 0 && \Leftrightarrow \\
\Leftrightarrow \quad & p = 9539 \lor p = 8887
\end{aligned}$$

$$p = 9539 \land q = 8887$$

$$n = pq = 9539 * 8887 = 84773093$$

Security, Attacks & Vulnerabilities

Low Public Exponent

Ex. It is possible to recover the plaintext if the algorithm uses a **small exponent**, sends it to **different recipients** and **does not use padding.**

$$e = 3$$

$$x \equiv c_1 (mod\ n_1)$$

$$x \equiv c_2 (mod\ n_2) \xrightarrow[m^3 < n_1 n_2 n_3]{Chinese\ Remainder\ Theorem} x = m^3 \Leftrightarrow m = x^{\frac{1}{3}}$$

$$x \equiv c_3 (mod\ n_3)$$

With a small message

If $m^e < n \Rightarrow m^e = m^e (mod\ n) \Rightarrow m = \sqrt[e]{m^e}$ $\longrightarrow$ $m^e > n$, to guarantee security

Security, Attacks & Vulnerabilities

Common modulus: Users in a group should not use the same modulus.

1. $user_n$ given his (e_n, d_n) pair can factorize n, and compute the d of all others.

$$d_n = \frac{1}{e_n} (mod\ \varphi(n))$$

2. An attacker can also obtain the original plaintext.

Attacker sees:

$$c_1 = m^{e_1}\ (mod\ n)$$

$$c_2 = m^{e_2}\ (mod\ n)$$

$$\xrightarrow[\text{(knows } e_n\text{)}]{}$$

$$t_1 = e_1^{-1}\ (mod\ e_2)$$

$$t_2 = \frac{(t_1 e_1 - 1))}{e_2}$$

$$\longrightarrow$$

$$c_1^{t_1} c_2^{-t_2} = m(mod\ n)$$

Security, Attacks & Vulnerabilities

Timing Attacks: Analyze the time it takes to encrypt/decrypt and extrapolate information.

Quantum Algorithms

- **Shor's Algorithm:** Given N, finds it's prime factors. (In polynomial time)

Defenses:

- Use random padding to the message
- Do not use low public exponents

Pratical Applications

- Biomedical info

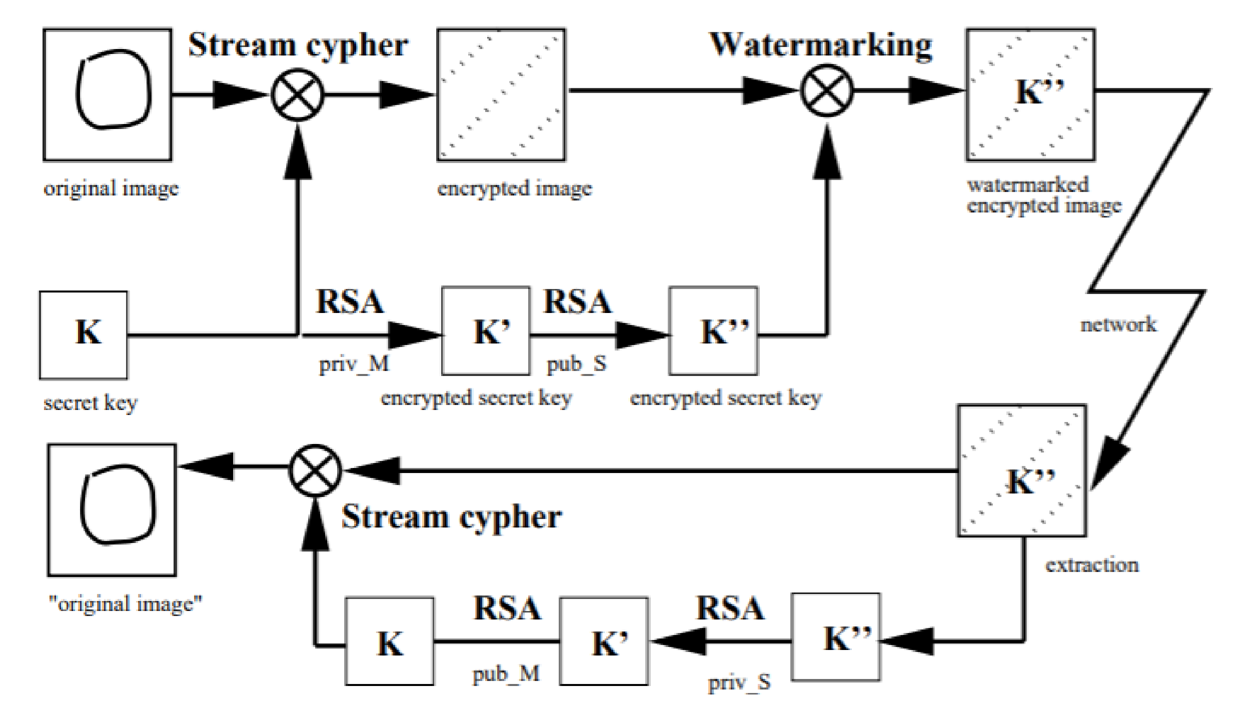


Figure 3: Combination of public key encryption, secret key encryption, and watermarking.

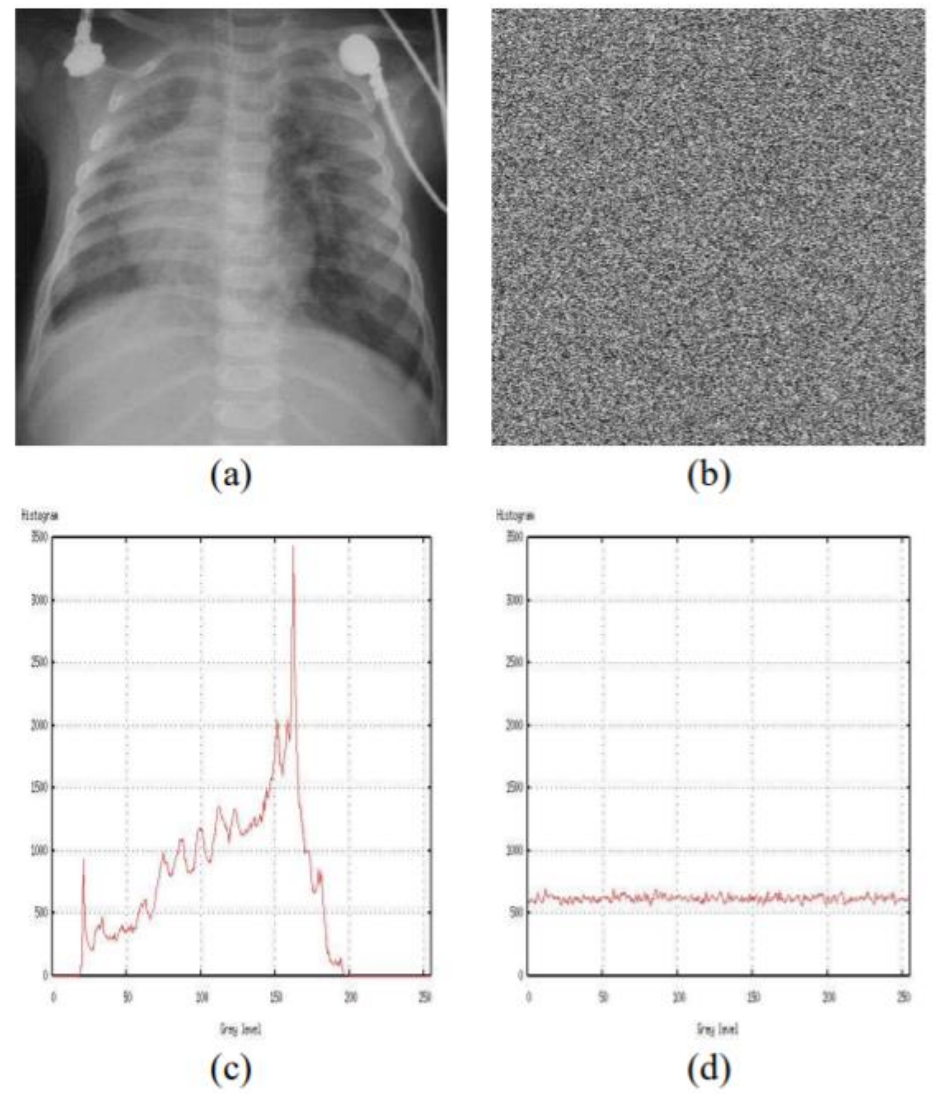


Figure 4: a) Original image, b) Encrypted image with the stream cypher algorithm, with a key of 128 bits, c) Original image histogram, d) Histogram of the image (b).

Sep 2004: A New Crypto-Watermarking Method for Medical Images Safe Transfer

Pratical Applications

- Biomedical info

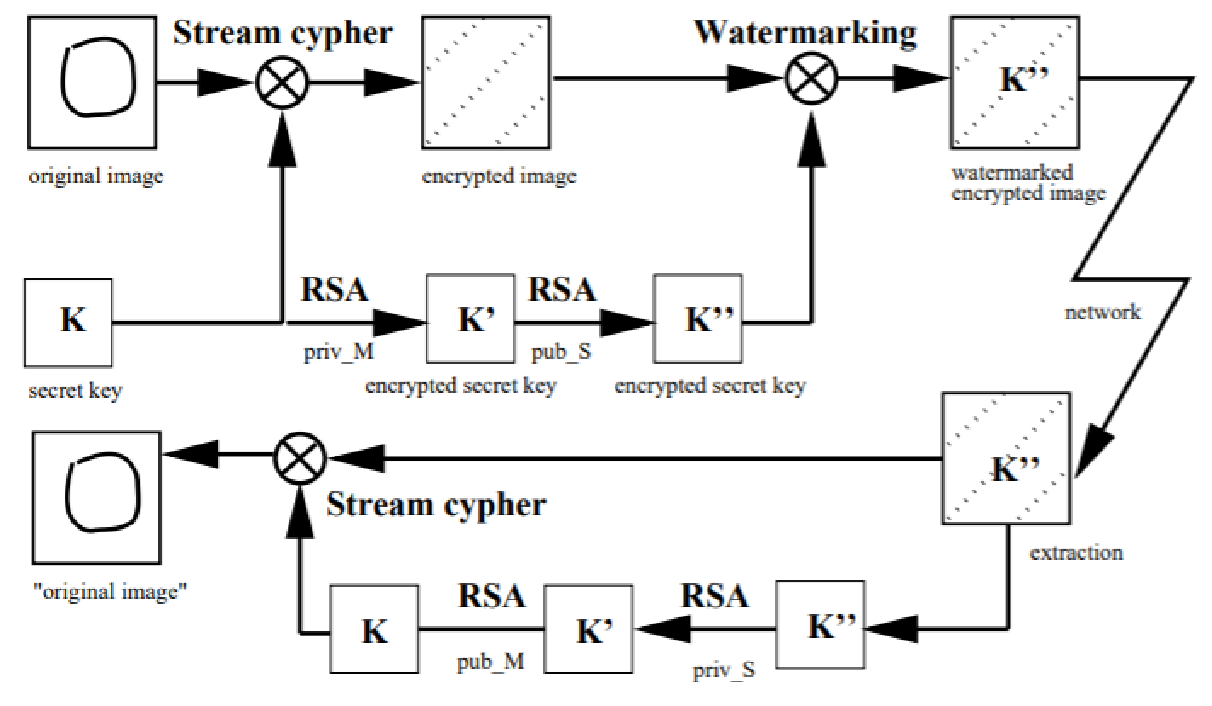


Figure 3: Combination of public key encryption, secret key encryption, and watermarking.

Sep 2004: A New Crypto-Watermarking Method for Medical Images Safe Transfer

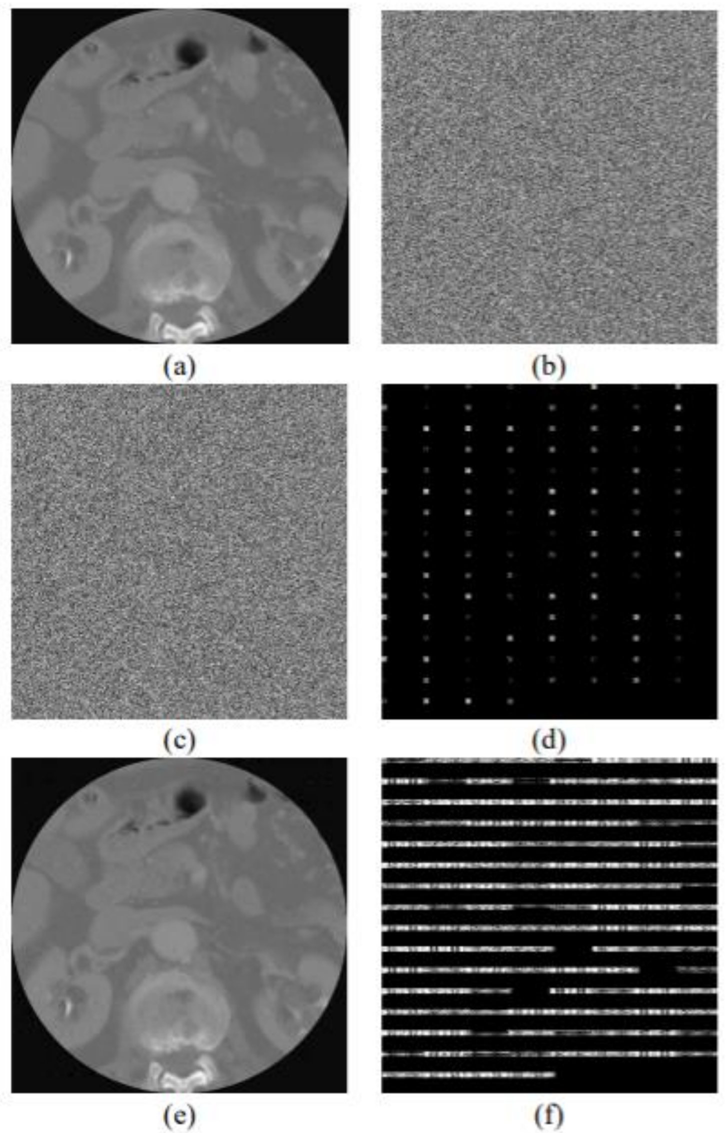


Figure 5: a) Original medical image, b) Encrypted image, c) Watermarked encrypted image with 128-bits key, d) Difference between the encrypted image and the watermarked encrypted image, e) Decryption of the watermarked encrypted image, f) Difference between original image and the decrypted watermarked one.

Pratical Applications

2018: Decrypting the Encryption Debate: A Framework for Decision Makers. (National Academies of Sciences, Engineering, and Medicine. 2018)
https://www.nap.edu/read/25010/

- Biomedical info

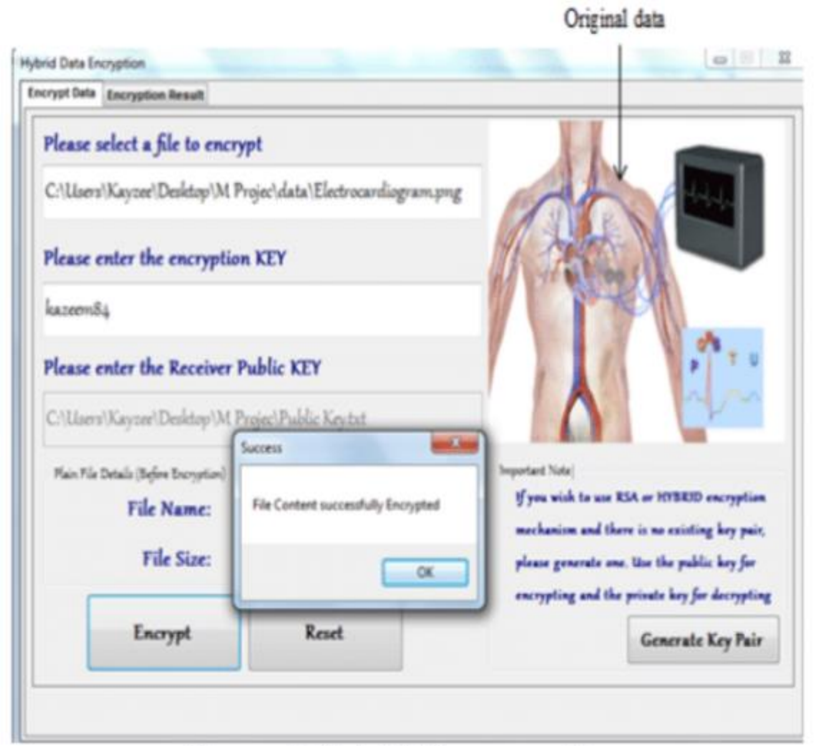


Figure 5: Hybrid files encryption

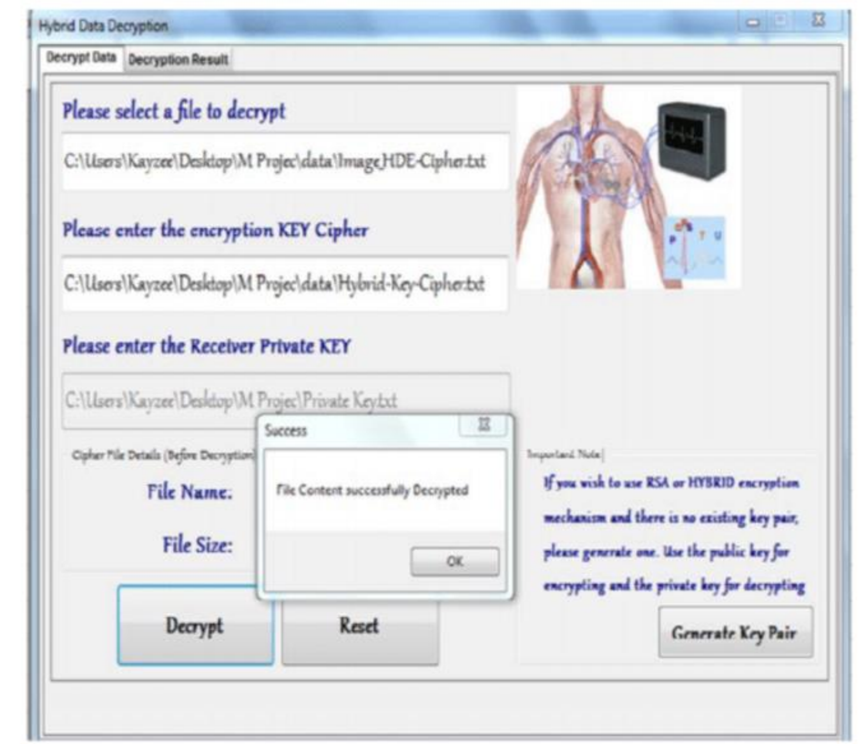
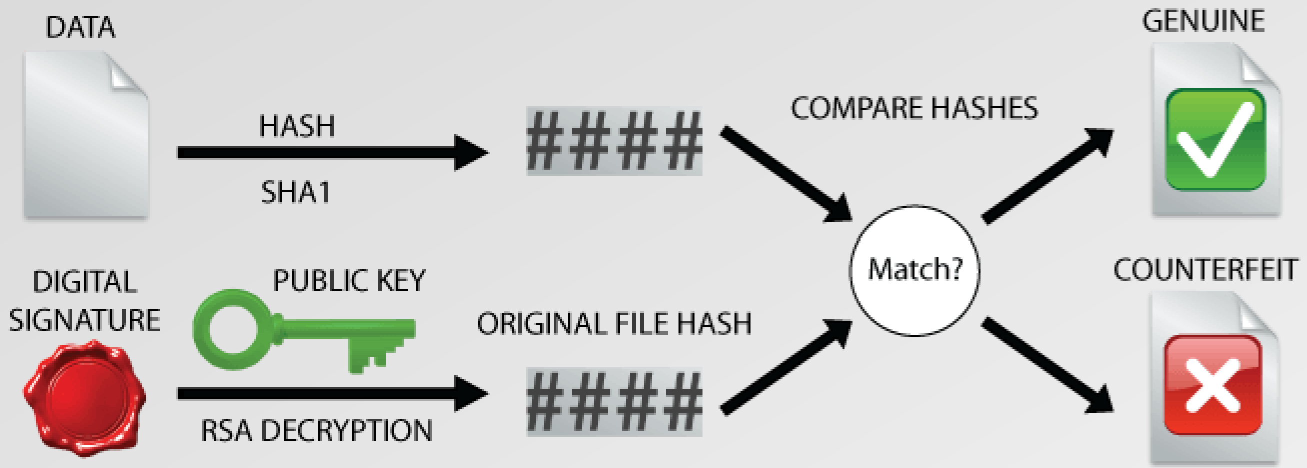


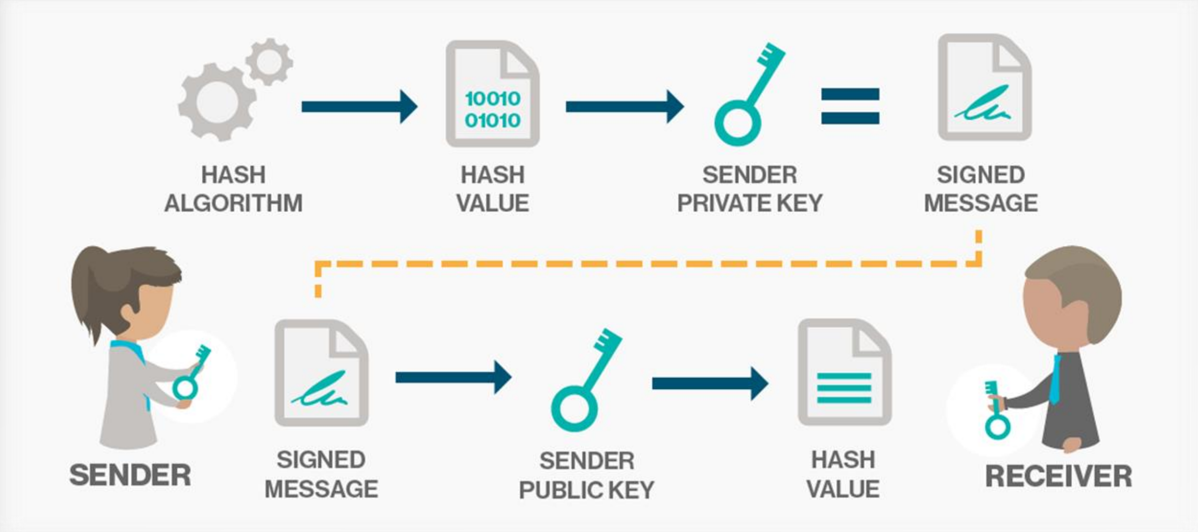
Figure 8: The hybrid data decryption

Nov 2014: Development of a GUI for Hybrid (DES-RSA) Data Encryption and Decryption for Transmission of Biomedical Data

Pratical Applications

- Biomedical info
- Digital signing
- Signature verification





RSA cryptosystem (Rivest-Shamir-Adleman)

Criptografia 2018/2019 | Prof. Isabel Oitavem
FCT NOVA – 10 de Maio de 2019

Emanuel Carvalho 45466, MIEI
Sara Lobo 45622, MIEB

Bibliography

Nov 2014: Development of a GUI for Hybrid (DES-RSA) Data Encryption and Decryption for Transmission of Biomedical Data
https://www.researchgate.net/publication/303498522_Development_of_a_GUI_for_Hybrid_DES-RSA_Data_Encryption_and_Decryption_for_Transmission_of_Biomedical_Data

Dec 2007: Teoria Aritmética dos Números e Criptografia RSA
https://www.ime.unicamp.br/~ftorres/ENSINO/MONOGRAFIAS/antonio_RSA.pdf

May 2007: RSA Theory
https://www.di-mgt.com.au/rsa_theory.pdf

Sep 2004: A New Crypto-Watermarking Method for Medical Images Safe Transfer
https://www.lirmm.fr/~wpuech/recherche/publications/04_eusipco_puech.PDF

Feb 1978: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems (R.L. Rivest, A. Shamir, and L. Adleman)
http://people.csail.mit.edu/rivest/Rsapaper.pdf

Bibliography

2019: Fun with bignum: how RSA encryption works
https://cran.r-project.org/web/packages/openssl/vignettes/bignum.html

2018: Decrypting the Encryption Debate: A Framework for Decision Makers. (National Academies of Sciences, Engineering, and Medicine. 2018)
https://www.nap.edu/read/25010/chapter/4#28

2017: How to Explain Modern Security Concepts to your Children - Hal
https://hal.archives-ouvertes.fr/hal-01397035/document

Bibliography

2016: Common Attacks on RSA and its Variants with Possible Countermeasures
https://www.researchgate.net/publication/316588561_Common_Attacks_on_RSA_and_its_Variants_with_Possible_Countermeasures

2013: Analysis and Research of the RSA Algorithm
https://scialert.net/fulltext/?doi=itj.2013.1818.1824

2006: Introduction to RSA and to Authentication
https://www.nku.edu/~christensen/section%2026%20RSA.pdf

Bibliography

The RSA Algorithm: A Mathematical History of the Ubiquitous Cryptological Algorithm
https://www.sccs.swarthmore.edu/users/10/mkelly1/rsa.pdf

Twenty Years of Attacks on the RSA Cryptosystem
https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf

RSA Encryption - Australian Mathematical Sciences Institute
http://www.amsi.org.au/teacher_modules/pdfs/Maths_delivers/Encryption5.pdf

RSA Encryption – Keeping the Internet Secure | AMS Grad Blog
https://blogs.ams.org/mathgradblog/2014/03/30/rsa/

Public Key Cryptography: RSA Encryption Algorithm
https://www.youtube.com/watch?v=wXB-V_Keiu8

RSA-129 - Numberphile (featuring Ron Rivest, co-inventor of RSA)
https://www.youtube.com/watch?v=YQw124CtvO0

Bibliography

The Mathematics of the RSA Public-Key Cryptosystem
http://www.mathaware.org/mam/06/Kaliski.pdf

Prime Number Generator
https://www.browserling.com/tools/prime-numbers

RSA Calculator
https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html

Prime Factorization Calculator
https://www.calculatorsoup.com/calculators/math/prime-factors.php

MIT PGP Public Key Server
https://pgp.mit.edu/

Convert Characters to ASCII Codes
https://www.browserling.com/tools/text-to-ascii

Bibliography

RSA Encryption
https://brilliant.org/wiki/rsa-encryption/

How RSA & PKI works and the math behind it.
https://www.youtube.com/watch?v=Jt5EDBOcZ44

What is the relation between RSA & Fermat's little theorem?
https://crypto.stackexchange.com/a/398

Dr Clifford Cocks
http://www.bristol.ac.uk/graduation/honorary-degrees/hondeg08/cocks.html
https://www.wired.com/1999/04/crypto/

Public-key cryptography, RSA, Attacks against RSA – Système et Sécurité
https://www.lri.fr/~fmartignon/documenti/systemesecurite/6-PublicKey.pdf

Bibliography

Why Does RSA Work (Udacity video)
https://www.youtube.com/watch?v=kKgp0KdpOhQ

RSA Encryption and Decryption (Live Demo)
http://demonstrations.wolfram.com/RSAEncryptionAndDecryption/

Cryptology
https://cs.lmu.edu/~ray/notes/cryptology/

RSA Algorithm
https://www.di-mgt.com.au/rsa_alg.html

RSA Theory
https://www.di-mgt.com.au/rsa_theory.html