CYBER CHALLENGE
CyberChallenge.IT

cini
Cybersecurity National Lab

SPONSOR PLATINUM

accenturesecurity

aizoon AUSTRALIA EUROPE USA
TECHNOLOGY CONSULTING

B5

EY Building a better working world

eni

exprivia | ITALTEL

IBM

KPMG

LEONARDO

NTT DATA
Trusted Global Innovator

NUMERA
SISTEMI E INFORMATICA S.p.A.

Telsy

SPONSOR GOLD

bip.

cisco

MONTE DEI PASCHI DI SIENA
BANCA DAL 1472

negg

NOVANEXT
connecting the future

pwc

SPONSOR SILVER

DiGi ONE
the leading digital company

ICT CYBER CONSULTING

Asymmetric Encryption & Key Exchange

Rocco DE NICOLA
IMT Lucca

CYBER CHALLENGE
CyberChallenge.IT

cini
Cybersecurity
National Lab

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

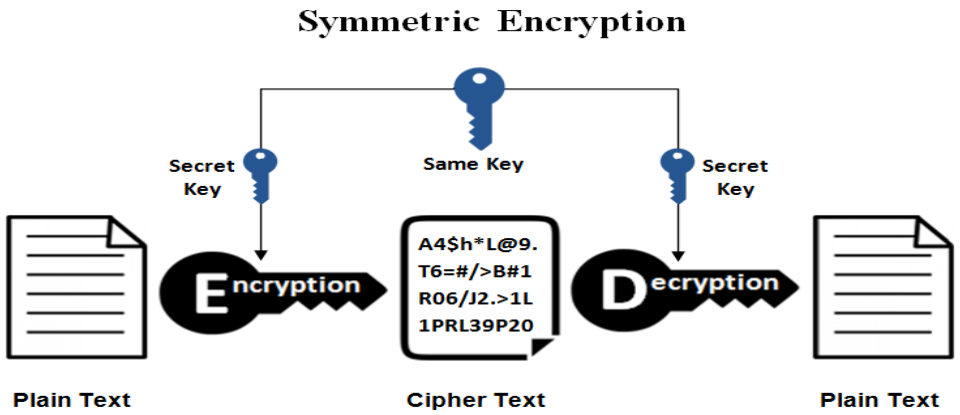
- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- 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.

Outline

- Introduction
- Double Key Cryptography
- Two breakthroughs
  - Key Sharing: Diffie-Hellman
  - Double keys: RSA
- Examples
- Attacks to RSA

Symmetric key cryptography

Symmetric key encryption refers to methods in which both sender and recipient share the same key.



- Also called *single key* or *conventional encryption*
- The only type of encryption in use before the end of the 1970s
- The most common of the two types of encryption

Symmetric key cryptography

- Plaintext: the original message given as input to the algorithm
- Secret key: Another input to the encryption algorithm
- Encryption Algorithm: performs various substitutions and clear transformations using the secret key to obtain an encrypted text
- Ciphertext: the encrypted message produced as output; depends on the plain text and secret key
- Decryption algorithm: it is the symmetric of the encryption algorithm; from the cipher text it produces the original clear plaintext.

Symmetric key cryptography

(Traditional) private/secret/single key encryption schemes:

- require that sender and receiver have obtained a copy of the secret key securely, and they keep it safe
- do not protect a sender from a recipient who falsifies a message and then claims that the request was sent by the sender.

Outline

- Introduction
- Double Key Cryptography
- Two breakthroughs
  - Key Sharing: Diffie-Hellman
  - Double keys: RSA
- Examples
- Attacks to RSA

Double Key Encryption

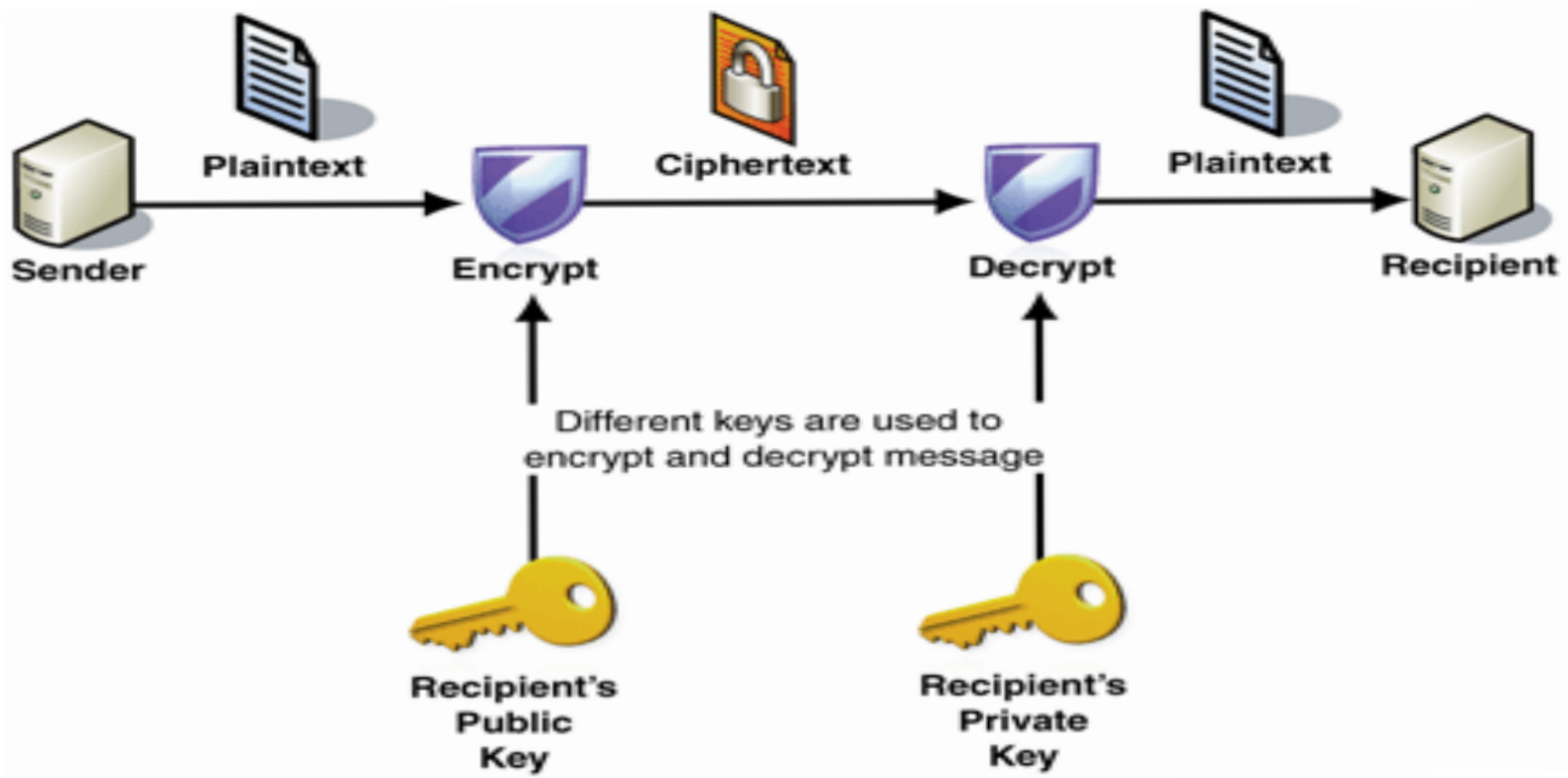
New encryption schemes

- 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 {public key , 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 private key encryption.
- Is based on properties guaranteed by number theory rather than on the use of permutations and substitutions.

Asymmetric encryption



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.

Double Key Encryption

- Plaintext: the original message given as input to the algorithm.
- Keys: a pair of public/private keys generated so that one is used for encryption and the other for decryption.
- Encryption algorithm: performs substitutions and transformations on the plain text using one of the two keys to encrypt.
- Ciphertext: the encrypted message produced as output, which depends on the plain text and on one of the two keys.
- Decryption algorithm: accepts the cipher text and the appropriate key and produces the original plain text.

Keys management

- Each user generates a pair of keys to be used for encryption and decryption of messages:
  - The public key is entered into a public registry or other accessible file.
  - The "twin" key is kept private.
- Each user keeps a collection of the public keys of those he wants to be in contact with
- If Bob encrypts a message using Alice's public key, only Alice will be able to decrypt it using her private key.
- If Bob encrypts a message using his private key, Alice and anyone who knows Bob's public key can decrypt the message

Impact of double key Cryptography

- Published by Diffie and Hellman in 1976
- Based on properties of (Prime) Numbers
- Uses two keys, different but related: public and e private
- Has had profound consequences on: confidentiality, key distribution, authentication

Main proposals for PKC

- D-H (Diffie-Hellman 1976): First public key algorithm to be made public. Allows two users to securely share a secret to be used as a key for subsequent symmetric encryption of messages
- RSA (Rivest, Shamir, Adleman) Developed in 1977: The most widely accepted and implemented approach to public key cryptography
- ECC (Koblitz and Miller 1985) - Elliptic curve cryptography proposed as an alternative to RSA, with the same level of security but with much smaller keys
- DSS (U.S.NIST 1991) Digital Signature Standard revised often until 2013. Provides digital signature function only, cannot be used for encryption or key exchange.

Main proposals for PKC

Algorithm	Digital Signature	Key Exchange	Encryption / Decryption
Rivest, Shamir, Adleman (RSA)	Yes	Yes	Yes
Diffie-Hellman (D-H)	No	Yes	No
Digital Signature Standard (DSS)	Yes	No	No
Elliptic Curve Cryptography (ECC)	Yes	Yes	Yes

Outline

- Introduction
- Double Key Cryptography
- Two breakthroughs
  - Key Sharing: Diffie-Hellman
  - Double keys: RSA
- Examples
- Attacks to RSA

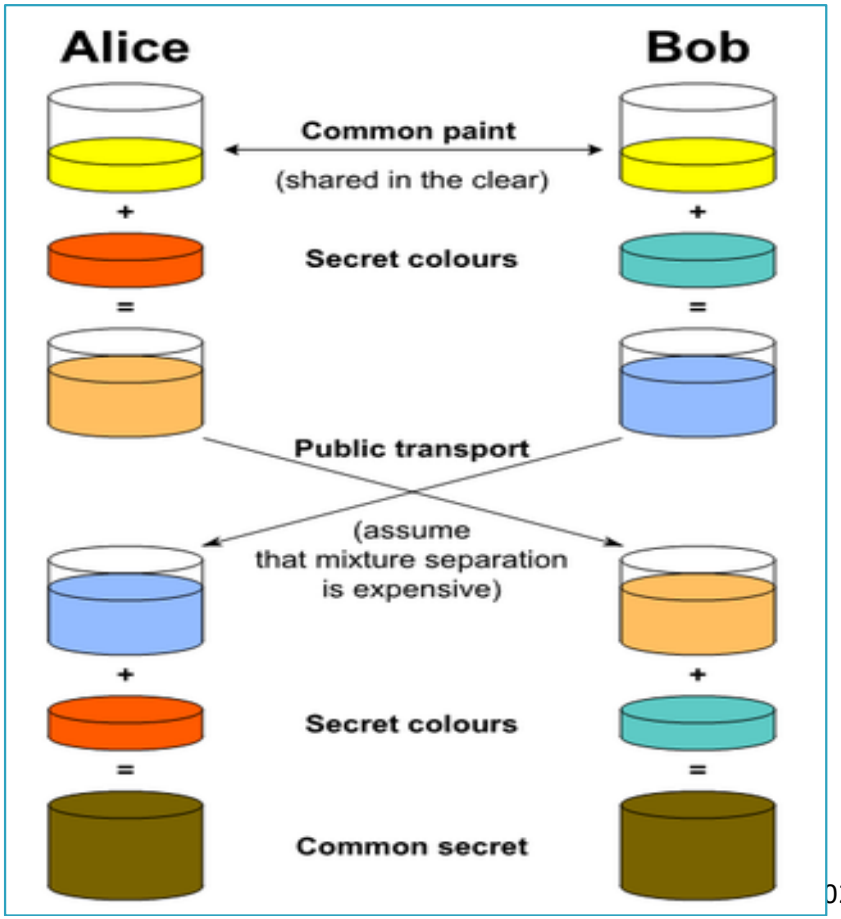
Two Breaktroughs

- Key exchange algorithm (Diffie-Hellman-Merkle 1976):
  - The algorithm was designed to enable users to reach a secure agreement on a shared secret to be used as the key for subsequent symmetric encryptions.
  - The now expired U.S. Patent 4,200,770 from 1977 describes the now public-domain algorithm.
- Public-key scheme (RSA - Rivest, Shamir, Adleman 1977)
  - The most widely accepted and implemented approach to public key cryptography
  - the encryption key is public and distinct from the decryption key which is kept secret (private).

Diffie-Hellman

- Diffie-Hellman (DH) key exchange is a method for secure exchange of cryptographic keys on a public channel and was the first public key protocol to be made public.
- In 2002, Hellman suggested the algorithm be called Diffie–Hellman–Merkle key exchange in recognition of Ralph Merkle's contribution to the invention of public key cryptography.
- Traditionally, secure encrypted communication between two parties required keys to be exchanged first through some physical channel (e.g., a trusted courier).
- The Diffie-Hellman key exchange method allows two parties, who have no knowledge of each other, to establish a shared secret key on an unsecured channel.

Diffie-Hellman - Intuition



How to obtain a shared colour starting from a commonly agreed paint while keeping another one secret.

Diffie-Hellman

- A and B share two values, a prime q and a number α, to be used in subsequent computations.
- A generates a one-time private key X_A, calculates Y_A, and sends it to B.
- B generates a private value X_B, calculates Y_B, and sends it to A.
- Both users can now calculate the same key.
- N.B: A could decide the values of q and α and include them in the first message to B.)

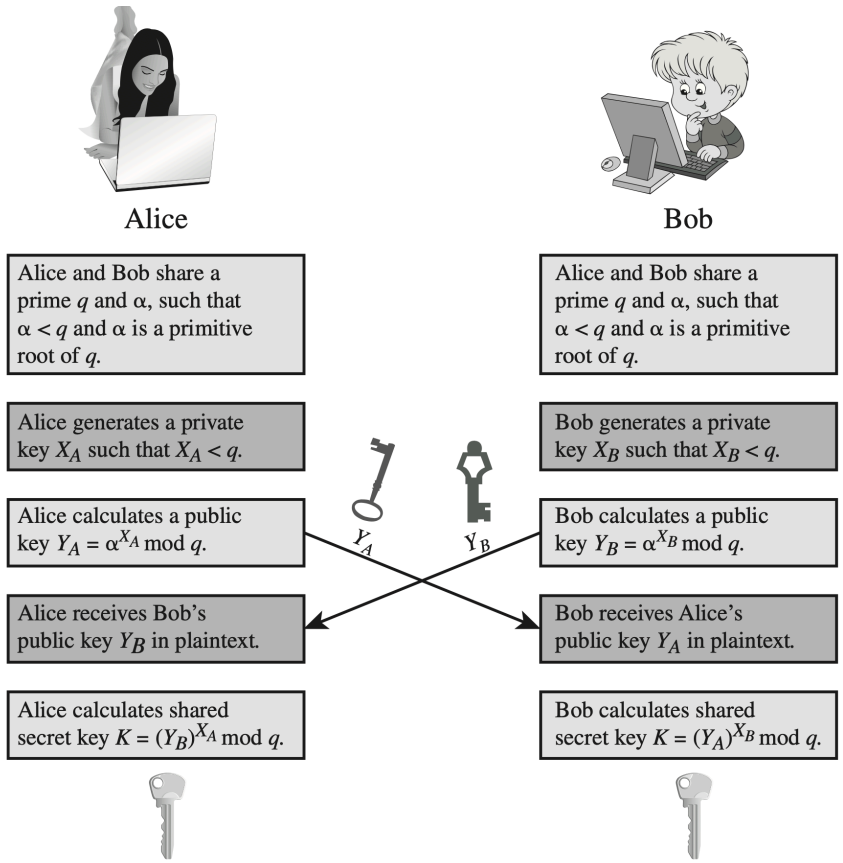


Figure 21.10 **Diffie-Hellman Key Exchange**

Diffie-Hellman: mathematically

- Alice generates:
  - a very high prime number q (1024 bits, about 300 decimal places)
  - a primitive root α of q smaller than q
  - a random secret number a
- Alice calculates $\alpha^a \bmod q$ and sends it to Bob (possibly along with α and q).
- Bob chooses a random secret number b, and calculates $\alpha^b \bmod q$ and sends it to Alice

Why Diffie-Hellman works

- Alice computes $\alpha^a \bmod q$ and sends it to Bob along with α and q.
- Bob computes $\alpha^b \bmod q$ and sends it to Alice
- With subsequent calculations the two get the same key!
  - K_A
    $= (\alpha^b \bmod q)^a \bmod q$
    $= \alpha^{ba} \bmod q$
    $= \alpha^{ab} \bmod q$
    $= (\alpha^a \bmod q)^b \bmod q$
    $= K_B$

Diffie-Hellman: an example

- Alice chooses the primes q = 23 and α = 5.
- Alice chooses the random value a = 6 (kept secret), calculates A = 5^6 mod 23 (= 8) and sends 8 to Bob together with 23 and 5.
- Bob chooses a random value b = 15 (kept secret), calculates B = 5^{15} mod 23 (= 19) and sends 19 to Alice
- With subsequent calculations the two get the same key $K_A = K_B$
  - Alice calculates $K_A = 19^6$ mod 23 = 2
  - Bob calculates $K_B = 8^{15}$ mod 23 = 2
- The secrets are 6 (a), 15 (b) and above all, 2 (α^{ab} e α^{ba}) the future key
- Eve without a and b, with just A and B, cannot do anything.

RSA

- Based on exponentiation of integers modulo (a prime number) n. Very large integers are used (typically 1024 bits)
- Encryption and decryption are single exponentiation mod n: Exponentiation is easy (requires $O((\log n)^3)$ operations).
- Security is guaranteed by the cost of factoring large numbers: factoring is difficult (requires $O(e^{\log n \log \log n})$ operations).

RSA

- RSA is a block cipher, where each message block (ciphertext and plaintext) is an integer between 0 and n
- Message blocks are strings of 1024 bits. Each single block is a decimal number with 309 digits ($2^{1024} \cong 10^{309}$)
- Sender and recipient must know the value of n, and their public keys
- The crucial steps is the choice of the modulus and of the exponents.

RSA encryption and decryption

- Public Key - PU = {e, n} - Private Key - PR = {d, n}
- To encrypt message M, the sender :
  - Gets the recipient's public key PU = {e,n}
  - Computes $C = M^e \bmod n$, with $0 \leq M < n$
- To decipher cyphertext C, the recipient:
  - Uses his private key PR = {d,n}
  - Computes : $M = C^d \bmod n$
- The "magic" is due to the fact that $(M^e \bmod n)^d \bmod n = M$

RSA Key Generation

A user generates a pair of public/private keys as follows:

- Randomly chooses two prime numbers: p, q
- Computes n = p X q and $\phi(n) = (p-1) X (q-1)$ (ϕ: 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
  - $(e X d) \bmod \phi(n) = 1$ with $0 \leq d \leq n$
- Publishes public key (PU={e,n}) and keeps private key (PR={d,n}) secret.

Why RSA Works

- Euler's theorem:
  - $a^{\phi(n)} \bmod n = 1$ 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 $\phi(n)$
  - $e \times d = 1 + (k \times \phi(n))$ for some k

Why RSA Works

- Euler's theorem:
  - $a^{\phi(n)} \bmod n = 1$ if $\gcd(a, n)=1$
- Thus: (working mod n)

$C^d = M^{(eXd)}$ since $C = M^e$

$= M^{(1+(k \text{ X } \phi(n)))}$ since (e, d) are inverses mod $\phi(n)$

$= M^1 \text{ X } (M^{\phi(n)})^k$ with simple arithmetic

$= M^1 \text{ X } (1)^k$ because of Euler's theorem

$= M^1 = M$

Outline

- Introduction
- Double Key Cryptography
- Two breakthroughs
  - Key Sharing: Diffie-Hellman
  - Double keys: RSA
- Examples
- Attacks to RSA

An example in RSA - Key Setup

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

An example in RSA - En/Decryption

- Public key = {7, 187}
- Private key = {23, 187}
- Given M = 88 (88 < 187)
- Cypher M:
  - $C = 88^7 \bmod 187 = 11$
- Decypher C:
  - $M = 11^{23} \bmod 187 = 88$

Using RSA

Encryption uses the exponentiation to the power of the public key e

- if e is small, exponentiation is fast, often e up to 65537 (2^{16}-1) is chosen but if e and is too small the system can be attacked
- To fix e one must be sure that GCD(e,ϕ(n)) = 1, while rejecting all those p and q that are not relatively prime with e

Using RSA

- Decryption uses the exponentiation to the power of the private key d
  - d must be very large, if it's not, the system is insecure.
  - One can use the Chinese remainder theorem to perform computations mod p and mod q separately.
  - Only the owner of the private key who knows the values of p and q can use this technique
- Chinese remainder theorem
  - If one knows the remainders of the Euclidean division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise relatively prime.

Outline

- Introduction
- Double Key Cryptography
- Two breakthroughs
  - Key Sharing: Diffie-Hellman
  - Double keys: RSA
- Examples
- Attacks to RSA

Issues of asymmetric key cryptography

- Brute force attacks are theoretically possible.
- Very large keys are needed: a 64-bit private key scheme has a security more or less similar to that of a 512-bit RSA (the most used Public Key Cryptography).
- In general, the problem is well known, but is made difficult enough to make it unworkable by resorting to very large numbers.
- Encryption and decryption are much slower than for single shared key schemes.

Attacks to RSA

- A key pair is generated by taking two secret random primes and calculating the keys from them.
- If this is done incorrectly it may be possible to reconstruct the primes and calculate the private key.
- There are many attacks against key pairs
- Caveat: An RSA implementation may have vulnerabilities of its own even if the used keys are secure.
- https://www.sjoerdlangkemper.nl/2019/06/19/attacking-rsa/
- https://github.com/Ganapati/RsaCtfTool

Attacks to RSA

- Brute force attacks: The first step in cracking the private key is to find the two prime numbers p and q that were multiplied together to produce the modulus n.
- Timing attacks: RSA algorithm takes different amounts of time to perform its crypto operations according to the key's value. Based on the time required to apply the private key estimates can be made of the private key.
- Cryptographic attacks: The mathematical nature of RSA makes it vulnerable to attacks against message confidentiality and attacks against public key generation techniques.

Cryptographic attacks to RSA

- Modulus too small: If the RSA key is too short, the modulus can be factored by just using brute force.
- Low private exponent: Decrypting a message consists of calculating c^d (mod n). The smaller d is, the faster this operation goes. If the private exponent is small, the public exponent is necessarily large. A public key with a large public exponent, is a good hint for attackers.
- Low public exponent: Encrypting is performed by calculating m^e (mod n). Having a low public exponent makes the system vulnerable to certain attacks if used incorrectly.
- Generator p and q close together: When creating the key, two random primes p and q are multiplied. If $p \approx q$. Then $n \approx p^2$ and N can be efficiently factored using Fermat's factorization method.

The factorization Problem

- A key step in key generation is determining the two primes *p* and *q* that are used to generate the number n (p X q) to be used for the modular arithmetic.
- The factorization problem consists in finding p and q knowing only n.
- Currently the largest number that has been factored is 768 bits.
- Finding efficient ways for factorization would make RSA insecure.

Progress in factorisation

Number of Decimal Digits	Approximate Number of Bits	Date Achieved	MIPS-years	Algorithm
100	332	April 1991	7	quadratic sieve
110	365	April 1992	75	quadratic sieve
120	398	June 1993	830	quadratic sieve
129	428	April 1994	5000	quadratic sieve
130	431	April 1996	1000	generalized number field sieve
140	465	February 1999	2000	generalized number field sieve
155	512	August 1999	8000	generalized number field sieve
160	530	April 2003	—	Lattice sieve
174	576	December 2003	—	Lattice sieve
200	663	May 2005	—	Lattice sieve

MIPS-year is the number of instructions executed after a year of calculation by executing one million instructions per second.

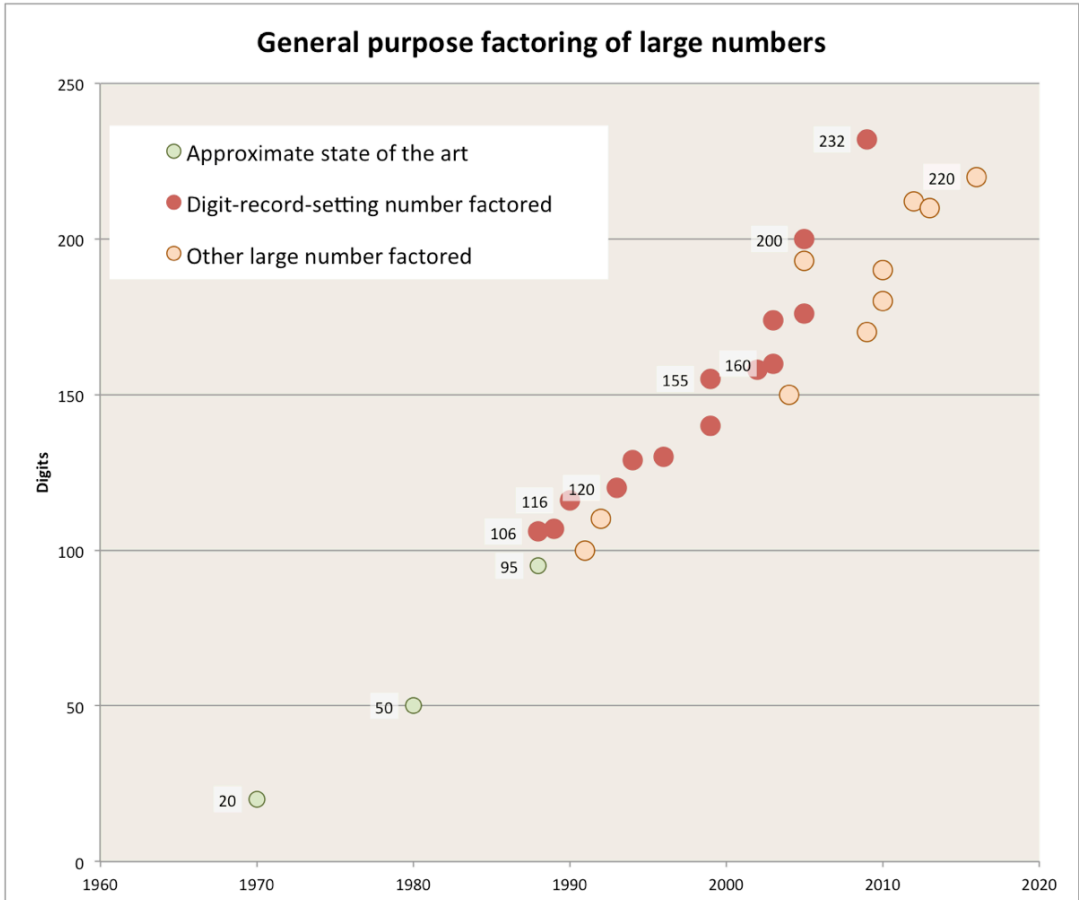
One MIPS-year corresponds to approximately 31.5 trillion instructions.

Progress in factorisation

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-100	100	330	US$1,000[4]	April 1, 1991[5]	Arjen K. Lenstra
RSA-110	110	364	US$4,429[4]	April 14, 1992[5]	Arjen K. Lenstra and M.S. Manasse
RSA-120	120	397	US$5,898[4]	July 9, 1993[6]	T. Denny *et al.*
RSA-129 [**]	129	426	US$100	April 26, 1994[5]	Arjen K. Lenstra *et al.*
RSA-130	130	430	US$14,527[4]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA-140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA-150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA-155	155	512	US$9,383[4]	August 22, 1999	Herman te Riele *et al.*
RSA-160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA-170 [*]	170	563		December 29, 2009	D. Bonenberger and M. Krone [***]
RSA-576	174	576	US$10,000	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA-180 [*]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[7]
RSA-190 [*]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA-640	193	640	US$20,000	November 2, 2005	Jens Franke *et al.*, University of Bonn
RSA-200 [*] ?	200	663		May 9, 2005	Jens Franke *et al.*, University of Bonn
RSA-210 [*]	210	696		September 26, 2013[8]	Ryan Propper
RSA-704 [*]	212	704	US$30,000	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220 [*]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA-230 [*]	230	762		August 15, 2018	Samuel S. Gross, Noblis, Inc.
RSA-232 [*]	232	768		February 17, 2020[9]	N. L. Zamarashkin, D. A. Zheltkov and S. A. Matveev.
RSA-768 [*]	232	768	US$50,000	December 12, 2009	Thorsten Kleinjung *et al.*
RSA-240 [*]	240	795		Dec 2, 2019[10]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA-250 [*]	250	829		Feb 28, 2020[11]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann

RSA Factoring Challenge.
From Wikipedia

Progress in factorisation



Numbers of digit factorised up to now.

- The green dots represent state of the art
- The yellow dots indicate the largest factorised numbers
- The red dots indicate when they were factorized the first time.

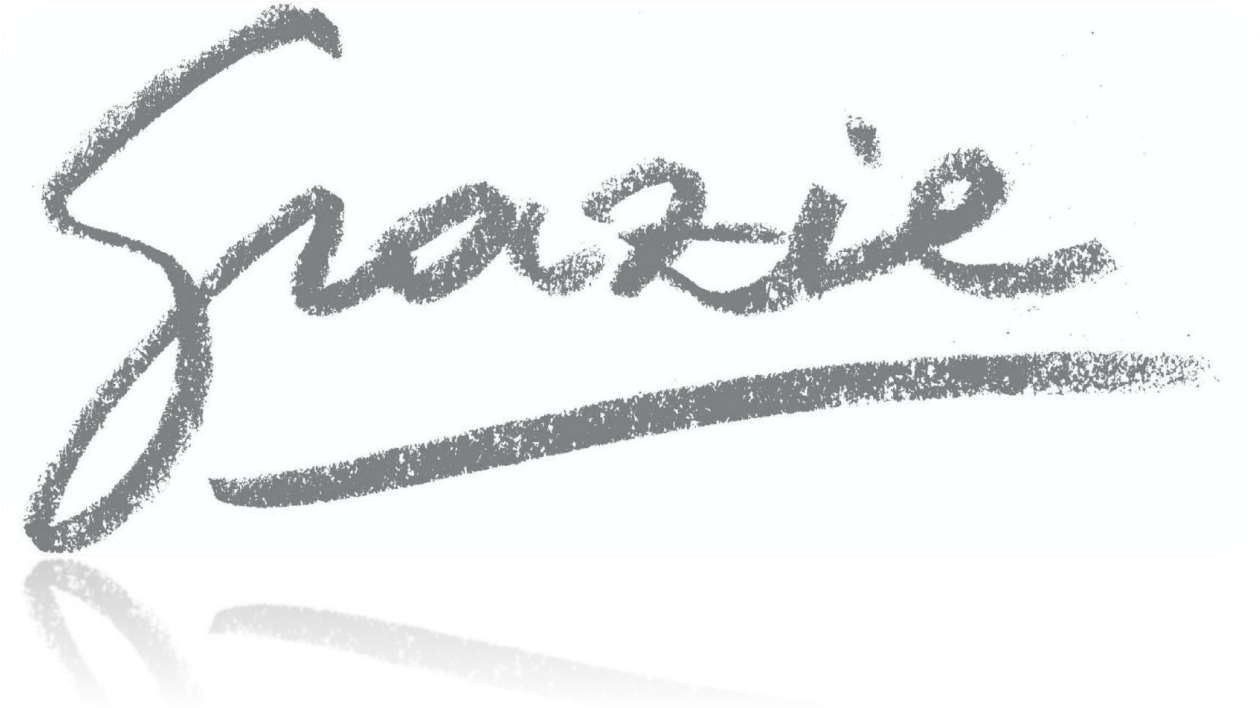
Brute force attacks to RSA

With short keys, the modulus can be factored by brute force.

- A 256-bit modulus can be factored in a couple of minutes.
- A 512-bit modulus takes several weeks on modern consumer hardware.
- Factoring 1024-bit keys is definitely not possible in a reasonable time with reasonable means, but may be possible for well equipped attackers.
- 2048-bit is secure against brute force factoring.
- Quantum Computers could change the scene: "*How to factor 2048 bit RSA Integers in 8 Hours using 20 million noisy qubits*" (arxiv.org/abs/1905.09749)

Misconceptions

- Public key cryptography is more secure by cryptanalysis than symmetric encryption
  - There is nothing in principle that makes one superior to the other in terms of resistance to cryptanalysis
- Public key cryptography has made symmetric encryption obsolete
  - The computational overhead of public key cryptography suggests that symmetric encryption will not be abandoned
- Key distribution is trivial for public key cryptography, while the use of key distribution centers for symmetric encryption is heavy.
  - Even for public key encryption, protocols that often involve a central agent are necessary, and the procedures are no longer simple.



CYBER CHALLENGE
CyberChallenge.IT

 Rel. 05.04.2020

cini
Cybersecurity National Lab

CYBER CHALLENGE
CyberChallenge.IT

cini
Cybersecurity National Lab

SPONSOR PLATINUM

accenturesecurity

aizoon AUSTRALIA EUROPE USA
TECHNOLOGY CONSULTING

B5

EY Building a better working world

eni

exprivia | ITALTEL

IBM

KPMG

LEONARDO

NTT DATA
Trusted Global Innovator

NUMERA
SISTEMI E INFORMATICA S.p.A.

Telsy

SPONSOR GOLD

bip.

cisco

MONTE DEI PASCHI DI SIENA
BANCA DAL 1472

negg

NOVANEXT
connecting the future

pwc

SPONSOR SILVER

DiGi ONE
the leading digital company

ICT CYBER CONSULTING