Cryptocomplexity II

Kryptokomplexität II

Sommersemester 2024

Chapter 1: Reminder: Tasks and Aims of Cryptology and RSA

Dozent: Prof. Dr. J. Rothe

Wpsmftvohtxfctjuf

Vorlesungswebsite

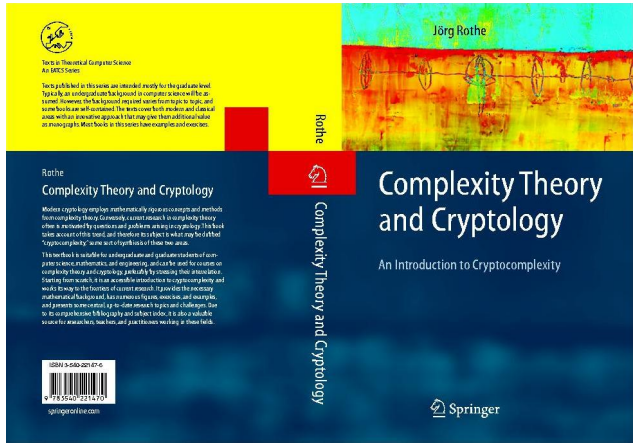
- Some information and material for this module can be found in **ILIAS**.
- In addition, slides, exercises, and other material can also be downloaded from:

  `https://ccc.cs.uni-duesseldorf.de/~rothe/cryptocomp2`

Literature

Jörg Rothe: "Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität", eXamen.Press, Springer-Verlag, 2008

Jörg Rothe: "Complexity Theory and Cryptology. An Introduction to Cryptocomplexity", EATCS Texts in Theoretical Computer Science, Springer-Verlag, 2005



Literature

- **Douglas R. Stinson: "Cryptography: Theory and Practice"**, Chapman & Hall/CRC, 2. Auflage, 2002
- **Johannes Buchmann: "Einführung in die Kryptographie"**, Springer-Verlag, 2. Auflage, 2001
- **Arto Salomaa: "Public-Key Cryptography"**, Springer-Verlag, 1990
- **Oded Goldreich: "Foundations of Cryptography"**, Cambridge University Press, 2001
- **Bruce Schneier: "Applied Cryptography"**, John Wiley & Sons, 1996

What is Cryptology?

Cryptology

is the art &

science of

Cryptography

Cryptanalysis

What is Cryptology?

Cryptology

is the art &

science of

Cryptography	**Cryptanalysis**
encrypting texts and messages such that unauthorized decryption is prevented	

What is Cryptology?

Cryptology

is the art &

science of

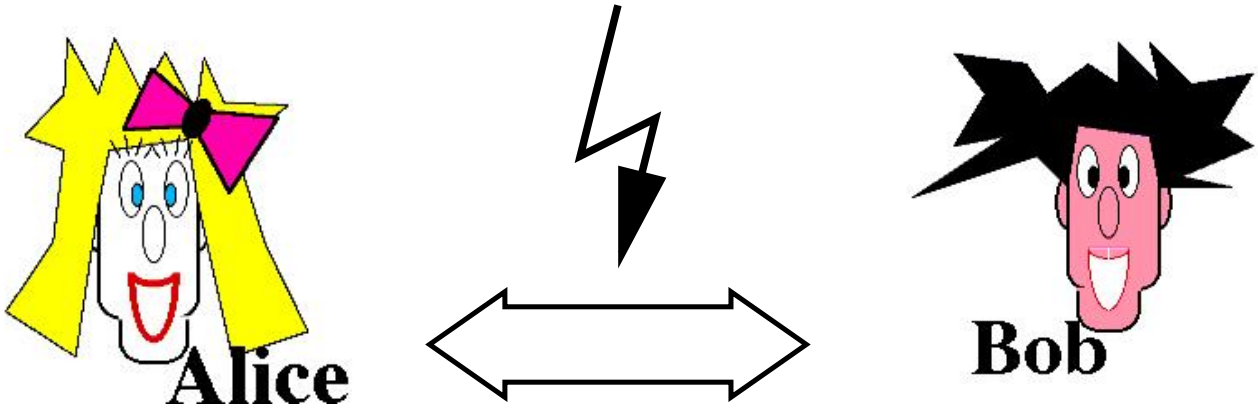
Cryptography	**Cryptanalysis**
encrypting texts and	breaking existing cryptosystems
messages such that	by determining the encryption
unauthorized decryption	keys and deciphering encrypted
is prevented	messages without authorization

Related Fields . . .

- . . . we will *not* consider:
  - Steganography
  - Coding Theory
- . . . whose notions, results, and methods will be used:
  - Complexity Theory
  - Number Theory and (Linear) Algebra
  - Probability Theory
  - Algorithmics

A Typical Cryptographic Scenario

Erich

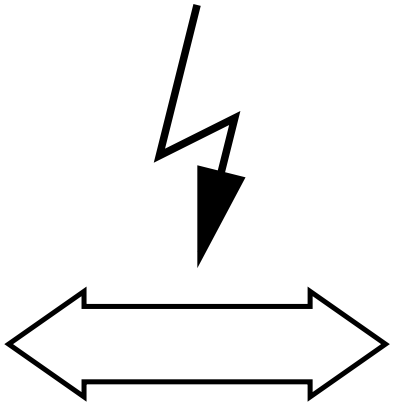


© The design of Alice and Bob is due to Crépeau.

Why Alice and Bob?

A Typical Cryptographic Scenario

Jennifer



Angelina **Brad**

Cryptosystem

Definition

A *cryptosystem* is a quintuple $S = (M, C, K, \mathscr{E}, \mathscr{D})$ such that:

1. M, C, and K are sets, where
   - M is the *message space* (or *"plaintext space"* or *"cleartext space"*),
   - C is the *ciphertext space*, and
   - K is the *key space*.
2. $\mathscr{E} = \{E_k \mid k \in K\}$ is a family of functions $E_k : M \to C$ that are used for encryption, and
3. $\mathscr{D} = \{D_k \mid k \in K\}$ is a family of functions $D_k : C \to M$ that are used for decryption.
4. For each key $e \in K$, there exists a key $d \in K$ such that for each message $m \in M$:

$$D_d(E_e(m)) = m. \tag{1}$$

Cryptosystem

Definition

- A *cryptosystem* is called *symmetric* (or *"private-key"*) if $d = e$, or if d can at least be "easily" computed from e.
- A *cryptosystem* is called *asymmetric* (or *"public-key"*) if $d \neq e$, and it is "practically infeasible" to compute d from e. Here, d is the *private key*, and e is the *public key*.

Types of Attack

- Ciphertext-Only Attack
  - Known: some ciphertexts
  - Determine: the corresponding plaintext/keys
- Known-Plaintext Attack
  - Known: $(p_1, c_1), (p_2, c_2), \ldots, (p_k, c_k)$
  - Determine: the corresponding keys/other ciphertexts
- Chosen-Plaintext Attack
  - Choose: some plaintexts at will
  - Obtain: the corresponding ciphertexts
  - Determine: the corresponding keys

Types of Attack and Kerckhoffs's Principle

- Chosen-Ciphertext Attack
  - Choose: some ciphertexts at will
  - Obtain: the corresponding plaintexts
  - Determine: the corresponding keys

- Key-Only Attack (relevant only for public-key cryptosystems)
  - Known: the public keys
  - Determine: the corresponding private keys

Kerckhoffs's Principle:

The security of a cryptosystem must not depend on the secrecy of the system used. Rather, the security of a cryptosystem may depend only on the secrecy of the keys used.

Digital Signatures and Authentication

- Digital Signatures: Alice wants to sign her (encrypted) messages to Bob such that
  (a) Bob can verify that indeed she is the sender of the message, and
  (b) also third parties (who perhaps do not trust Bob) can convince themselves of the authenticity of her signature.

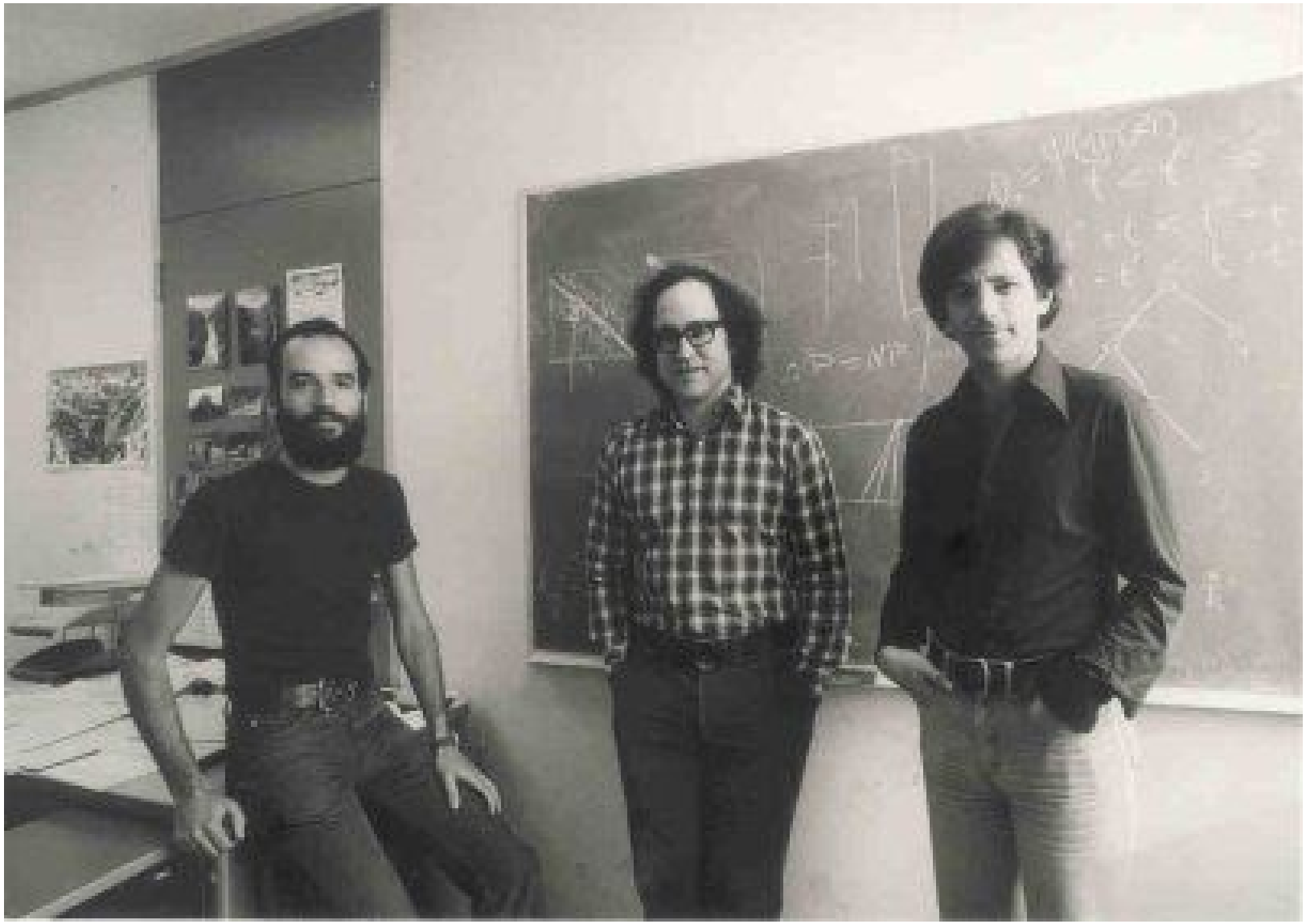
  Property (a) is already achieved by symmetric authentication codes.

- Authentication codes:
  - provide a method of ensuring the integrity of a message.
  - Active Attacks:
    - *Substitution Attack*: Erich might try to tamper with (i.e., to change or replace) the messages transmitted.
    - *Impersonation Attack* (a.k.a. "*Man-in-the-middle Attack*"): Erich might try to introduce a message of his own into the channel, hoping it is accepted as authentic by Bob.

Authentication Problems

- **Message integrity**: How can one be sure that no intruder has tampered with the message received?
- **Message authentication**: How can one be sure that a message indeed originated from the sender asserted and was not introduced by an intruder?
- **User authentication**: How can one be sure of the identity of an individual?

Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman

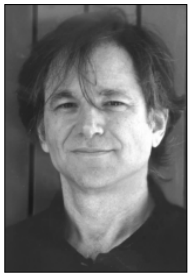


Turing Award 2002 for Rivest, Shamir, and Adleman

Rivest, Shamir, and Adleman Receive 2002 Turing Award

Ronald L. Rivest

Adi Shamir



Leonard M. Adleman

The Association for Computing Machinery (ACM) has named RONALD L. RIVEST, ADI SHAMIR, and LEONARD M. ADLEMAN as winners of the 2002 A. M. Turing Award, considered the "Nobel Prize of Computing", for their contributions to public key cryptography. The Turing Award carries a $100,000 prize, with funding provided by Intel Corporation.

As researchers at the Massachusetts Institute of Technology in 1977, the team developed the RSA code, which has become the foundation for an entire generation of technology security products. It has also inspired important work in both theoretical computer science and mathematics. RSA is an algorithm—named for Rivest, Shamir, and Adleman—that uses number theory to provide a pragmatic approach to secure transactions. It is today's most widely used encryption method, with applications in Internet browsers and servers, electronic transactions in the credit card industry, and products providing email services.

Rivest is the Viterbi Professor of Computer Science in MIT's Department of Electrical Engineering and Computer Science. He is a founder of MIT's Cryptography and Information Security Group. He received a B.A. in mathematics from Yale University and a Ph.D. in computer science from Stanford University.

Shamir is the Borman Professor in the Applied Mathematics Department of the Weizmann Institute of Science in Israel. He received a B.S. in mathematics from Tel Aviv University and a Ph.D. in computer science from the Weizmann Institute.

Adleman is the Distinguished Henry Salvatori Professor of Computer Science and Professor of Molecular Biology at the University of Southern California. He earned a B.S. in mathematics at the University of California, Berkeley, and a Ph.D. in computer science, also at Berkeley.

The ACM presented the Turing Award on June 7, 2003, in conjunction with the Federated Computing Research Conference in San Diego, California. The award was named for Alan M. Turing, the British mathematician who articulated the mathematical foundation and limits of computing and who was a key contributor to the Allied cryptanalysis of the German Enigma cipher during World War II. Since its inception in 1966, the ACM's Turing Award has honored the computer scientists and engineers who created the systems and underlying theoretical foundations that have propelled the information technology industry.

—From an ACM news release

Historical Notes on RSA

- The RSA public-key cryptosystem and the related digital signature scheme are due to **R**ivest, **S**hamir, and **A**dleman (1978), who received the Turing Award in 2002.
- This is the very first public-key cryptosystem in the open literature.
- The idea of public-key cryptography was first published by Diffie and Hellman (1976).
- Decades later, in December of 1997, the British Government Communications Headquarters (GCHQ) revealed that Ellis, Cocks, and Williamson, employed at the Communications Electronics Security Group of GCHQ, had independently and even earlier discovered
  - the principle of public-key cryptography (1969 by Ellis),
  - the cryptosystem now called RSA (1973 by Cocks), and
  - the secret-key agreement protocol now called Diffie–Hellman (1976 by Williamson).

Reminder: Some Mathematical Foundations

- For $k \geq 1$, define the multiplicative group

$$\mathbb{Z}_k^* = \{i \mid 1 \leq i \leq k-1 \text{ and } \gcd(i,k) = 1\}.$$

- Recall the *extended Euclidean Algorithm*.

- Recall the *Euler function* φ, which gives the order of the group $\mathbb{Z}_k^*$, i.e., $\varphi(k) = \|\mathbb{Z}_k^*\|$. By definition, we have:
  - $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$ for each $m, n \in \mathbb{N}$ with $\gcd(m,n) = 1$, and
  - $\varphi(p) = p - 1$ for each prime p.

  These properties immediately imply that if $n = p \cdot q$ for prime numbers p and q, then

$$\varphi(n) = (p-1)(q-1).$$

Reminder: Some Mathematical Foundations

Theorem (Euler)

For each a with $\gcd(a,n)=1$, $a^{\varphi(n)} \equiv 1 \bmod n$. **without proof**

Example (application of Euler's theorem)

What is $103^{1025} \bmod 51$?

$$\begin{aligned} n &= 51 = 3 \cdot 17 = p \cdot q \\ \varphi(n) &= \varphi(51) = (p-1)(q-1) = 2 \cdot 16 = 32. \end{aligned}$$

By Euler's theorem, we have $103^{32} \equiv 1 \bmod 51$, which implies

$$103^{1025} \equiv 103^{32 \cdot 32+1} \equiv 1^{32} \cdot 103 \equiv 103 \equiv 1 \bmod 51.$$

Reminder: Some Mathematical Foundations

Remark:

- Euler's theorem is a special case of Lagrange's theorem, which states that for every finite group $\mathfrak{G}$ of order k, the order of each subgroup of $\mathfrak{G}$ divides k.
- Define the *order of an element x of $\mathfrak{G}$* to be the smallest positive integer k such that $x^k = \underbrace{x \circ x \circ \cdots \circ x}_{k \text{ times}} = e$.
- Since the order of any group element a is the order of the subgroup generated by a, it follows that the order of a divides k. Letting e denote the neutral element of $\mathfrak{G}$, we have $a^k = e$. Since $\mathbb{Z}_n^*$ is a finite multiplicative group of order $\varphi(n)$, we have proven Euler's theorem.
- The special case of Euler's theorem with a prime number n coprime with a is known as Fermat's Little Theorem.

Reminder: Some Mathematical Foundations

Corollary (Fermat's Little Theorem)

If p is prime and a an integer with $\gcd(a,p)=1$, then $a^{p-1} \equiv 1 \bmod p$.

Proof: To simplify the proof, we assume

- first that $0 \leq a \leq p-1$ (otherwise, simply reduce a modulo p);
- second that even $1 \leq a \leq p-1$ (indeed, $a^{p-1} \equiv 1 \bmod p$ is the same as $a^p \equiv a \bmod p$, and if $a=0$, this holds trivially).

Now, let k be the order of a, i.e., k is the smallest positive integer such that $a^k \equiv 1 \bmod p$.

Thus $1, a, a^2, \ldots, a^{k-1}$ reduced modulo p form a subgroup of $\mathbb{Z}_p^*$ whose order is k.

Reminder: Some Mathematical Foundations

By Lagrange's theorem, k divides the order of $\mathbb{Z}_p^*$, which is $p-1$.

So $p-1 = km$ for some positive integer m.

It follows that

$$a^{p-1} \equiv a^{km} \equiv (a^k)^m \equiv 1^m \equiv 1 \bmod p. \quad \square$$

Reminder: Some Mathematical Foundations

Theorem (Chinese Remainder Theorem)

Let $m_1, m_2, \ldots, m_k$ be k positive integers that are pairwise relatively prime (i.e., $\gcd(m_i, m_j) = 1$ for $i \neq j$), let

$$M = \prod_{i=1}^{k} m_i,$$

and let $a_1, a_2, \ldots, a_k$ be any integers. For each i with $1 \leq i \leq k$, define $q_i = {}^{M}/_{m_i}$, and let q_i^{-1} denote the inverse element of q_i in $\mathbb{Z}_{m_i}^*$.
Then, the system of k congruences $x \equiv a_i \bmod m_i$, where $1 \leq i \leq k$, has the unique solution

$$x = \sum_{i=1}^{k} a_i q_i q_i^{-1} \bmod M.$$

without proof

Reminder: Some Mathematical Foundations

Example (Chinese Remainder Theorem)

We want to solve the following system of congruences:

$$\begin{aligned} x &\equiv 2 \bmod 4 \\ x &\equiv 1 \bmod 3 \\ x &\equiv 0 \bmod 5 \end{aligned}$$

Thus $M = 4 \cdot 3 \cdot 5 = 60$ and

$$q_1 = \frac{60}{4} = 15, \quad q_2 = \frac{60}{3} = 20, \quad q_3 = \frac{60}{5} = 12.$$

Reminder: Some Mathematical Foundations

Example (Chinese Remainder Theorem, continued)

For each $i \in \{1,2,3\}$, solve

$$q_i q_i^{-1} \equiv 1 \bmod m_i.$$

That is,

$$\begin{aligned} -q_1^{-1} &\equiv 1 \bmod 4 && \text{has the solution } q_1^{-1} = -1, \\ -q_2^{-1} &\equiv 1 \bmod 3 && \text{has the solution } q_2^{-1} = -1, \\ 2q_3^{-1} &\equiv 1 \bmod 5 && \text{has the solution } q_3^{-1} = 3. \end{aligned}$$

Hence,

$$x = -2 \cdot 15 - 1 \cdot 20 = -50 = 10 \bmod 60.$$

That is, $x = 10$ solves the system of congruences simultaneously.

RSA Public-Key Cryptosystem

Step	Alice	Erich	Bob
1			chooses two large primes, p and q, at random, computes $n = pq$ and $\varphi(n) = (p-1)(q-1)$, his public key (n, e), and his private key d satisfying (2) and (3)
2		$\Leftarrow (n, e)$	
3	encrypts m as $c = m^e \bmod n$		
4		$c \Rightarrow$	
5			decrypts c as $m = c^d \bmod n$

Table: RSA protocol

RSA Public-Key Cryptosystem

1. **Key Generation.**
   - Bob chooses two distinct large prime numbers, p and q with $p \neq q$, and computes their product $n = pq$.
   - Then, he chooses an exponent $e \in \mathbb{N}$ satisfying

$$1 < e < \varphi(n) = (p-1)(q-1) \quad \text{and} \quad \gcd(e, \varphi(n)) = 1. \tag{2}$$

   - Using the extended Euclidean Algorithm, he then determines the inverse element of e mod $\varphi(n)$, i.e., the unique number d satisfying

$$1 < d < \varphi(n) \quad \text{and} \quad e \cdot d \equiv 1 \bmod \varphi(n). \tag{3}$$

   - The pair (n, e) is Bob's public key, and d is Bob's private key.
2. **Communication.** Bob makes his key (n, e) public.

RSA Public-Key Cryptosystem

3. **Encryption.**
   - Messages are strings over an alphabet Σ, which can be viewed as natural numbers in $\|\Sigma\|$-adic representation.
   - Every message can be encoded block-wise with a fixed block length.
   - Let $m < n$ be the number encoding one block of the message Alice wants to send to Bob.
   - Alice knows Bob's public key (n, e) and encrypts m as the number $c = E_{(n,e)}(m)$, where the encryption function is defined by

$$E_{(n,e)}(m) = m^e \bmod n. \tag{4}$$

4. **Communication.** Alice sends her encrypted message c to Bob.

RSA Public-Key Cryptosystem

5. **Decryption.**
   - Let c with $0 \leq c < n$ be the number encoding one block of the ciphertext that Bob receives.
   - The eavesdropper Erich may also know c, but he does not know Bob's private key d.
   - Bob decrypts c using d and the following decryption function:

$$D_d(c) = c^d \bmod n. \tag{5}$$

RSA Public-Key Cryptosystem

Theorem

Let (n, e) be the public key, and let d be the private key used in the RSA protocol. Then, for each message m with $0 \leq m < n$,

$$m = (m^e)^d \bmod n.$$

Proof: By (3), we have $e \cdot d \equiv 1 \bmod \varphi(n)$.

Thus, there exists an integer k such that

$$e \cdot d = 1 + k(p-1)(q-1),$$

where $n = pq$.

RSA Public-Key Cryptosystem

Hence, we have

$$\begin{aligned}(m^e)^d &= m^{e\cdot d} &&= m^{1+k(p-1)(q-1)} \\ &&&= m\left(m^{k(p-1)(q-1)}\right) \\ &&&= m\left(m^{p-1}\right)^{k(q-1)}.\end{aligned}$$

It follows that

$$(m^e)^d \equiv m \bmod p, \tag{6}$$

since

- if p divides m then both sides of (6) are congruent to 0 mod p, and
- if p does not divide m (i.e., $\gcd(p, m) = 1$) then we have

$$m^{p-1} \equiv 1 \bmod p \tag{7}$$

by Fermat's Little Theorem.

RSA Public-Key Cryptosystem

A symmetric argument shows that

$$(m^e)^d \equiv m \bmod q. \tag{8}$$

Since p and q are distinct primes, (7) and (8) imply via the Chinese Remainder Theorem that

$$(m^e)^d \equiv m \bmod n.$$

Since $m < n$, the proof is complete. ❑

RSA Public-Key Cryptosystem

Example (applying the Chinese Remainder Theorem in the previous proof)

Consider $x = m^{e \cdot d}$ with $p = 3$ and $q = 5$ (so $n = 3 \cdot 5 = 15$) and $m = 11$.

Then

$$5^{-1} \bmod 3 = 2 \quad \text{and} \quad 3^{-1} \bmod 5 = 2.$$

By the Chinese Remainder Theorem,

$$x = 11 \cdot 5 \cdot 2 + 11 \cdot 3 \cdot 2 = 110 + 66 = 176 \equiv 11 \bmod 15.$$

RSA Public-Key Cryptosystem

Remark: Alice has to compute $c = m^e \bmod n$ and Bob has to compute $m = c^d \bmod n$. Performed naively, these computations would require a large number of multiplications depending on the size of the exponent.

Fortunately, however, the modular exponentiation function can be computed efficiently by employing the *"square-and-multiply."*

The "Square-and-Multiply" Algorithm

SQUARE-AND-MULTIPLY(a, b, m) {

(a is the exponent, $b < m$ is the base, and m is the modulus *)*

Determine the binary expansion of exponent $a = \sum_{i=0}^{k} a_i 2^i$, $a_i \in \{0,1\}$;

Successively, compute $b^{2^0}, b^{2^1}, \ldots, b^{2^k}$ by applying the congruence

$$b^{2^{i+1}} \equiv \left(b^{2^i}\right)^2 \bmod m;$$

(the intermediate values b^{2^i} need not be stored *)*

In the arithmetics modulo m, compute $b^a = \prod_{\substack{i=0 \\ a_i=1}}^{k} b^{2^i}$;

`return` b^a;

}

Figure: The "square-and-multiply" algorithm

RSA Public-Key Cryptosystem

Remark: The computation of b^a mod m in this algorithm is correct, since in the arithmetics modulo m,

$$b^a = b^{\sum_{i=0}^{k} a_i 2^i} = \prod_{i=0}^{k} \left(b^{2^i}\right)^{a_i} = \prod_{\substack{i=0 \\ a_i=1}}^{k} b^{2^i}.$$

RSA Public-Key Cryptosystem

Example ("square-and-multiply")

Suppose we want to determine

$$2^{17} \bmod 10.$$

Binary expansion of the exponent: $17 = 1 \cdot 2^0 + 1 \cdot 2^4$.

Successively, compute the squares (modulo 10):

- $2^{2^0} = 2$
- $2^{2^1} = 4$
- $2^{2^2} = 6$
- $2^{2^3} = 6$
- $2^{2^4} = 6$

$$\Longrightarrow \quad 2^{17} \bmod 10 = 2 \cdot 6 = 12 \equiv 2 \bmod 10.$$

We have to compute four squares and one multiplication (instead of 16 multiplications).

Example of an RSA Encryption

Example

- Bob chooses the primes $p = 67$ and $q = 11$ and computes $n = 67 \cdot 11 = 737$ and $\varphi(n) = (p-1)(q-1) = 66 \cdot 10 = 660$.
- If Bob now chooses the smallest possible exponent for $\varphi(n) = 660$, which is $e = 7$, then his public key is the pair $(n, e) = (737, 7)$.
- Using the extended Euclidean Algorithm, Bob determines his private key $d = 283$, and we have:

$$e \cdot d = 7 \cdot 283 = 1981 \equiv 1 \bmod 660.$$

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7				

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7				
7	2				

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7				
7	2				
2	1				

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7				
7	2				
2	1				
1	0	1	1	0	`if` $(m = 0)$ `return` $(n, 1, 0)$

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7				
7	2				
2	1	1	0	1	$x := y'; \quad y := x' - y' * \lfloor \frac{n}{m} \rfloor$
1	0	1	1	0	`if` $(m = 0)$ `return` $(n, 1, 0)$

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7				
7	2	1	1	-3	
2	1	1	0	1	$x := y'; \quad y := x' - y' * \lfloor \frac{n}{m} \rfloor$
1	0	1	1	0	`if` $(m = 0)$ `return` $(n, 1, 0)$

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7	1	-3	283	
7	2	1	1	-3	
2	1	1	0	1	$x := y'; \quad y := x' - y' * \lfloor \frac{n}{m} \rfloor$
1	0	1	1	0	`if` $(m = 0)$ `return` $(n, 1, 0)$

Table: Extended Euclidean Algorithm

Example of an RSA Encryption

Why? What is the greatest common divisor of $n = 660$ and $m = 7$?

n	m	g	x	y	Remark
660	7	1	-3	283	
7	2	1	1	-3	
2	1	1	0	1	$x := y'; \quad y := x' - y' * \lfloor \frac{n}{m} \rfloor$
1	0	1	1	0	`if` $(m = 0)$ `return` $(n, 1, 0)$

Table: Extended Euclidean Algorithm

This result indeed is correct, since

$$(-3) \cdot 660 + 283 \cdot 7 = 1 = \gcd(660, 7).$$

Example of an RSA Encryption

- Identify the alphabet $\Sigma = \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\}$ with the set
$$\mathbb{Z}_{26} = \{0, 1, \ldots, 25\}.$$

- Any block
$$b = b_1 b_2 \cdots b_\ell$$
of length ℓ with $b_i \in \mathbb{Z}_{26}$ is represented by the integer
$$m_b = \sum_{i=1}^{\ell} b_i \cdot 26^{\ell - i}.$$

- From the definition of the block length $\ell = \lfloor \log_{26} n \rfloor$, we have
$$0 \leq m_b \leq 25 \cdot \sum_{i=1}^{\ell} 26^{\ell - i} = 26^\ell - 1 < n.$$

Example of an RSA Encryption

- Using the RSA encryption function (4), the integer m_b corresponding to the block b is encrypted by

$$c_b = (m_b)^e \bmod n,$$

where $c_b = c_0 c_1 \cdots c_\ell$ with $c_i \in \mathbb{Z}_{26}$ is the ciphertext for block b.

- RSA thus maps blocks of length ℓ injectively to blocks of length $\ell + 1$.

- The resulting integer c_b is again written in 26-adic representation and may have length $\ell + 1$:

$$c_b = \sum_{i=0}^{\ell} c_i \cdot 26^{\ell - i},$$

where $c_i \in \mathbb{Z}_{26}$.

Example of an RSA Encryption

- Decryption also works block-wise, using the private key d and the RSA decryption function (5):

$$m_b = (c_b)^d \bmod n.$$

- Concretely, consider the following RSA encryption:

	R S	A I	S T	H E	K E	Y T	O P	U B	L I
m_b	460	8	487	186	264	643	379	521	294
c_b	697	387	229	340	165	223	586	5	189

	C K	E Y	C R	Y P	T O	G R	A P	H Y
m_b	62	128	69	639	508	173	15	206
c_b	600	325	262	100	689	354	665	673

Table: Example of an RSA encryption

Example of an RSA Encryption

- Consider the first block: $b = \text{RS} \mathrel{\widehat{=}} 17\ 18$, which is turned into an integer as follows:

$$m_b = 17 \cdot 26^1 + 18 \cdot 26^0 = 442 + 18 = 460,$$

which in turn is encrypted as (using square-and-multiply):

$$c_b = (m_b)^e \bmod n = 460^7 \bmod 737 = 697,$$

which again is written in 26-adic representation and may have length $\ell + 1$:

$$c_b = \sum_{i=0}^{\ell} c_i \cdot 26^{\ell - i},$$

where $c_i \in \mathbb{Z}_{26}$.

Example of an RSA Encryption

- In particular, the first block

$$697 = 676 + 21 = 1 \cdot 26^2 + 0 \cdot 26^1 + 21 \cdot 26^0$$

  is turned into the ciphertext "BAV."

- Decryption also works block-wise. For instance, to decrypt the first block using the private key $d = 283$, compute

$$697^{283} \bmod 737$$

  again employing fast exponentiation (square-and-multiply).

- It is useful to reduce modulo $n = 737$ after each multiplication to prevent the integers from becoming too large.

Example of an RSA Encryption

- The binary expansion of the exponent is

$$283 = 2^0 + 2^1 + 2^3 + 2^4 + 2^8,$$

and we obtain

$$\begin{aligned} 697^{283} &\equiv 697^{2^0} \cdot 697^{2^1} \cdot 697^{2^3} \cdot 697^{2^4} \cdot 697^{2^8} \\ &\equiv 697 \cdot 126 \cdot 9 \cdot 81 \cdot 15 \\ &\equiv 460 \bmod 737 \end{aligned}$$

as desired.

RSA Digital Signature Scheme

Step	Alice	Erich	Bob
1	chooses $n = pq$, her public key (n, e) and her private key d just as Bob does in the RSA protocol		
2		$(n, e) \Rightarrow$	
3	signs the message m with $\text{sig}_A(m) = m^d \bmod n$		
4		$\langle m, \text{sig}_A(m) \rangle \Rightarrow$	
5			verifies Alice's signature by $m \equiv (\text{sig}_A(m))^e \bmod n$

Table: RSA digital signature scheme

RSA Digital Signature Scheme

Remark: This method

$$\text{PKCS} \implies \text{Digital Signature}$$

works whenever encryption and decryption are exchangeable:

$$m = E_e(D_d(m)).$$

For RSA, this property is satisfied because

$$(m^d)^e \equiv (m^e)^d \equiv m \bmod n.$$

Then $s = D_d(m)$ is the signature of m.

Verification:

$$m = E_e(s) = E_e(D_d(m)).$$

Factoring Attacks on RSA

- Brute-force attack
- Special-purpose factoring methods
  - Pollard's $p-1$ method (works well if $n = pq$ and $p-1$ has only small prime factors)
  - Lenstra's elliptic curve method (generalizes Pollard's $p-1$ method and is the more effective for breaking RSA, the smaller the smallest prime factor of n is)
- General-purpose factoring methods
  - quadratic sieve
  - number field sieve
- Using the Euler function to factor n

Using the Euler Function to Factor n

- If the attacker can factor $n = pq$, he can efficiently determine $\varphi(n) = (p-1)(q-1)$ and thus break RSA.
- Conversely, if he knows $\varphi(n)$, he can efficiently factor n.
- That is, factoring the RSA modul n and computing $\varphi(n)$ are "equally hard" problems.
- Suppose the attacker knows both $n = pq$ and $\varphi(n)$.
- He can then determine the prime factors of $n = pq$ by solving the following two equations for the unknowns p and q:

$$\begin{aligned} n &= p \cdot q \\ \varphi(n) &= (p-1)(q-1). \end{aligned}$$

Using the Euler Function to Factor n

- Substituting $q = n/p$ into the second equation gives a quadratic equation in p:

$$p^2 - (n - \varphi(n) + 1)p + n = 0. \tag{9}$$

  By Vieta's Theorem, p and q are the solutions of a quadratic equation of the form $p^2 + ap + b = 0$ if and only if
  - $p + q = -a$ and
  - $pq = b$.

  Since the prime factors p and q of n satisfy both $pq = n$ and

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

  (9) has the roots p and q.

- Thus a cryptanalyst who knows $\varphi(n)$ can easily break RSA.

Using the Euler Function to Factor n

Example: Let $n = 60477719$.

Suppose that Erich was able to determine the value $\varphi(n) = 60462000$.

By (9), he can determine the prime factors of n simply by solving the quadratic equation

$$p^2 - 15720p + 60477719 = 0$$

as follows:

$$p = \frac{15720}{2} + \sqrt{\left(\frac{15720}{2}\right)^2 - 60477719} = 9001 \quad \text{and}$$

$$q = \frac{15720}{2} - \sqrt{\left(\frac{15720}{2}\right)^2 - 60477719} = 6719.$$

Other Attacks on RSA

- Superencryption
- Small-Message Attack
- Wiener's Attack
- Low-Exponent Attack
- Forging RSA Signatures by a Chosen-Plaintext Attack

Superencryption

- Simmons and Norris proposed an attack on RSA as early as 1977, shortly after the invention of RSA.
- Their attack, called *superencryption*, is based on the observation that a sufficient number of encryptions, cycling through $\mathbb{Z}_n$, may eventually recover the original message m.
- This attack is a threat to the security of RSA, provided that the number of encryptions required to recover m is small.
- Fortunately, if the primes p and q are large and are chosen at random, then superencryption is not a practical attack.

Superencryption

Example

- Let $n = 5 \cdot 7 = 35$, so $\varphi(n) = 4 \cdot 6 = 24$.
- Choose the encryption exponent $e = 5$; note that

$$\gcd(24, 5) = 1.$$

- Encrypting the message $m = 11$ yields

$$11^5 \bmod 35 = 16.$$

- Now, encrypting the message $m' = 16$ recovers the original message:

$$16^5 \bmod 35 = 11,$$

Superencryption

- which actually is not suprising, since the decryption key d happens to be equal to e in this case: $5^2 \bmod 24 = 1$, so $d = 5 = e$. In fact, every number e with $\gcd(24, e) = 1$ equals its inverse modulo 24.
- So, let us now choose $n = 11 \cdot 13 = 143$; thus, $\varphi(n) = 10 \cdot 12 = 120$.
- The encryption exponent $e = 7$ has the inverse $d = 103$ modulo 120, so $e \neq d$ in this case.
- Still, encrypting the message $m = 11$ now yields

$$11^7 \bmod 143 = 132 \quad \text{and} \quad 132^7 \bmod 143 = 11.$$

- Thus, without knowing the private key $d = 103$, a cryptanalyst can recover the original message simply by a double encryption.

Small-Message Attack

- If both the message m to be encrypted and the encryption exponent e are small relative to the modulus n, then the RSA encryption is not effective.
- In particular, if the ciphertext $c = m^e$ is smaller than n, then m can be recovered from c by ordinary root extraction.
- To prevent this from happening, the public exponent should be large or the messages to be encrypted should always be large.
- It is this latter suggestion that is more useful, since a small public exponent is often preferred in order to speed up encryption and to preclude Wiener's attack.

Wiener's Attack: Rough Sketch of the Idea

- Wiener's attack (1990) uses a continued fraction approximation and the public key (n, e) so as to compute the private key d.
- It is a concern only if d is too small relative to n.
- More precisely, Wiener's attack works if and only if

$$3d < \sqrt[4]{n} \qquad \text{and} \qquad q < p < 2q, \tag{10}$$

  where $n = pq$.
- Since the encryption and decryption exponent satisfy

$$ed \equiv 1 \bmod \varphi(n),$$

  there is some integer $k < d$ such that

$$ed - k\varphi(n) = 1,$$

Wiener's Attack: Rough Sketch of the Idea

- which implies
$$\left|\frac{e}{\varphi(n)} - \frac{k}{d}\right| = \frac{1}{d\varphi(n)}. \tag{11}$$
- Since $n = pq > q^2$, we have $q < \sqrt{n}$.
- Since $q < p < 2q$ by (10), we have
$$0 < n - \varphi(n) = p + q - 1 < 2q + q - 1 < 3q < 3\sqrt{n}.$$
- Hence,
$$\left|\frac{e}{n} - \frac{k}{d}\right| = \left|\frac{ed - kn}{dn}\right| = \left|\frac{1 + k(\varphi(n) - n)}{dn}\right| \tag{12}$$
$$< \frac{3k\sqrt{n}}{dn} = \frac{3k}{d\sqrt{n}} < \frac{1}{d\sqrt[4]{n}}, \tag{13}$$

  where the latter inequality follows from $3k < 3d < \sqrt[4]{n}$, which is implied by $k < d$ and (10).

Wiener's Attack: Rough Sketch of the Idea

- Again, since $3d < \sqrt[4]{n}$, we have

$$\left|\frac{e}{n} - \frac{k}{d}\right| < \frac{1}{3d^2}. \tag{14}$$

- Note that the encryption key (n, e) is public.
- Inequality (14) says that the fraction k/d is a very close approximation to the fraction e/n.
- Hence, to recover the private key d from the public key (n, e), an attacker might employ the following fact:

> Every approximation of e/n that is as close as shown in (14) must be one of the convergents of the continued fraction expansion of e/n.

Wiener's Attack: Rough Sketch of the Idea

- A (finite) *continued fraction* is the rational number

$$c_1 + \cfrac{1}{c_2 + \cfrac{1}{c_3 + \cdots + \cfrac{1}{c_t}}}, \tag{15}$$

  which is represented as the t-tuple

$$(c_1, c_2, \ldots, c_t)$$

  of nonnegative integers, where $c_t \neq 0$.

- For example, the continued fraction expansion of $101/37$ is

$$(2,1,2,1,2,3),$$

  which means that $\dfrac{101}{37} = 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{2 + \frac{1}{3}}}}}$.

Wiener's Attack: Rough Sketch of the Idea

- Why? Suppose that a and b are positive integers satisfying $\gcd(a, b) = 1$, and let $r_0, r_1, \ldots, r_t$ be the sequence of integers generated by running $\textsc{Euclid}(a, b)$.
- That is, $r_0 = a$, $r_1 = b$, and for $1 \leq i < t$:

$$r_{i+1} \equiv r_{i-1} \bmod r_i.$$

- Let $c_i = \left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor$ for $1 \leq i \leq t$.
- Then, $\frac{a}{b}$ equals the continued fraction from (15), and

$$(c_1, c_2, \ldots, c_t)$$

is said to be the *continued fraction expansion of* $\frac{a}{b}$.

Wiener's Attack: Rough Sketch of the Idea

- $(2,1,2,1,2,3)$ is the continued fraction expansion of $\frac{101}{37}$:

i	0	1	2	3	4	5	6
r_i	101	37	27	10	7	3	1
c_i		2	1	2	1	2	3

Wiener's Attack: Rough Sketch of the Idea

- $(2,1,2,1,2,3)$ is the continued fraction expansion of $\frac{101}{37}$:

i	0	1	2	3	4	5	6
r_i	101	37	27	10	7	3	1
c_i		2	1	2	1	2	3

- For each i, $1 \leq i \leq t$, $\mathbf{C}_i = (c_1, c_2, \ldots, c_i) = \frac{x_i}{y_i}$ is the *i^{th} convergent of* $(c_1, c_2, \ldots, c_t)$, where x_i and y_i are the solutions of these recurrences:

$$x_i = \begin{cases} 1 & \text{if } i = 0 \\ c_1 & \text{if } i = 1 \\ c_i x_{i-1} + x_{i-2} & \text{if } i \geq 2 \end{cases} \qquad y_i = \begin{cases} 0 & \text{if } i = 0 \\ 1 & \text{if } i = 1 \\ c_i y_{i-1} + y_{i-2} & \text{if } i \geq 2. \end{cases} \quad (16)$$

Wiener's Attack: Rough Sketch of the Idea

- $(2,1,2,1,2,3)$ is the continued fraction expansion of $\frac{101}{37}$:

i	0	1	2	3	4	5	6
r_i	101	37	27	10	7	3	1
c_i		2	1	2	1	2	3
$\mathbf{C}_i = \frac{x_i}{y_i}$		$\frac{2}{1}$	$\frac{3}{1}$	$\frac{8}{3}$	$\frac{11}{4}$	$\frac{30}{11}$	$\frac{101}{37}$

- For each i, $1 \leq i \leq t$, $\mathbf{C}_i = (c_1, c_2, \ldots, c_i) = \frac{x_i}{y_i}$ is the i^{th} *convergent of* $(c_1, c_2, \ldots, c_t)$, where x_i and y_i are the solutions of these recurrences:

$$x_i = \begin{cases} 1 & \text{if } i = 0 \\ c_1 & \text{if } i = 1 \\ c_i x_{i-1} + x_{i-2} & \text{if } i \geq 2 \end{cases} \qquad y_i = \begin{cases} 0 & \text{if } i = 0 \\ 1 & \text{if } i = 1 \\ c_i y_{i-1} + y_{i-2} & \text{if } i \geq 2. \end{cases} \quad (16)$$

Wiener's Attack: Rough Sketch of the Idea

Theorem

If a, b, c, and d are positive integers such that

$$\left|\frac{a}{b} - \frac{c}{d}\right| < \frac{1}{2d^2} \text{ and } \gcd(a,b) = \gcd(c,d) = 1,$$

then $\frac{c}{d}$ is one of the convergents of the continued fraction expansion of $\frac{a}{b}$. **without proof**

- By this theorem, (14) implies that $\frac{k}{d}$ is one of the convergents of the continued fraction expansion of $\frac{e}{n}$.
- Since $\frac{e}{n}$ is known, all one has to do to determine $\frac{k}{d}$ is to compute all convergents of $\frac{e}{n}$ and to check if one of them is the correct one.

Wiener's Attack: Rough Sketch of the Idea

- To this end, if some convergent $\mathbf{C}_i = \frac{x_i}{y_i}$ of $\frac{e}{n}$ is suspected to be equal to $\frac{k}{d}$, one computes the value of $\varphi(n)$ by

$$\varphi(n) = \frac{e \cdot d - 1}{k} = \frac{e \cdot y_i - 1}{x_i}.$$

- Once both n and $\varphi(n)$ are known, n can be factored by solving the quadratic equation (9), whose roots will be the prime factors of n.
- If this test fails, then $\mathbf{C}_i$ was not the correct convergent, and one proceeds to check the next suspect.
- If none of the convergents of $\frac{e}{n}$ was tested successfully, one concludes that the assumptions made in (10) do not apply.

Wiener's Attack: Example

- Let $n = 60477719$ and $e = 47318087$, so the public key is $(n, e) = (60477719, 47318087)$.
- Thus, a cryptanalyst knows the value

$$\frac{e}{n} = \frac{47318087}{60477719} = 0.78240528549.$$

- Running $\textsc{Euclid}(47318087, 60477719)$ and computing the values r_i and c_i as above gives the following continued fraction expansion of $\frac{e}{n}$:

$$(0, 1, 3, 1, 1, 2, 8, 1, 9, 4, 1, 4, 1, 1, 4, 2, 1, 1, 2, 2, 3). \tag{17}$$

Wiener's Attack: Example

- Now, using the recurrences (16) to compute the x_i and y_i, one can determine the 21 convergents $\mathbf{C}_i = \frac{x_i}{y_i}$ of this continued fraction expansion of $\frac{e}{n}$. The following table shows the first 10 convergents.

i	1	2	3	4	5	6	7	8	9	10	$\cdots$
c_i	0	1	3	1	1	2	8	1	9	4	$\cdots$
$\mathbf{C}_i = \frac{x_i}{y_i}$	0	1	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{7}{9}$	$\frac{18}{23}$	$\frac{151}{193}$	$\frac{169}{216}$	$\frac{1672}{2137}$	$\frac{6857}{8764}$	$\cdots$

- Each convergent is a suspect of being equal to $\frac{k}{d}$, and one after the other is to be checked. The first five tests will fail.
- However, when checking $\mathbf{C}_6 = \frac{18}{23}$, one obtains

$$\varphi(n) = \frac{e \cdot y_6 - 1}{x_6} = \frac{47318087 \cdot 23 - 1}{18} = 60462000.$$

Wiener's Attack: Example

- The cryptanalyst proceeds to compute the prime factors 6719 and 9001 of $n = 60477719$.
- Note that Wiener's attack works in this example, since the prime factors of n are of roughly the same size and (10) is satisfied:

$$3 \cdot 23 = 69 < 88 = \left\lfloor \sqrt[4]{60477719} \right\rfloor .$$

Wiener's Attack: Example

Remark:

- Wiener's attack is a real threat only if the hypotheses in (10) are satisfied, in particular, only if $3d < \sqrt[4]{n}$.
- Since the encryption exponent e is chosen first (usually small to speed up encryption), it is unlikely that a small d will be generated.
- That is, if e is small enough, then d is likely to be large enough to resist Wiener's attack.
- One should keep in mind, though, that it might be dangerous if one seeks to speed up decryption by using a small private key d.

Low-Exponent Attack

Low-Exponent Attack

- A recommended value of the encryption exponent e that is commonly used today is $e = 2^{16} + 1$.
- One advantage of this value for e is that its binary expansion has only two ones, which implies that the "square-and-multiply" algorithm requires very few operations. Thus, encryption is very efficient.
- However, one should be cautious not to choose the public encryption exponent too small.
- A preferred value of e that has been used often in the past is $e = 3$.
- Suppose that three parties participating in the same system encrypt the same message m using the same public exponent $e = 3$, yet distinct RSA moduli, say n_1, n_2, and n_3.

Low-Exponent Attack

- Then, a cryptanalyst can easily compute m from the three ciphertexts:

$$\begin{aligned} c_1 &\equiv m^3 \bmod n_1 \\ c_2 &\equiv m^3 \bmod n_2 \\ c_3 &\equiv m^3 \bmod n_3. \end{aligned}$$

- Since the message m must be smaller than each of the moduli n_i, it follows that m^3 must be smaller than $n_1 n_2 n_3$.
- Using the Chinese Remainder Theorem, one can compute the unique solution

$$c \equiv m^3 \bmod n_1 n_2 n_3 = m^3.$$

- Hence, one can recover m from c by ordinary root extraction.

Low-Exponent Attack

- More generally, suppose that k related plaintexts are encrypted with the same exponent e:

$$\begin{aligned} c_1 &\equiv (a_1 m + b_1)^e \bmod n_1 \\ c_2 &\equiv (a_2 m + b_2)^e \bmod n_2 \\ &\vdots \\ c_k &\equiv (a_k m + b_k)^e \bmod n_k, \end{aligned}$$

where a_i and b_i, $1 \leq i \leq k$, are known constants, $k > {}^{e(e+1)}/_2$, and we have

$$\min(n_i) > 2^{e^2}.$$

Low-Exponent Attack

- Then, an attacker can solve the above system of k congruences for m in polynomial time using so-called *"lattice reduction techniques."*
- This observation was made by Håstad in the late 1980s.
- This attack is a concern if the messages are related in a known way.
- In this case, they should not be encrypted with many RSA keys of the form (n_i, e).
- A recommended countermeasure, which prevents mounting this attack in practice, is to pad the messages with pseudorandom strings prior to encryption.

Forging RSA Signatures

- We present a **chosen-plaintext attack** that is based on the fact that the RSA encryption function is a homomorphism: If (n, e) is the public key and m_1 and m_2 are two messages, then

$$m_1^e \cdot m_2^e \equiv (m_1 \cdot m_2)^e \bmod n. \tag{18}$$

- Another congruence that can easily be verified is

$$(m \cdot r^e)^d \equiv m^d \cdot r \bmod n. \tag{19}$$

- The congruences (18) and (19) can be used to mount an attack on the RSA digital signature scheme.

Forging RSA Signatures

- Given previous message-signature pairs

$$\langle m_1, \mathrm{sig}_A(m_1)\rangle, \langle m_2, \mathrm{sig}_A(m_2)\rangle, \ldots, \langle m_k, \mathrm{sig}_A(m_k)\rangle,$$

Erich can use the congruences (18) and (19) to compute a new message-signature pair $\langle m, \mathrm{sig}_A(m)\rangle$ by

$$\begin{aligned} m &= r^e \prod_{i=1}^{k} m_i^{e_i} \bmod n; \\ \mathrm{sig}_A(m) &= r \prod_{i=1}^{k} (\mathrm{sig}_A(m_i))^{e_i} \bmod n, \end{aligned}$$

where r and the e_i are arbitrary.

Forging RSA Signatures

- Hence, Erich can forge Alice's signature without knowing her private key, and Bob will not detect the forgery, since

$$m \equiv (\mathrm{sig}_A(m))^e \bmod n.$$

- The above attack looks like a known-plaintext attack at first glance.
- However, note that, in (18), even if m_1 and m_2 are meaningful plaintexts, $m_1 \cdot m_2$ usually is not. Thus, Erich can forge Alice's signature only for messages that may or may not be useful.
- However, he might choose the messages m_i so as to generate a meaningful message m with a forged digital signature.
- This chosen-plaintext attack can again be avoided by pseudorandom padding techniques that destroy the algebraic relations between messages.

A Chosen-Ciphertext Attack on RSA

- Pseudorandom padding is also a useful countermeasure against the following **chosen-ciphertext attack**:
- Erich intercepts some ciphertext c, chooses $r \in \mathbb{N}$ at random, and computes $c \cdot r^e \bmod n$, which he sends to the legitimate receiver Bob.
- By (19), Bob will decrypt the string

$$\hat{c} = c^d \cdot r \bmod n,$$

which is likely to look like a random string.

- Erich, however, if he were to get his hands on $\hat{c}$, could obtain the original message m by computing

$$m = r^{-1} \cdot c^d \cdot r \bmod n,$$

i.e., he multiplies by r^{-1}, the inverse of r modulo n.