Lecture 17
Encryption

ECE 422: Reliable and Secure Systems Design

UNIVERSITY OF ALBERTA

Instructor: An Ran Chen
Term: 2024 Winter

Schedule for today

- Key concepts from last class
- Encryption
- Symmetric encryption
  - Caesar cipher
- Man-in-the-middle attack
- Asymmetric encryption
  - Rivest–Shamir–Adleman (RSA) algorithm

Confidentiality

Confidentiality

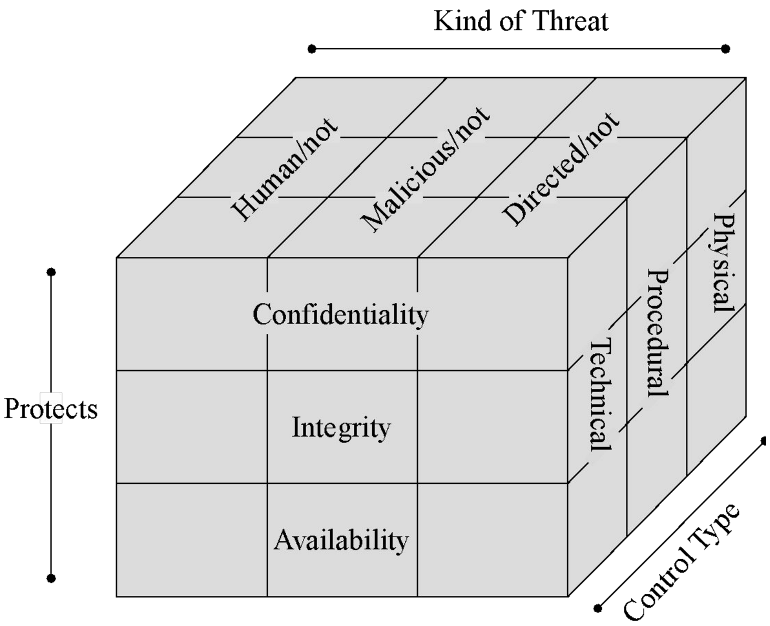
- Only the authorized user can access particular resources

Methods to achieve confidentiality:

- Encryption: encoding/decoding of the plaintext
  - E.g., Symmetric/asymmetric encryption
- Access controls: restricted access
  - E.g., Our library website
- Authentication: credentials check
  - E.g., Mobile authentication for faculty and staff

UoA's Information Services & Technology

Threats, vulnerabilities, and control types



- CIA are the basic security principles.
- Vulnerabilities are weaknesses in a system that affect the CIA triad.
- Threats exploit those weaknesses in the system.
- Controls protect those weaknesses from exploitation.

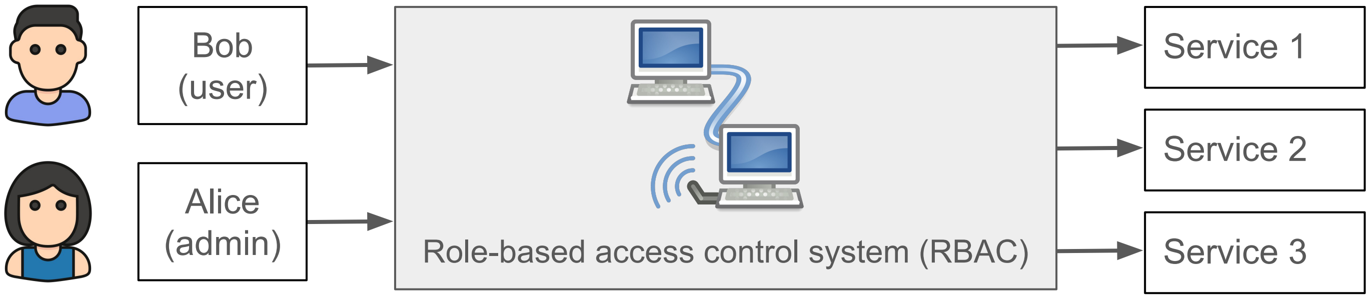
Access control list (ACL)

An access control list is a set of instructions that either allow access to a computer environment or deny it.

- Restrict access to unauthorized users
- Control traffic by limiting the number of users

It is analogous to a guest list to a wedding.

- Only those on the lists are authorized to entries



Access control models

Users receive access based on access control models.

- Different systems have different access control requirements.

There are four models of access controls:

- Discretionary access control (DAC)
- Role-based access control (RBAC)
- Mandatory access control (MAC)
- Attribute-based access control (ABAC)

Schedule for today

- Key concepts from last class
- Encryption
- Symmetric encryption
  - Caesar cipher
- Man-in-the-middle attack
- Asymmetric encryption
  - Rivest–Shamir–Adleman (RSA) algorithm

Encryption

Encryption is used to provide confidentiality.

- Encryption: from a plain text to the ciphertext
- Decryption: from the ciphertext back to the plain text

There are two types of encryption:

- Symmetric encryption
- Asymmetric encryption

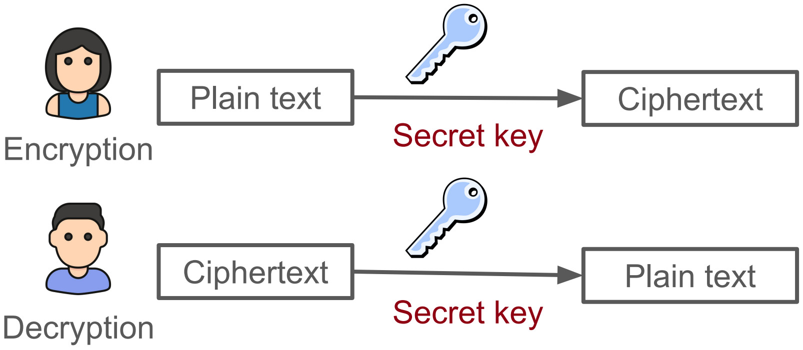
Symmetric encryption

Symmetric encryption uses a single key to encrypt and decrypt.

- Same key for encryption and decryption
- The key is kept secret
- Example of secret key: number, word, random string

Benefits of symmetric encryption include:

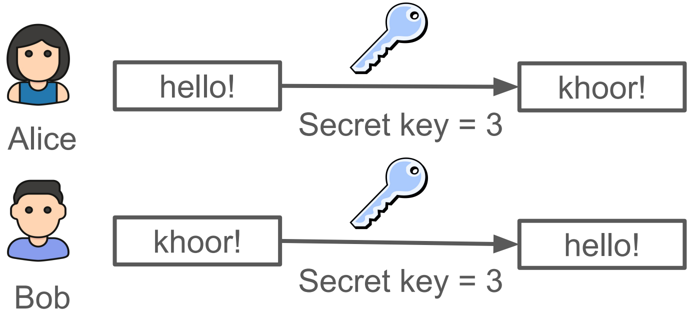
- Fast encryption and decryption
  - Inexpensive to process, single key
- Easy to implement, easy to use
  - straightforward encryption and decryption



Example of symmetric encryption

Symmetric encryption with Caesar cipher with a secret key, e = 3:

- Alice wants to send a message, m = “hello!”.
- Alice encrypts the message “hello!” into “khoor!”
    - by shifting the letters by 3 positions based on the secret key.
- Alice sends the ciphertext, c = “khoor!”, to Bob.
- Bob decrypts “khoor” with the same secret key.



Man-in-the-middle attack

In a man-in-the-middle attack, the attacker secretly intercepts and relays messages between two parties.

- Allow the attacker to eavesdrop the communication
- Sometimes worst, the attacker can intercept and then control the entire conversation

Common types of man-in-the-middle attack:

- IP spoofing (impersonate another computer system)
- Email hijacking (access to emails)
- Wifi eavesdropping
  - avoid public wifi where login is not required

Man-in-the-middle attack

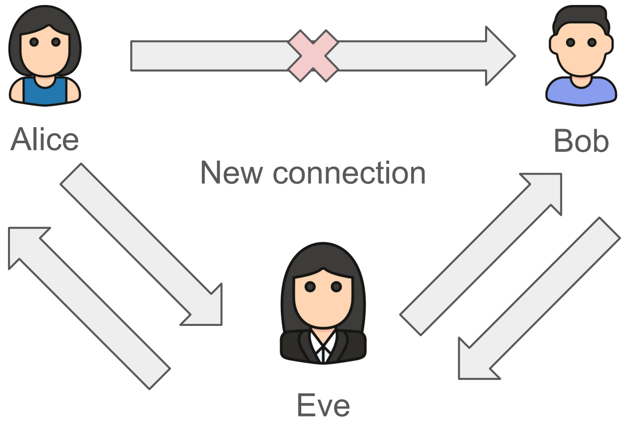
Alice wants to communicate with Bob

- Eve can intercept the messages from Alice and claim to be Bob
- At the same time, Eve can convince Bob that she is Alice

Eve can then intercept, manipulate, and relay the messages.

Solutions?

- Encryption (confidentiality)
- Digital signature (authentication)



Symmetric encryption

Problem: what happens if Eve tries to eavesdrop (adversary)?

- Eve will not be able to understand the encrypted message
- But with enough messages, Eve can comes up with the right decryption algorithm (brute force)
  - E.g., based on the frequency of the letter

Another attempt …

- We can change the encryption key everytime we send an encrypted message

A more severe problem: how do we securely distribute the secret key?

- The problem is still not solved.

Symmetric encryption

Disadvantages of symmetric encryption:

- Not scalable
  - E.g., 100 people, 100 unique keys to manage
- Key management is challenging
  - Key may be compromised, security at risk
  - Key may be lost, reissuing is expensive and time-consuming
- Lack of authentication
  - It does not verify the identity of the sender

Schedule for today

- Key concepts from last class
- Encryption
- Symmetric encryption
  - Caesar cipher
- Man-in-the-middle attack
- Asymmetric encryption
  - Rivest–Shamir–Adleman (RSA) algorithm

Asymmetric encryption

Asymmetric encryption uses a public key to encrypt and a private key to decrypt.

- Public key: anyone can see and use this key
- Private key: kept private
- Private and public keys come in pairs
- Data encrypted with the public key can only be decrypted with the private key

Suppose Alice needs to send a message to Bob

- Alice will use Bob’s public key to encrypt the message
- Bob will use his own private key to decrypt the message

Asymmetric encryption vs digital signature

Note that it is the other way around in digital signature:

- Alice will use her own private key to encrypt the message
  - Only Alice can close the envelope with her encryption key
- Bob will use Alice's public key to decrypt the message
  - Everyone is welcome to open the enclosed envelope from Alice with her public key.

In asymmetric encryption:

- Alice will use Bob's public key to encrypt the message
- Bob will use his own private key to decrypt the message
  - Only Bob can open the envelope with his private key

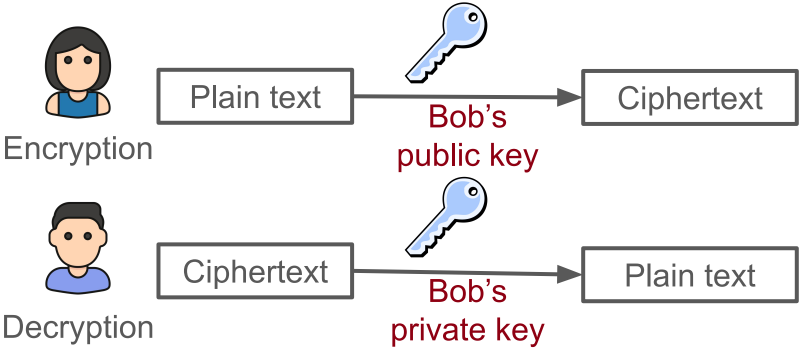
Example of asymmetric encryption

Bob generates two keys:

- public key for encryption
- private key for decryption

Alice wants to send a message to Bob

- Alice encrypts the message “hello!” with Bob’s public key.
- Alice sends the ciphertext *c* to Bob.
- Bob decrypts *c* with his private key.



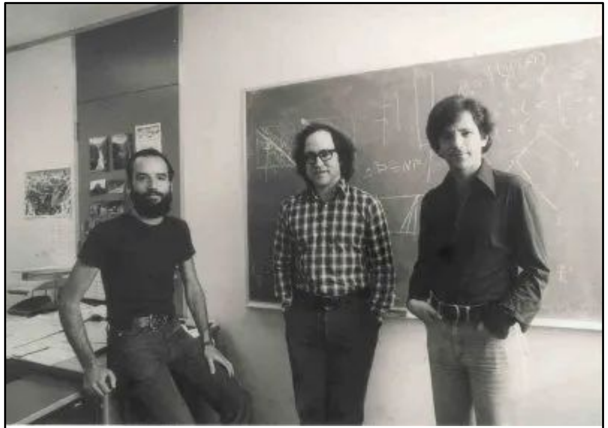
Although the encryption key is public, it is impossible to eavesdrop:

- Bob is the only one with decryption key

RSA algorithm

Rivest–Shamir–Adleman (RSA) algorithm is one of the oldest widely used for secure data transmission.

- Utilize private and public key pair
  - Private key kept secret
  - Public key is available to everyone
- Either one of the keys can be public, while the other key can be private
- Based on the factorization of large prime numbers



From left to right: Adi Shamir, Ron Rivest, and Len Adleman

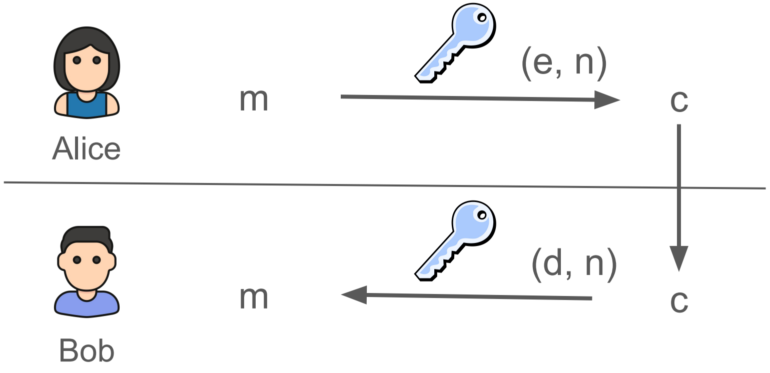
RSA algorithm

Encryption: encrypt the message m with the public key (e, n)

- $m^e \ mod(n) = c$

Decryption: decrypt the ciphertext c with the private key (d, n)

- $c^d \ mod(n) = m$



- Plain text m
- Encryption key: (e, n)
- Decryption key: (d, n)
- Ciphertext c

RSA algorithm

(e, n)

Encryption (3, 33), Plaintext = “b” = 2

Alice wants to send the number 2:

- $m^e \bmod(n) = c \rightarrow 2^3 \bmod (33) = 8$
- The ciphertext c is 8

(d, n)

Decryption (7, 33)

How does Bob decrypt the ciphertext 8?

Encryption key (e, n)

- $m^e \bmod(n) = c$

Decryption key (d, n)

- $c^d \bmod(n) = m$

RSA algorithm

Encryption (3, 33), Plaintext = “b” = 2

Alice wants to send the number 2:

- $m^e \; mod(n) = c \rightarrow 2^3 \bmod (33) = 8$
- The ciphertext c is 8

Decryption (7, 33)

Bob wants to decrypt the ciphertext:

- $c^d \; mod(n) = m \rightarrow 8^7 \bmod (33) = 2$
- The plaintext is 2

Encryption key (e, n)

- $m^e \; mod(n) = c$

Decryption key (d, n)

- $c^d \; mod(n) = m$

RSA algorithm

RSA algorithm as a three-part process:

Part I: Bob's public and private key setup

Part II: Alice encrypts m for Bob

Part III: Bob receives and decrypts c

RSA algorithm

Part I: Bob's public and private key setup

- Chooses two prime numbers, p and q
- Calculate the product $n = pq$

Example

- $p = 11$, $q = 3$
- $n = pq = 33$

n will eventually be published as part of the keys

- *Encryption key (e, n)*
- *Decryption key (d, n)*

However, p and q values remain secret.

RSA algorithm

Part I: Bob's public and private key setup

- Chooses two prime numbers, p and q
- Calculate the product $n = pq$
- Solve $\varphi(n) = (p-1)(q-1)$

Euler's totient function

Example

- $p = 11$, $q = 3$
- $n = pq = 33$
- $\varphi(n) = 10 \times 2 = 20$

Euler's totient function (also called Phi function), φ(n), counts the number of integers less than n that are coprime to n.

E.g., φ(6) from [1, 2, 3, 4, 5, 6]

φ(6) from [1, 5], φ(6) = 2 or $\varphi(2x3) = (p-1)(q-1) = 1x2 = 2$

RSA algorithm

Part I: Bob's public and private key setup

- Chooses two prime numbers, p and q
- Calculate the product $n = pq$
- Solve $\varphi(n) = (p-1)(q-1)$
- Choose numbers e and d so that ed has a remainder of 1 when divided by $\varphi(n)$
  - *$1 < e < \varphi(n)$, where e must be an integer*
  - *e and $\varphi(n)$ must be coprime*
  - $e*d \pmod{\varphi(n)} = 1$

Example

- $p = 11$, $q = 3$
- $n = pq = 33$
- $\varphi(n) = 10 \times 2 = 20$
- Pick e and d so that ed = 20+1

  e.g.,: $e = 3$, $d = 7$
  - 1 < 3 < 20
  - 3 and 20 are coprime

RSA algorithm

Part I: Bob's public and private key setup

- Chooses two prime numbers, p and q
- Calculate the product $n = pq$
- Solve $\varphi(n) = (p-1)(q-1)$
- Choose numbers e and d so that ed has a remainder of 1 when divided by $\varphi(n)$
  - *$1 < e < \varphi(n)$, where e must be an integer*
  - *e and $\varphi(n)$ must be coprime*
- Publish the public key (e, n)

Example

- $p = 11$, $q = 3$
- $n = pq = 33$
- $\varphi(n) = 10 \times 2 = 20$
- Pick e and d so that ed = 20+1

  e.g.,: $e = 3$, $d = 7$
  - 1 < 3 < 20
  - 3 and 20 are coprime
- Publish $(e, n) = (3, 33)$

RSA algorithm

Part II: Alice encrypts m for Bob

- Get Bob's public key *(e, n)*
- Encryption, calculate the ciphertext
  - $m^e \mod(n) = c$
- Send the ciphertext to Bob

Example

- $(e, n) = (3, 33)$
- Encrypting m = 2
  - $2^3 \bmod (33) = 8$
- Send c = 8 to Bob

Encryption key (e, n)

- $m^e \mod(n) = c$

RSA algorithm

Part III: Bob receives and decrypts c

- Get his private key *(d, n)*
- Decryption, calculate the plaintext
  - $c^d \, mod(n) = m$
- The plaintext should match what Alice sent

Example

- *(d, n)* = (7, 33)
- Decrypting c = 8
  - 8^7 mod (33) = 2
- m = 2, Alice's original message

Decryption key (d, n)

- $c^d \, mod(n) = m$

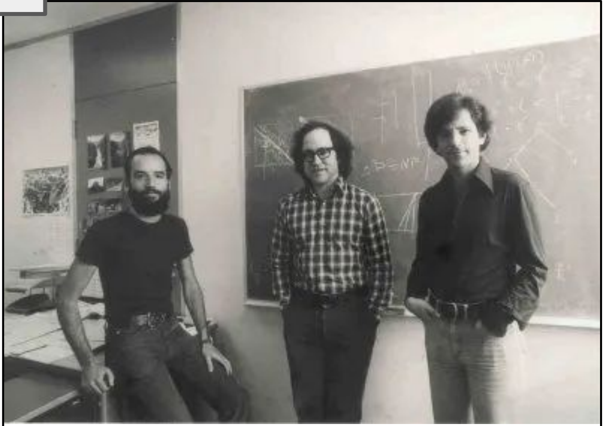
RSA algorithm

Rivest–Shamir–Adleman (RSA) algorithm is one of the oldest widely used for secure data transmission.

- Utilize private and public
  - Private key kept secret
  - Public key is available to anyone

What does this mean?

- Either one of the key can be public, while the other key can be private
- Based on the factorization of large prime numbers



From left to right: Adi Shamir, Ron Rivest, and Len Adleman

What happens if we swap the public and private keys?

Encryption ~~(3, 33)~~ (7, 33), Plaintext = “b” = 2

Alice wants to send the number 2:

- $m^e \bmod(n) = c$

Decryption ~~(7, 33)~~ (3, 33)

Bob wants to decrypt the ciphertext:

- $c^d \bmod(n) = m$

Public and private key swap

Encryption ~~(3, 33)~~ (7, 33), Plaintext = “b” = 2

Alice wants to send the number 2:

- $m^e \mathit{mod}(n) = c \rightarrow 2^7 \text{ mod } (33) = 29$
- The ciphertext c is 29

Decryption ~~(7, 33)~~ (3, 33)

Bob wants to decrypt the ciphertext:

- $c^d \mathit{mod}(n) = m \rightarrow 29^3 \text{ mod } (33) = 2$
- The ciphertext c is 2

… Still works!

RSA algorithm in practice

When our internet browser shows a URL beginning with https, the RSA algorithm is being used to protect our privacy.

For example, log in to Facebook:

- Our computer plays the role of Alice
- Facebook server plays the role of Bob, encrypting and decrypting the information passed back and forth.

In practice, the primes p and q are chosen to be very big numbers.

- Because the security of the RSA cryptosystem lies in the difficulty of factoring an integer that is the product of two large prime numbers.

Confidentiality

- Bob calculates *e, d, n* using *p* and *q*
  - Private: d, p and q
- Bob shares the information: encryption key (e, n)
  - Public: e, n
- Alice encrypts with (e, n), and shares the ciphertext
  - Public: ciphertext
- Bob decrypts with: decryption key (d, n), ciphertext
  - Public: n
  - Private: d

Confidentiality

To calculate the value of d, we need to use e:

$$\boxed{e} \xrightarrow{\varphi(n)} \boxed{d} \qquad \boxed{e*d \pmod{\varphi(n)} = 1}$$

To calculate φ(n), we need to use p and q:

$$\boxed{\varphi(n)} \longrightarrow \boxed{p,\ q} \qquad \boxed{\varphi(n) = (p-1)(q-1)}$$

To calculate p and q, we need to use n:

$$\boxed{n} \longrightarrow \boxed{p,\ q} \qquad \boxed{n = pq}$$

However, prime factorization is hard. We typically use 1024-bit or 2048-bit RSA keys. To factor the n value, we can only use a brute force solution.

Plaintext into numeric digits

Computers represent text as long numbers (01 for “A”, 02 for “B” and so on), so an email message is just a very big number.

Suppose we want to convert “hello world” into digits

- Convert “Hello world” into a sequence of bytes (hexadecimal)
    - “Hello world” -> “48 65 6C 6C 6F 20 57 6F 72 6C 64”
- Convert the sequence from hexadecimal to decimal
    - “48 65 6C 6C 6F 20 57 6F 72 6C 64” -> “87521618088882533792115812”
    - $(48\ 65\ 6C\ 6C\ 6F\ 20\ 57\ 6F\ 72\ 6C\ 64)_{16} = (4 \times 16^{21}) + (8 \times 16^{20}) + (6 \times 16^{19}) + (5 \times 16^{18}) + (6 \times 16^{17}) + (12 \times 16^{16}) + (6 \times 16^{15}) + (12 \times 16^{14}) + (6 \times 16^{13}) + (15 \times 16^{12}) + (2 \times 16^{11}) + (0 \times 16^{10}) + (5 \times 16^{9}) + (7 \times 16^{8}) + (6 \times 16^{7}) + (15 \times 16^{6}) + (7 \times 16^{5}) + (2 \times 16^{4}) + (6 \times 16^{3}) + (12 \times 16^{2}) + (6 \times 16^{1}) + (4 \times 16^{0}) = (87521618088882533792115812)_{10}$

U.S. Patent on RSA algorithm

U.S. Patent 4,405,829 on RSA public key encryption

United States Patent [19] [11] **4,405,829**

Rivest et al. [45] **Sep. 20, 1983**

[54] **CRYPTOGRAPHIC COMMUNICATIONS SYSTEM AND METHOD**

[75] Inventors: **Ronald L. Rivest**, Belmont; **Adi Shamir**, Cambridge; **Leonard M. Adleman**, Arlington, all of Mass.

[73] Assignee: **Massachusetts Institute of Technology**, Cambridge, Mass.

[21] Appl. No.: **860,586**

[22] Filed: **Dec. 14, 1977**

[51] **Int. Cl.**^{3} **H04K 1/00**; H04I 9/04

[52] **U.S. Cl.** **178/22.1**; 178/22.11

[58] **Field of Search** 178/22, 22.1, 22.11, 178/22.14, 22.15

Primary Examiner—Sal Cangialosi

Attorney, Agent, or Firm—Arthur A. Smith, Jr.; Robert J. Horn, Jr.

[57] **ABSTRACT**

A cryptographic communications system and method. The system includes a communications channel coupled to at least one terminal having an encoding device and to at least one terminal having a decoding device. A message-to-be-transferred is enciphered to ciphertext at the encoding terminal by first encoding the message as a number M in a predetermined set, and then raising that number to a first predetermined power (associated with the intended receiver) and finally computing the remainder, or residue, C, when the exponentiated number