CSCI471/971
Modern Cryptography

Public Key Cryptography I

Rupeng Yang

SCIT UOW

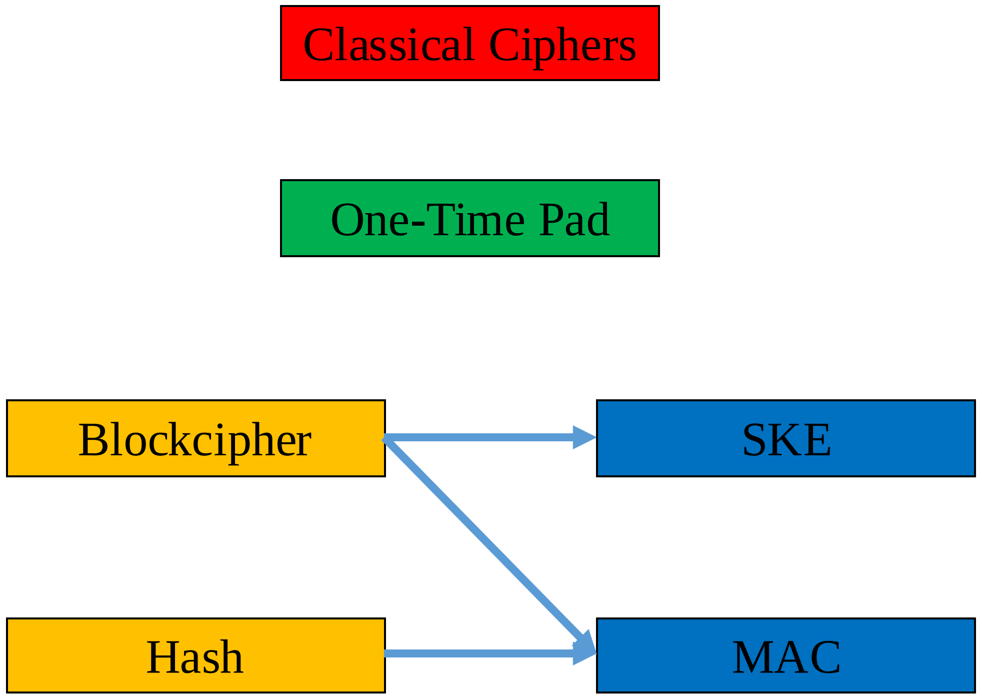
RoadMap

- Week 1-2: Preliminaries

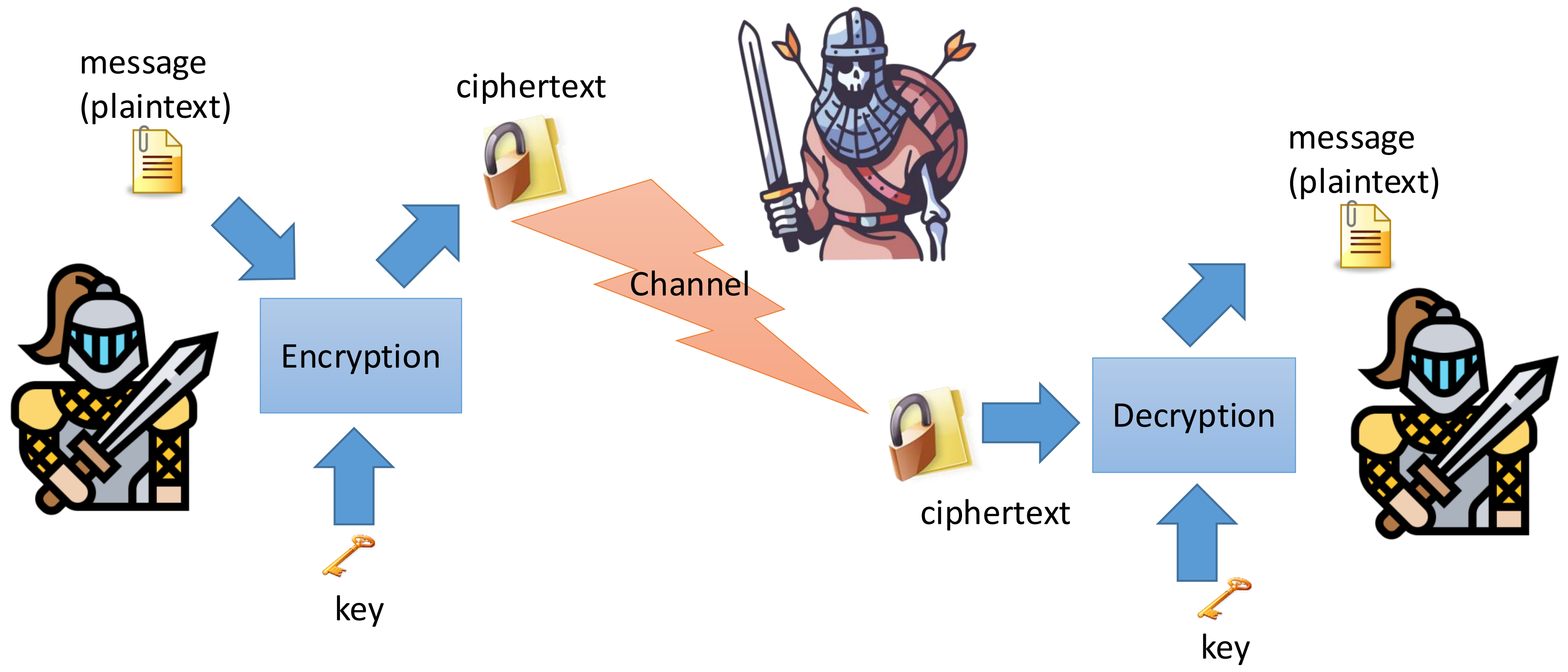
- Week 3-4: Symmetric-Key Cryptography
  - All parties share the same secret key
  - Symmetric-key encryption
  - Message authentication code

- From Week 5: Public-Key Cryptography

Roadmap

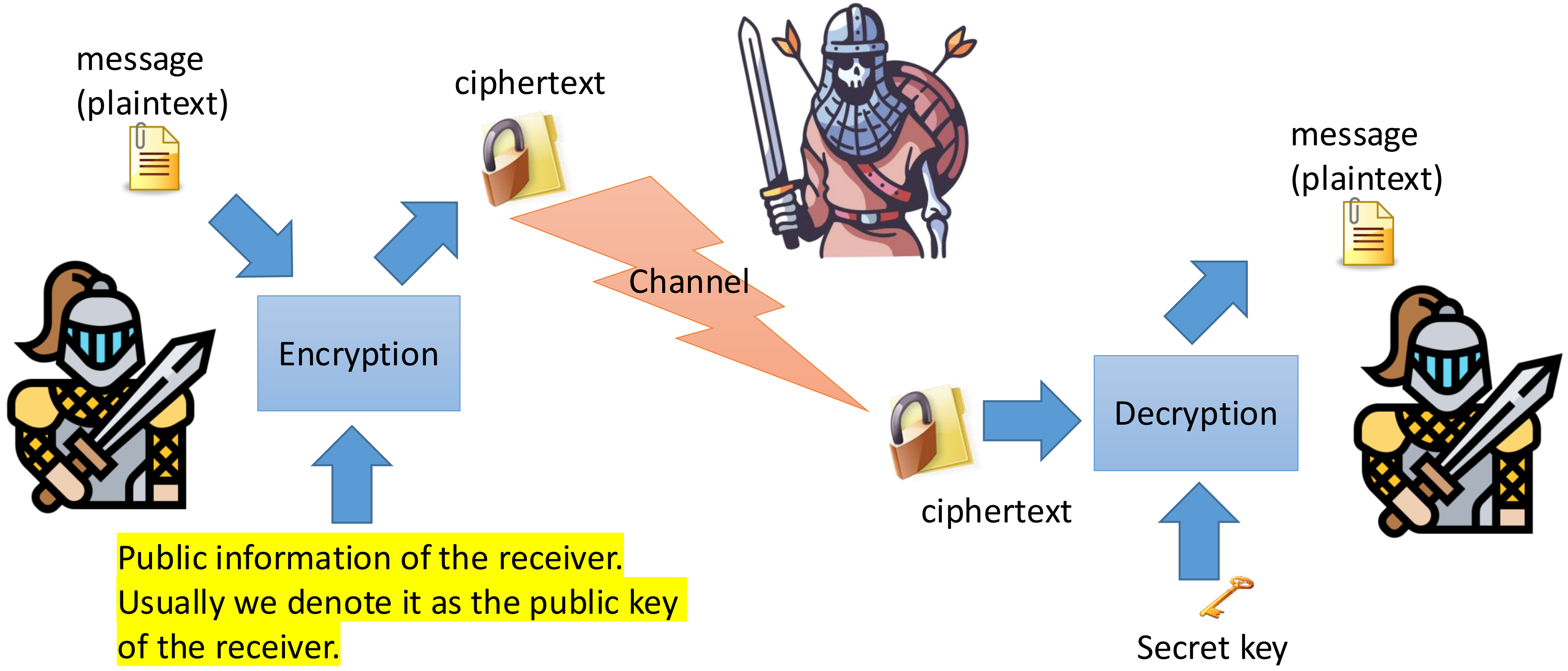


Symmetric-Key Encryption

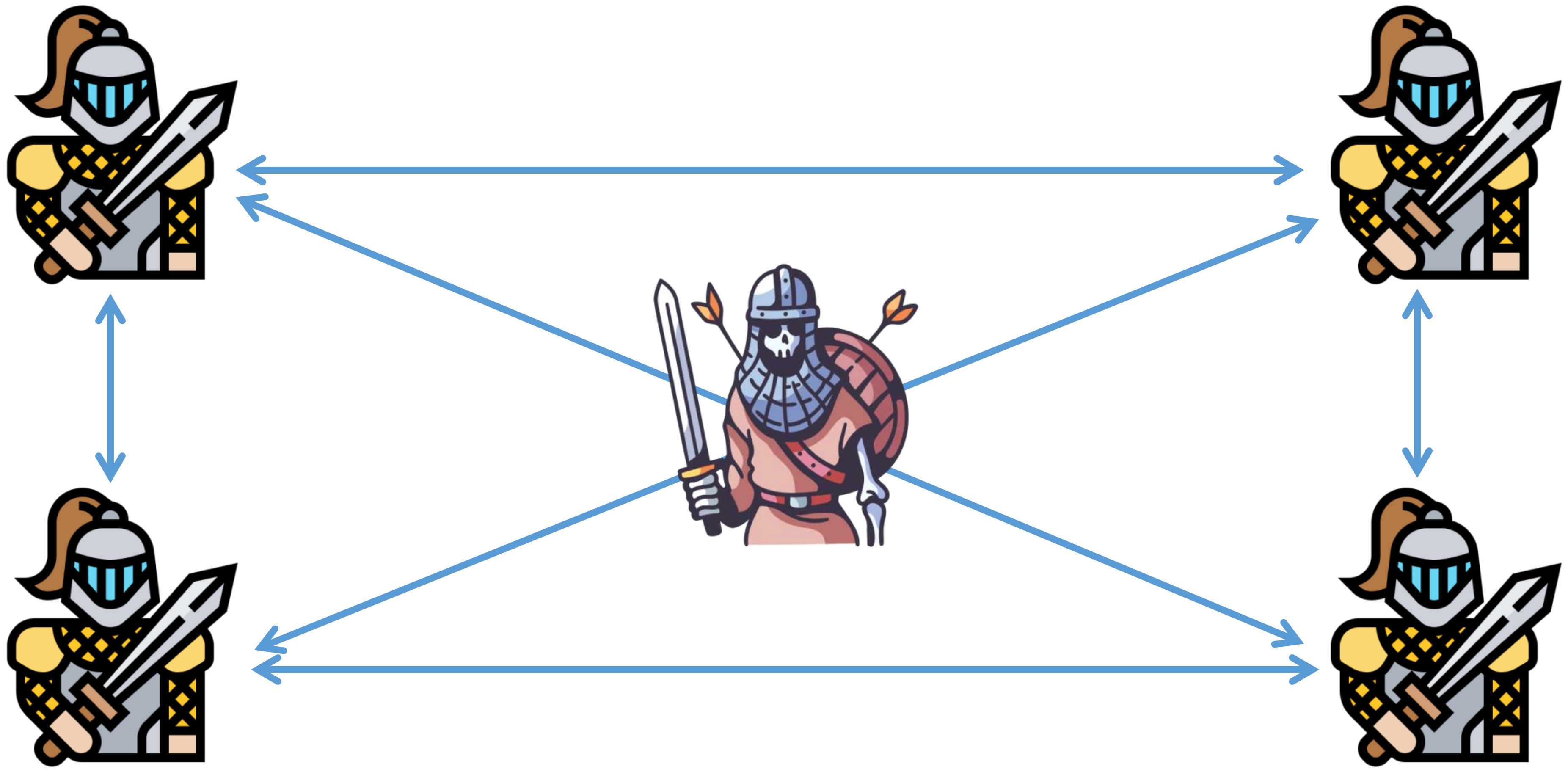


Does the key necessary?
For receiver: Yes!
For sender: Maybe not!

Public-Key Encryption



Public information of the receiver.
Usually we denote it as the public key
of the receiver.



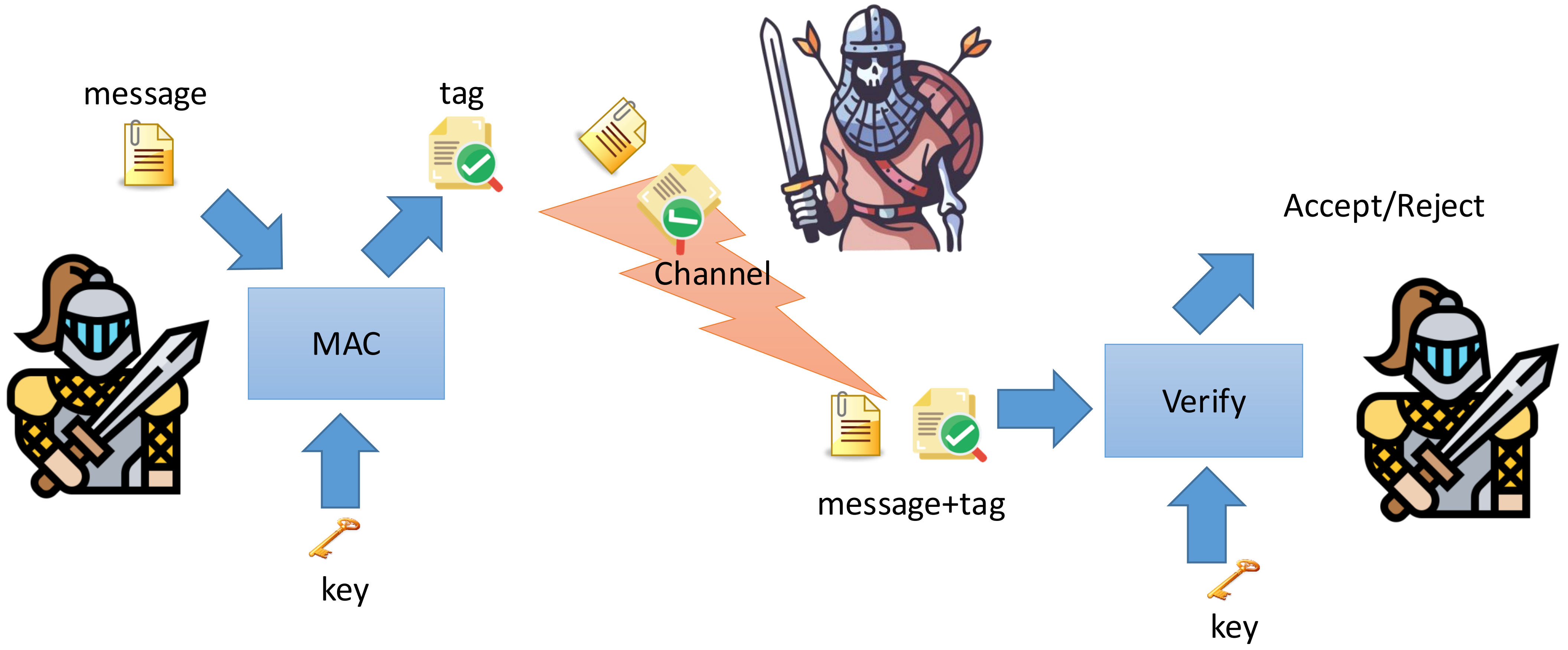
If there are 100 users and we hope to build a secure communication channel between each of them using SKE, how many secret keys should each user holds?

If there are 100 users and we hope to build a secure communication channel between each of them using PKE, how many secret keys should each user holds?

Advantage of Public-Key Encryption

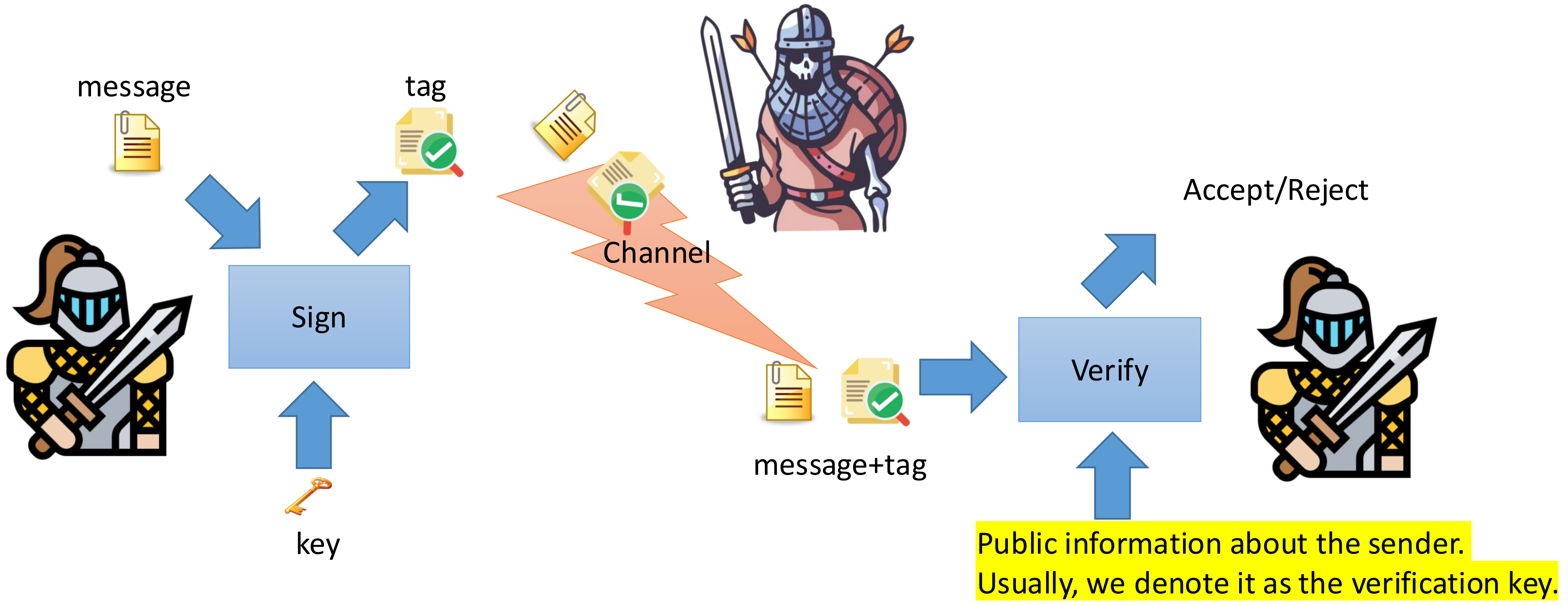
- It can significantly simplify key management.
  - For a network of N computer terminals, if we hope to establish secure communication between each pair of nodes using a secret-key encryption, we need to share a secret key between every two nodes. The total number of secret keys is N(N-1)/2 and two copies of each key exist in the network.
  - But if we use a public-key encryption, each party only needs to store its own secret key and publish its public key in a public bulletin board. So, there are only N public keys, which are stored in a public place, and N secret keys, each of which is stored in one terminal.
- It is much easier to publish a public key rather than sharing a secret key via a secure channel.

Message Authentication Code



Does the key necessary?
For sender: Yes!
For receiver: Maybe not!

Digital Signatures


message
tag
Sign
key
Channel
message+tag
Verify
Accept/Reject
Public information about the sender.
Usually, we denote it as the verification key.

The data that Alice sent to Bob cannot be modified by the adversary (even by Bob).
Bob only needs to know that pk belongs to Alice (no need to share a secret key!)

Advantage of Digital Signature

- It can significantly simplify key management.
- It is much easier to publish a public verification key rather than sharing a secret key via a secure channel.
- It ensures integrity of the message and **authenticity of the sender**. That is, if Bob gets a message which supposedly comes from Alice, he is able to check:
  - The message has not been modified.
  - The sender is truly Alice.
    - The authenticity holds even against the receiver.
    - This is similar to a handwritten signature.
- The signatures are transferrable and publicly verifiable
- The digital signature schemes provide non-repudiation

1976 Public-Key Cryptography

New Directions in Cryptography
W.Diffie & M.E.Hellman

Hellman

Diffie

“We stand today on the brink of a revolution in cryptography”

But How?

- Can we use blockcipher?
    - The encryption of blockcipher needs a secret key.
    - Actually, all oeprations of blockcipher needs a secret key.
    - It is proved that public-key encryption cannot be constructed from blociphers in a black-box manner.
- Can we use the techniques for constructing blockcipher (e.g., the diffusion-confusion principle)?
    - Probably no. The known techniques also relies on secret keys.
- New ideas are introduced to construct public-key encryption, which rely on number theory.

RSA

RSA

- The RSA Public Key Cryptosystem (**R**ivest, **S**hamir and **A**dleman (1978)) is based on the Integer Factorisation problem.

- It is *easy* to find primes and multiply them together to construct composite numbers, but it is *difficult* to factor a composite number. Compute p and q from N=pq

RON RIVEST, ADI SHAMIR & LEN ADLEMAN
2002
acm
RSA public-key cryptography
A.M. TURING

Preliminaries on RSA

- It is easy to generate large random prime numbers
  - First, you choose a large number P (say, a 1024-bit random number)
  - Then you test if the number is prime using the **Miller–Rabin primality test algorithm**.
  - You get a large random prime number if P is prime; otherwise, you repeat this procedure

- It is hard to factorize N=pq if p and q are large enough primes.

Preliminaries on RSA

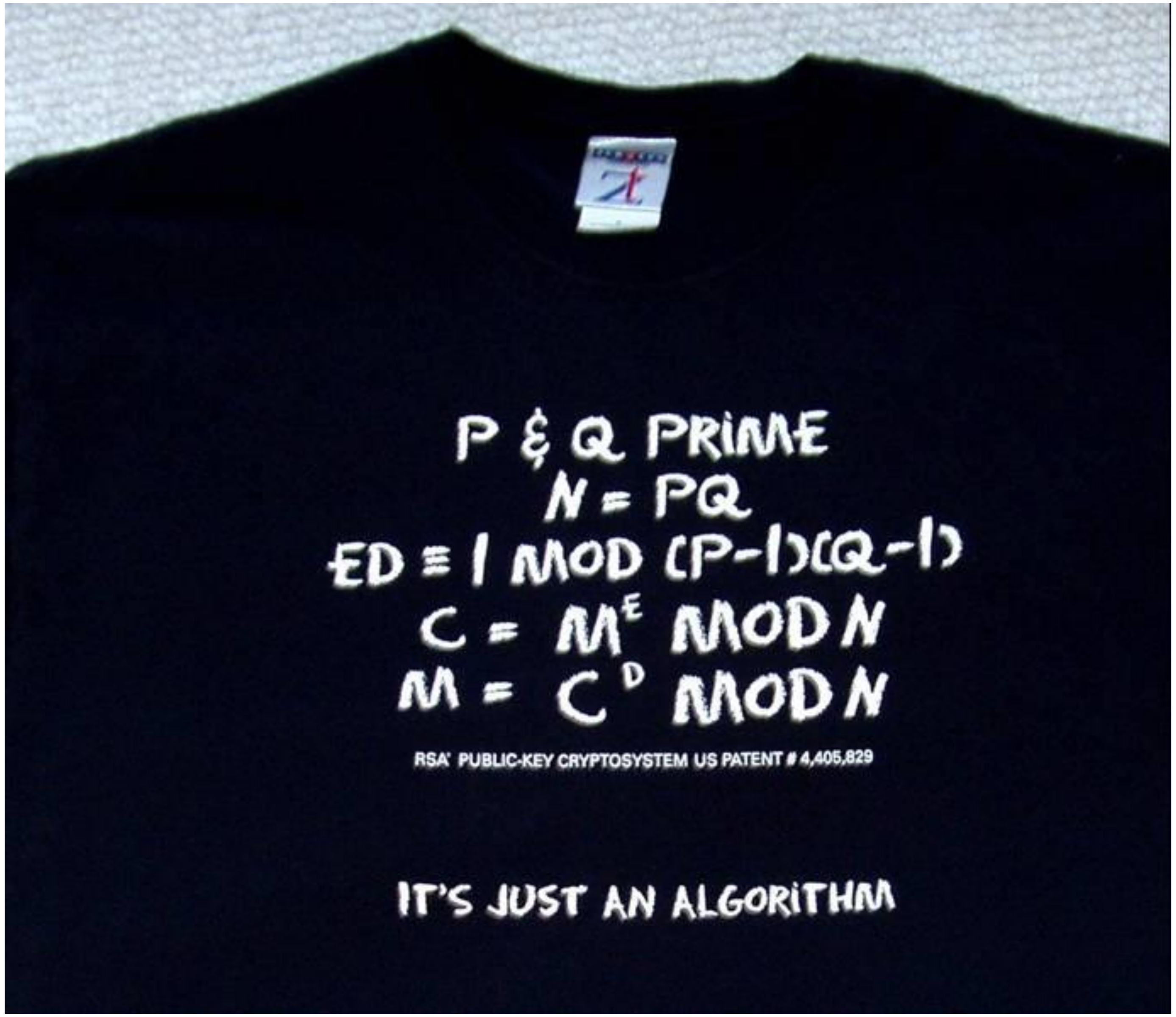
- Given any number N = pq, we define ϕ(N)= (p-1)(q-1).
  - This is is Euler’s totient function: It is the number of positive integers less than N that are relatively prime to N.
  - For any a that is relatively prime with N, we have

$$a^{\phi(N)} = 1 \, mod \; N$$

Preliminaries on RSA

- It is easy to compute x^e given x and e, using the **fast exponentiation algorithm.**
- It is easy to compute y s.t. $xy = 1\ mod\ N$ given x and N that are relatively prime
  - That means xy – 1 is a multiple of N
  - You can use the **Extended Euclidean algorithm** to compute y.

RSA Encryption


P & Q PRIME
$N = PQ$
$ED \equiv 1 \bmod (P-1)(Q-1)$
$C = M^E \bmod N$
$M = C^D \bmod N$
RSA' PUBLIC-KEY CRYPTOSYSTEM US PATENT # 4,405,829
IT'S JUST AN ALGORITHM

The Textbook RSA

1. Choose two large primes p and q. Compute n = pq and m=ϕ(n)= (p-1)(q-1).
   - We can sample random large primes by first sampling random large numbers and then perform the primality test.
   - ϕ(n) is Euler's totient function: It is the number of positive integers less than n that are relatively prime to n.
   - For any a that is relatively prime with N, we have
     $$a^{\phi(N)} = 1 \; mod \; N$$
2. Choose a random e, such that $1 \leq e \leq m - 1$ and gcd(e,m)=1.
3. Finds d such that ed=1 mod m.
   - This is possible because of the choice of e.
   - d is the multiplicative inverse of e modulo m and can be found using the extended Euclidean (gcd) algorithm.
4. The **Public key** is **(e, n)**.

   The **Secret key** is **(d, n)**.

The Textbook RSA

- **ENC:** Given a public key (e,n), to encrypt a message **X**, the ciphertext **Y** is

$$Y = X^e \bmod n$$

- **DEC:** Given the secret key (d,n), to decrypt **Y**, the decryption algorithm calculates

$$X = Y^d \bmod n$$

- We can use the fast exponentiation algorithm to perform both steps efficiently.

- Correctness of RSA:

$$Y^d \bmod n = (X^e)^d \bmod n = X^{ed} \bmod n = X^{km+1} \bmod n = X \bmod n$$

Assessing the security of RSA

- The security of the private component of a key depends on the difficulty of factoring large integers.
  - Let Z=(e,n). If n can be factorised then an attacker can find
    - p and q s.t. n=pq
    - m=(p-1)(q-1)
  - … and use the gcd algorithm to find the private key, $d=e^{-1}$ mod m.
- No efficient algorithm for factoring is known, so knowing n = pq does not yield p or q. This is the necessary condition for the security of RSA.

Assessing the security of RSA

- The security of the private component of a key depends on the difficulty of factoring large integers.
- No efficient algorithm for factoring is known, so knowing n = pq does not yield p or q. This is the necessary condition for the security of RSA.
- Does this imply that m=(p-1)(q-1) cannot be found?
  - Yes. You can make a quadratic equation.

Assessing the security of RSA

- The security of the private component of a key depends on the difficulty of factoring large integers.
- No efficient algorithm for factoring is known, so knowing n = pq does not yield p or q. This is the necessary condition for the security of RSA.
- This implies that m=(p-1)(q-1) cannot be found.
- Does this imply that d cannot be computed?
    - Yes! But the proof is more complicated.

Assessing the security of RSA

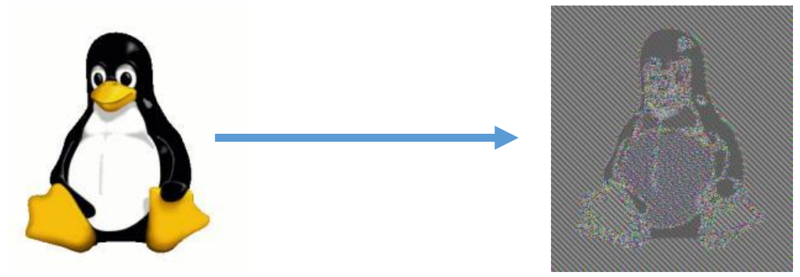
- The security of the private component of a key depends on the difficulty of factoring large integers.
- No efficient algorithm for factoring is known, so knowing n = pq does not yield p or q. This is the necessary condition for the security of RSA.
- This implies that m=(p-1)(q-1) cannot be found.
- This implies that d cannot be computed.
- Does this imply that the message cannot be recovered from the ciphertext?
  - This is still an open problem!
  - But we assume that it is hard to recover the message from the ciphertext using the textbook RSA encryption. This is called the RSA assumption.

The parameters of schemes based on RSA

- The modulus N should be large enough to guarantee the security
  - 80-bit security: N is a 1024-bit number
  - 112-bit security: N is a 2048-bit number
  - 128-bit security: N is a 3072-bit number
- This is due to the best known attack
  - Quadratic Sieve: $2^{O((\log N)^{1/2}(\log\log N)^{1/2})}$
  - General Number Field Sieve: $2^{O((\log N)^{1/3}(\log\log N)^{2/3})}$
  - Shor's Algorithm:$\mathrm{O}(\,(\log N)^2(\log\log N)(\log\log\log N))$
    - But it needs quantum computers!

Security of RSA Encryption

- Is the textbook RSA encryption Secure?
  - They are deterministic!



  - Partial information about the message might be learned even if you only encrypt random message.

RSA Signature

RSA Signature

- Key Generation:
  - Generate primes P and Q, compute N = PQ
  - Generate d and e such that de = 1 mod (P-1)(Q-1)
  - Public Key (N, e)
  - Private Key (N,d)
- Sign:
  - Given message m, compute $s = m^d \bmod N$
- Verify:
  - Given message m, signature s, check if $m = s^e \bmod N$
- Correctness

Security of RSA Signature

- Public key (N,e)
- The attacker picks a random value s.
    - Compute m = s^e mod N
- Carol outputs (m, s) as a message-signature pair!

Security of RSA Signature

- Public key (N,e)
- The attacker now hopes to generate a signature for a message m.
- It picks two values A and B s.t. AB=m mod N
  - It can pick a random A and compute B=A^{-1}m mod N
- Then it gets the signatures S_A, S_B for A and B respectively.
  - Recall that we should assume that the attacker could choose messages and get their tags/signatures when considering protecting the integrity.
- Finally, it outputs S_A S_B as the signature for m.

Security of RSA Signature

- Textbook RSA signature is not secure
  - A non-message attack to generate a valid message/signature pair
  - A chosen-message attack to forge signatures on any messages.

RSA and One-Way Trapdoor Functions

One-Way Trapdoor Function

(Definition) A function f:{0,1}*→{0,1}* is a one-way trapdoor function if

- Easy to Compute: There exists a P.P.T algorithm that can compute f(x) for any x.
- Hard to Invert: For every P.P.T. adversary, given f(x), where x is sampled uniformly at random, we have

$$Pr[f(\ A(f(x))\ ) = f(x)] \leq negl$$

- Easy to Invert with Trapdoor: There exists a trapdoor td and a P.P.T algorithm that given td and f(x), it is easy to compute x.
- We always consider Bijective functions for one-way trapdoor functions.

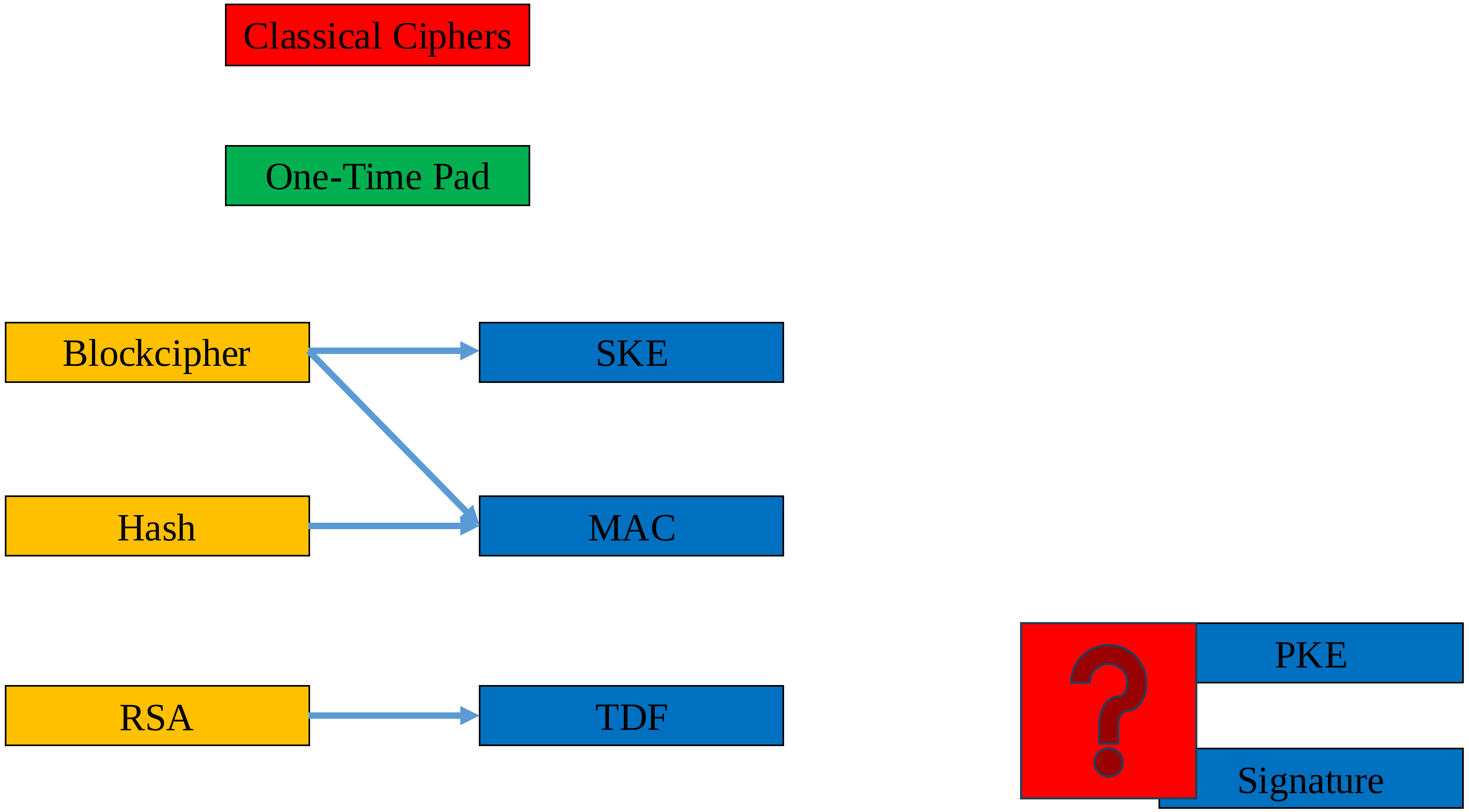
RSA as a Trapdoor One-Way Function

- Generating the function and the trapdoor
  1. Choose two large primes p and q. Compute $n = pq$ and $m=\phi(n)= (p-1)(q-1)$.
  2. Choose a random e, such that $1 \le e \le m - 1$ and $\gcd(e,m)=1$.
  3. Finds d such that $ed=1 \bmod m$.
  4. The **function** is described by **(e, n)**.

  The **trapdoor** is **(d, n)**.
- Computing the function given an input X: $Y = X^e \bmod n$.
- Inverting the function on Y: $X = Y^d \bmod n$.

- The trapdoor function is assumed to be a one-way trapdoor function (The RSA assumption).
  - The assumption is related to the factoring problem

Roadmap


Classical Ciphers
One-Time Pad
Blockcipher
SKE
Hash
MAC
RSA
TDF
PKE
Signature

Summary

- Public-Key Cryptography
  - PKE
  - Signature
  - Advantages
- RSA Preliminaries
- Textbook RSA Encryption
  - Construction
  - Correctness
  - Security
  - Insecurity
- Textbook RSA Signature
  - Construction
  - Correctness
  - Security
  - Insecurity
- Trapdoor Function
  - Notion and Definition
  - RSA TDF