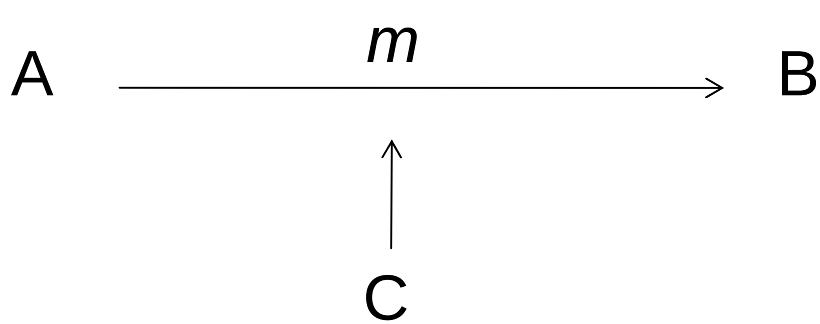
Cryptographic Security

Presenter: Hamid Al-Hamadi

Security Goals

- Consider the following security risks that could face two communicating entities in an unprotected environment:

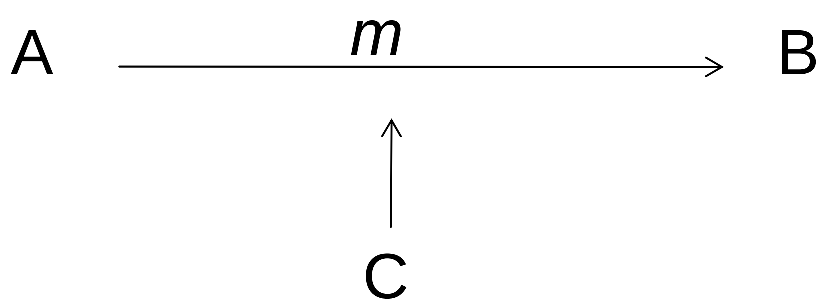
(1)



•C could view the secret message by eavesdropping on the communication.

Loss of privacy/confidentiality

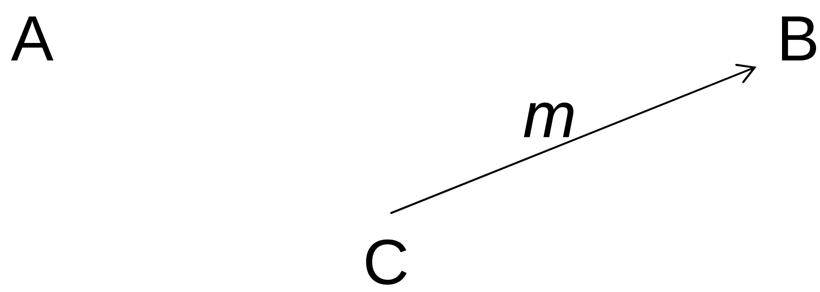
(2)



C could alter/corrupt the message, or the message could change while in transit. If B does not detect this, then we have Loss of Integrity

(3)

Or it could send a massage to B pretending to be A



If B cannot verify the source entity of the information then we lack authentication

(4) A $\xrightarrow{m}$ B

A might repudiate having sent m to B

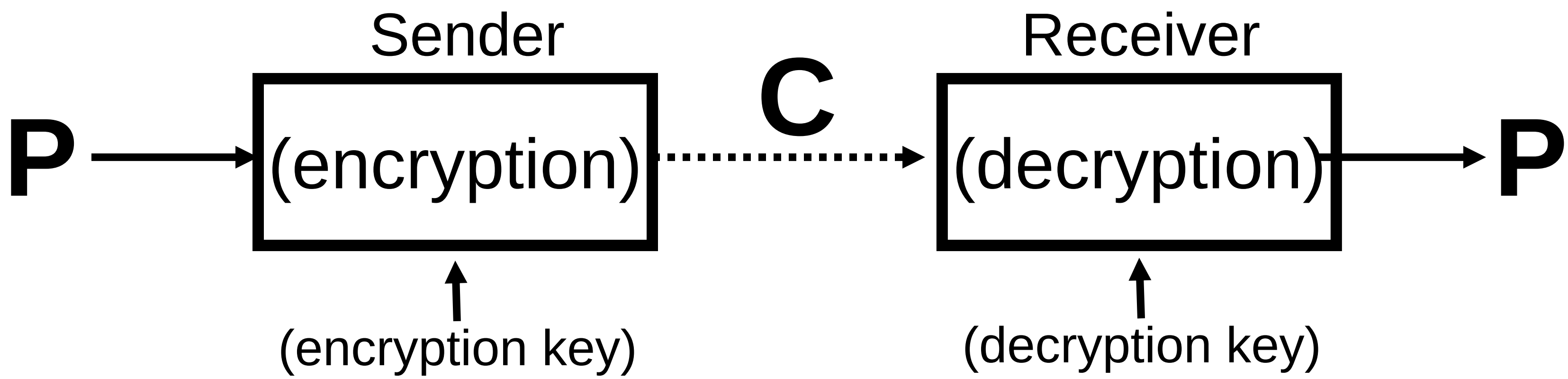
Hence, some possible goals for communication:

- **Privacy/confidentiality - information not disclosed to unauthorized entities**
- **Integrity - information not altered deliberately or accidentally**
- **Authentication - validation of identity of source of information**
- **Non-repudiation – Sender should not be able to deny sending a message**

What is Cryptography

- Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, authentication, and non-repudiation.

What is a cryptographic system composed of?

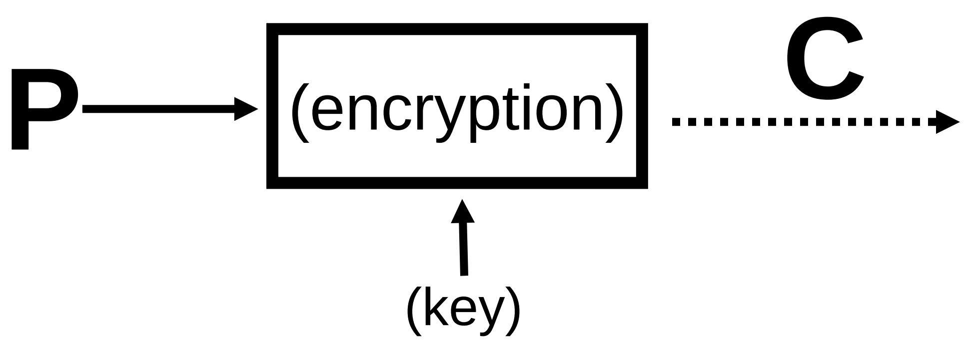


- **Plaintext**: original message or data (also called cleartext)
- **Encryption**: transforming the plaintext, under the control of the key
- **Ciphertext**: encrypted plaintext
- **Decryption**: transforming the ciphertext back to the original plaintext
- **Cryptographic key**: used with an algorithm to determine the transformation from plaintext to ciphertext, and v.v.

Attack classification

Ciphertext Alone attack: The attacker has available only the intercepted cryptogram C.

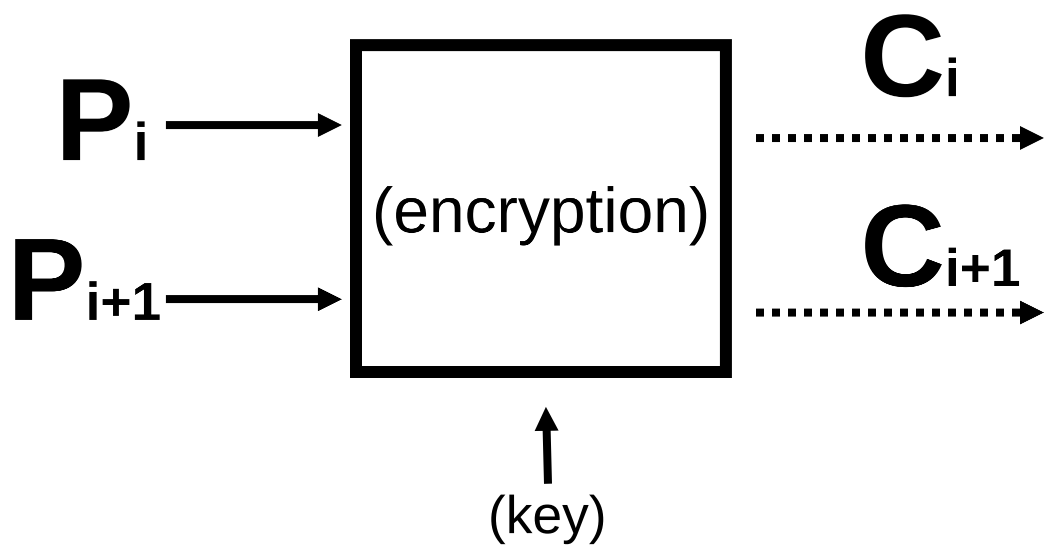
From C , try to find P or (even better) the key.



Attack classification

Known Plaintext attack: The attacker knows a small amount of plaintext (P_i) and its ciphertext Equivalent (C_i).

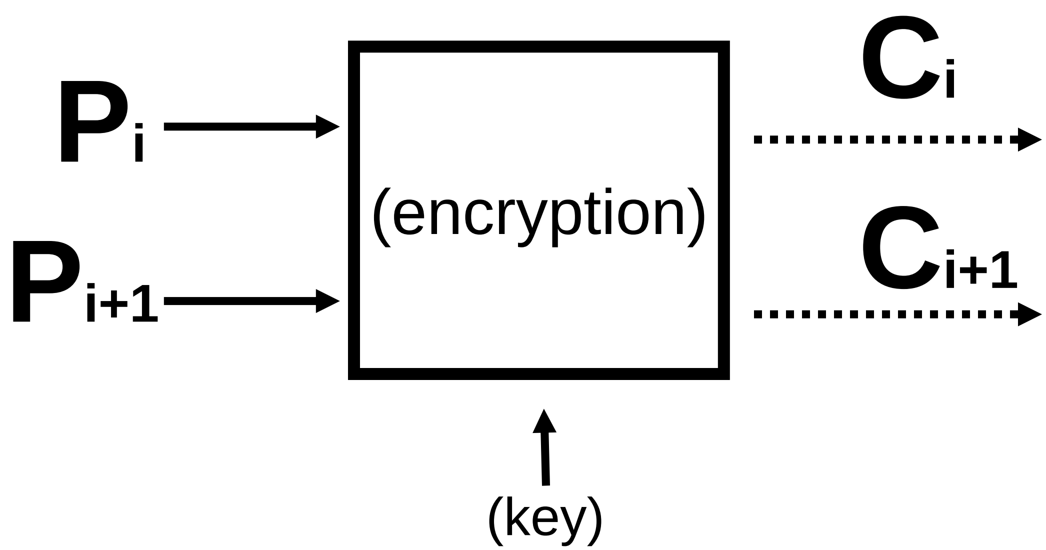
Attacker tries to find key or to infer P_{i+1} (next plaintext)



Attack classification

Chosen Plaintext attack: The attacker can choose plaintext (P_i) and obtain its ciphertext (C_i).

A careful selection of (P_i) would give a pair of (P_i, C_i) good for analyzing Enc. Alg. + key and in finding Pi+1 (next plaintext of sender)

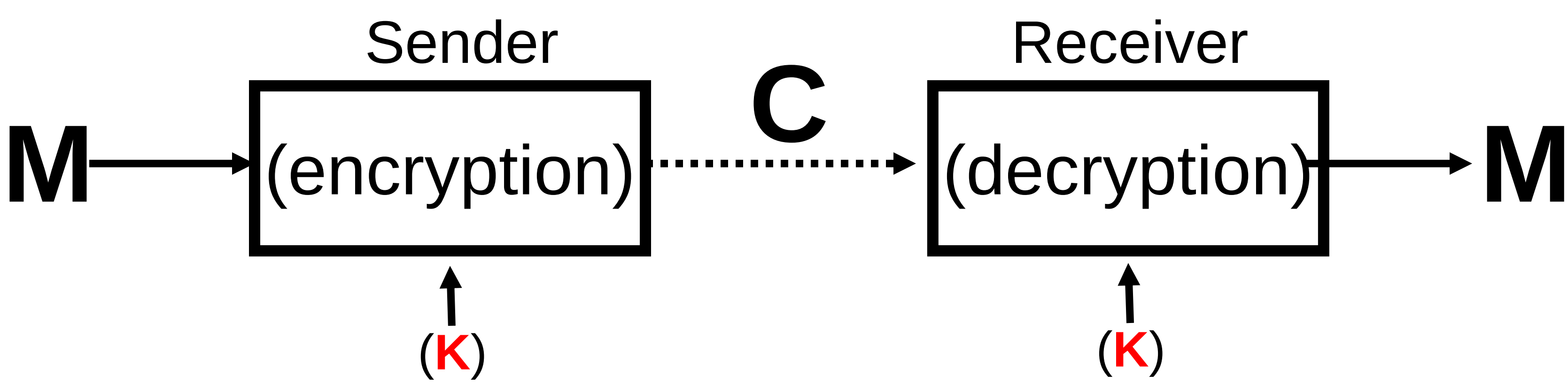


Forms of Cryptosystems

- **Private Key (symmetric) :**

  A single key (**K**) is used for both encryption and decryption and must be kept secret.

  Key distribution problem - a secure channel is needed to transmit the key before secure communication can take place over an unsecure channel.



$E_K(M) = C$ $\qquad$ $D_K(C) = M$

Forms of Cryptosystems

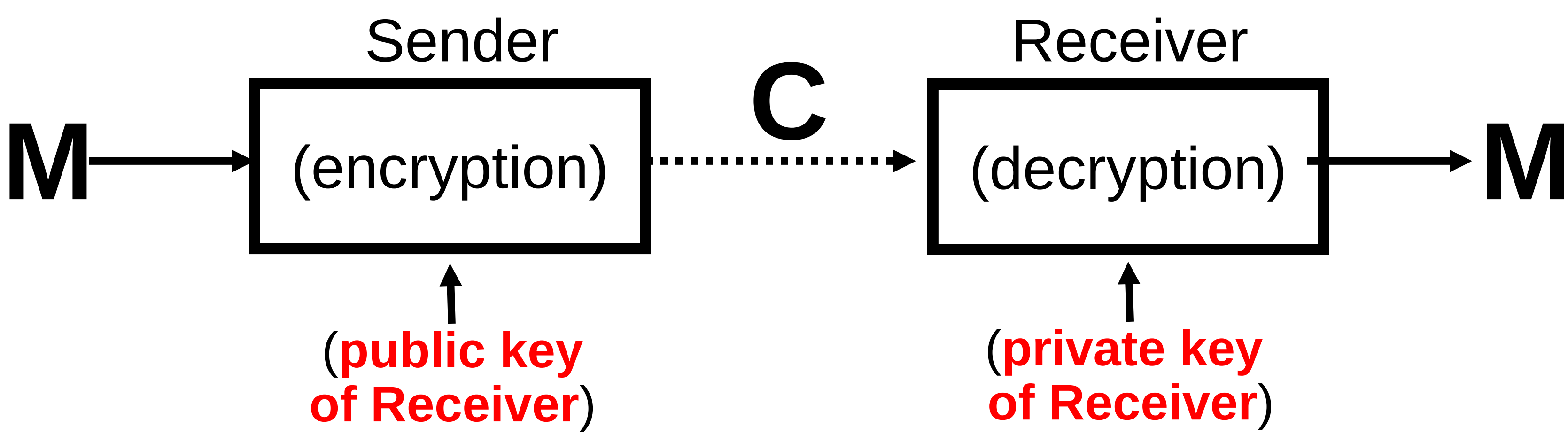
- **Public Key (asymmetric):**
  - The encryption procedure (key) is public while the decryption procedure (key) is private.
  - Each participant has a public key and a private key.
  - May allow for both encryption of messages and creation of digital signatures.

Forms of Cryptosystems

- **Public Key (asymmetric):**

  Requirements:

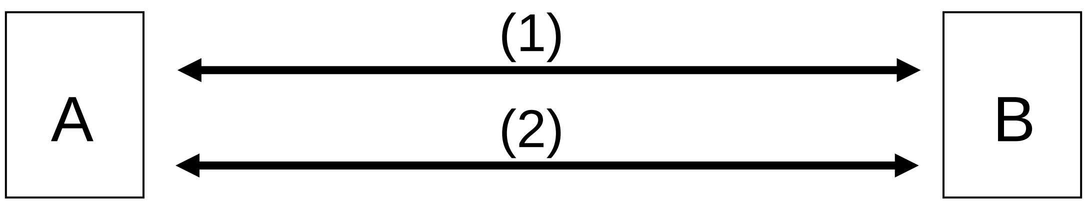
  1. For every message M, encrypting with public key and then decrypting resulting ciphertext with matching private key results in M.
  2. Encryption and Decryption can be efficiently applied to M
  3. It is impractical to derive decryption key from encryption key.



Combining Public/Private Key Systems

Public key encryption is more expensive than symmetric key encryption

For efficiency, combine the two approaches



(1) Use public key encryption for authentication; once authenticated, transfer a shared secret symmetric key

(2) Use symmetric key for encrypting subsequent data transmissions

Rivest-Shamir-Adelman (RSA) Method

- Named after the designers: Rivest, Shamir, and Adleman
- Public-key cryptosystem and digital signature scheme.
- Based on difficulty of factoring large integers
  - For large primes p & q, n = pq
  - Public key *e* and private key *d* calculated

RSA Key Generation

Every participant must generate a Public and Private key:

1. Let p and q be large prime numbers, randomly chosen from the set of all large prime numbers.
2. Compute n = pq.
3. Choose any large integer, d, so that:

   $GCD(d, \varphi(n)) = 1$ (where $\varphi(n) = (p-1)(q-1)$)
4. Compute $e = d^{-1} \pmod{\varphi(n)}$.
5. Publish n and e. Keep p, q and d secret.

Note:

- Step 4 can be written as:

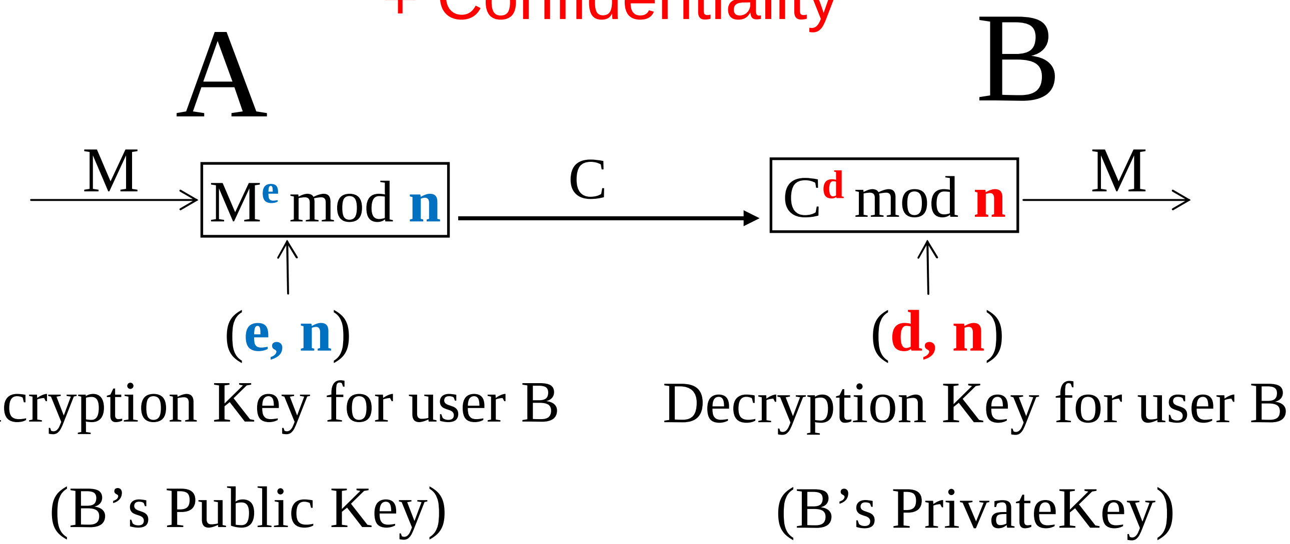
  Find e so that: $e \times d = 1 \text{ (modulo } \varphi(n))$
- If we can obtain p and q, and we have (n, e), we can find d

Rivest-Shamir-Adelman (RSA) Method

Assume A wants to send something confidentially to B:

- A takes M, computes $C = M^e \bmod n$, where (e, n) is B’s public key. Sends C to B
- B takes C, finds $M = C^d \bmod n$, where (d, n) is B’s private key

+ Confidentiality



RSA Method

Example:

1. p = 5, q = 11 and n = 55.
   (p-1)x(q-1) = 4 x 10 = 40

2. A valid d is 23 since GCD(40, 23) = 1

3. Then e = 7 since:
   23 x 7 = 161 modulo 40 = 1

in other words

$$e = 23^{-1} \pmod{40} = 7$$

Digital Signatures Based on RSA

In RSA algorithm the encryption and decryption operations are commutative:

$(m^e)^d = (m^d)^e = m$

We can use this property to create a digital signature with RSA.

Digital Signatures (Public Key)

Public Key System:

sender, A: (E_A : public, D_A : private)
receiver, B: (E_B : public, D_B : private)

A signs the message m using its private key,
the result is then encrypted with B's public key, and the resulting ciphertext is sent to B:
$C= E_B (D_A (M))$

B receives ciphertext C decrypts it using its private key
The result is then encrypted with the senders public key (A's public key) and the message m is retreived

$M = E_A (D_B (C))$

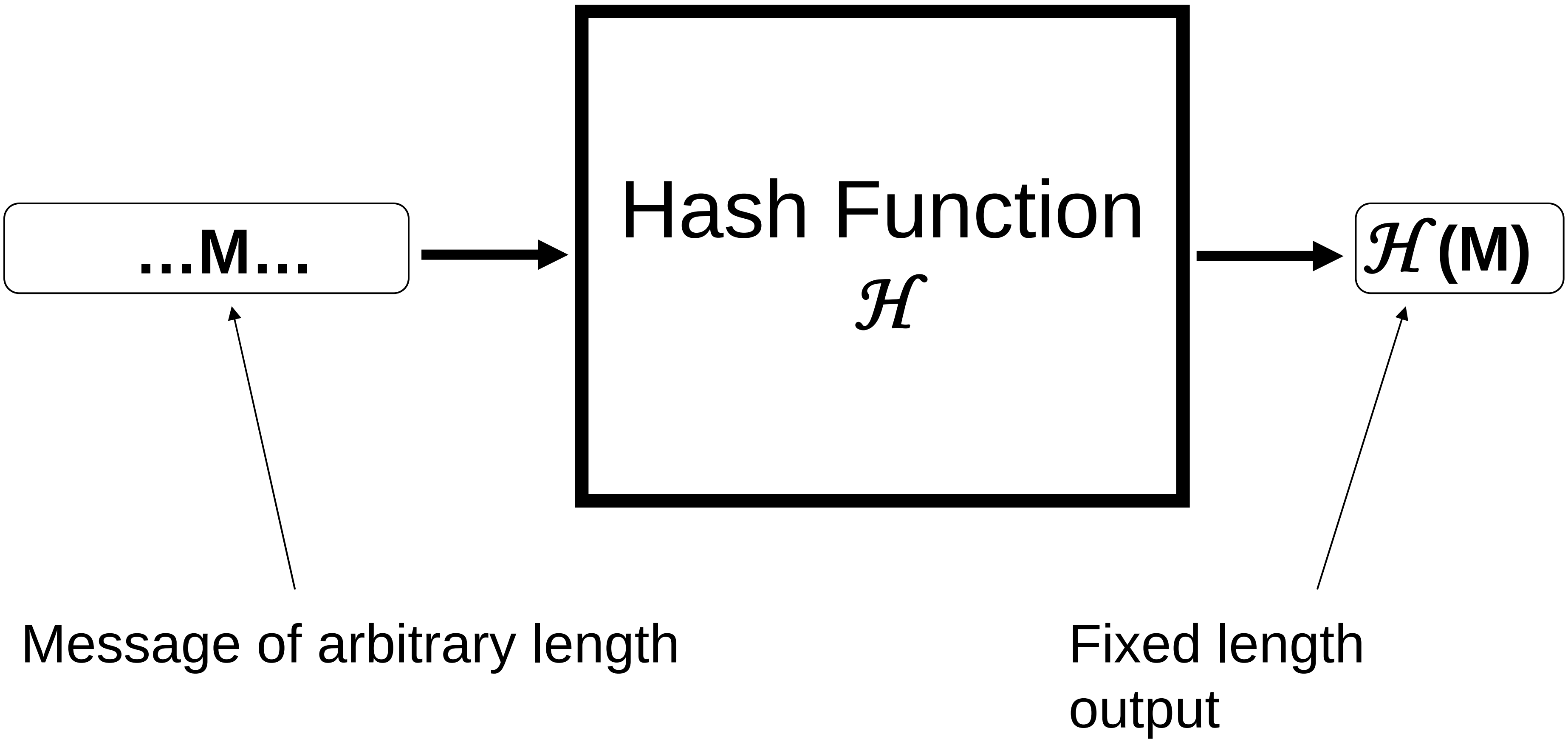
Hashing

A one-way hash function h is a public function h (which should be simple and fast to compute) that satisfies three properties:

1. A message m of arbitrary length must be able to be converted into a message digest h(m) of fixed length.
2. It must be one-way, that is given y = h(m) it must be computationally infeasible to find m.
3. It must be collision free, that is it should be computationally infeasible to find m1 and m2 such that h(m1) = h(m2).

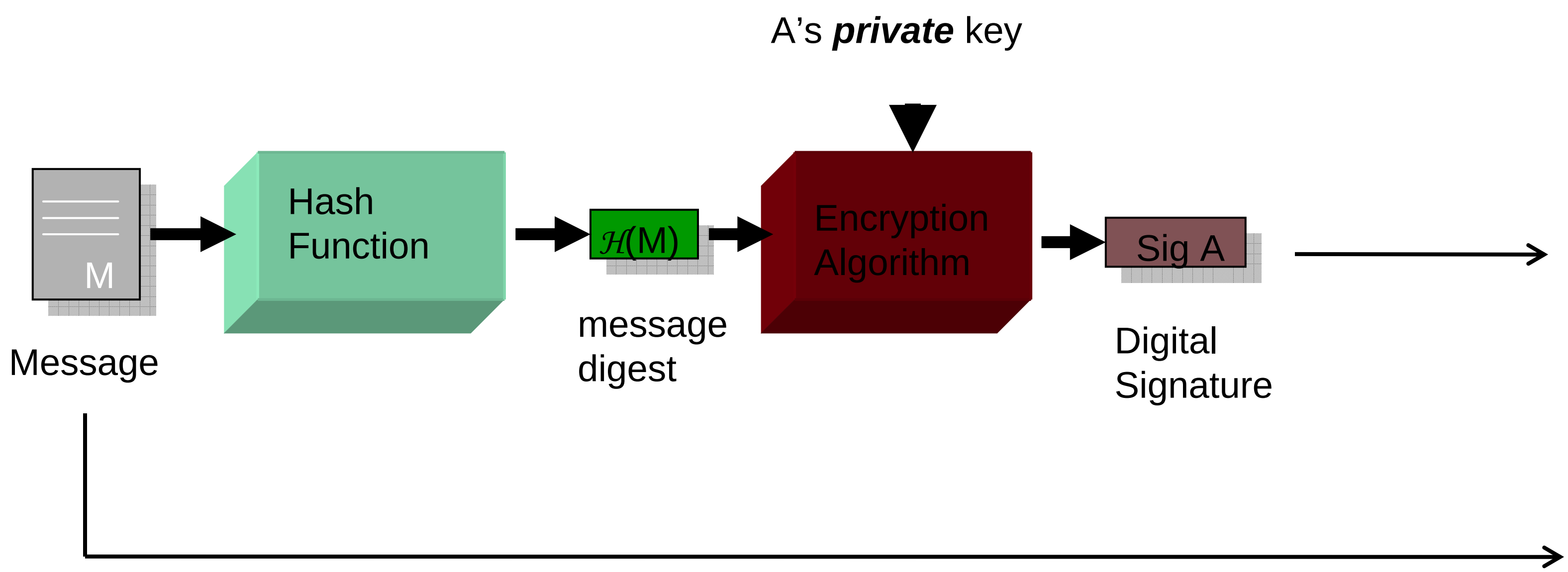
Examples: MD5 , SHA-1

Hash Function



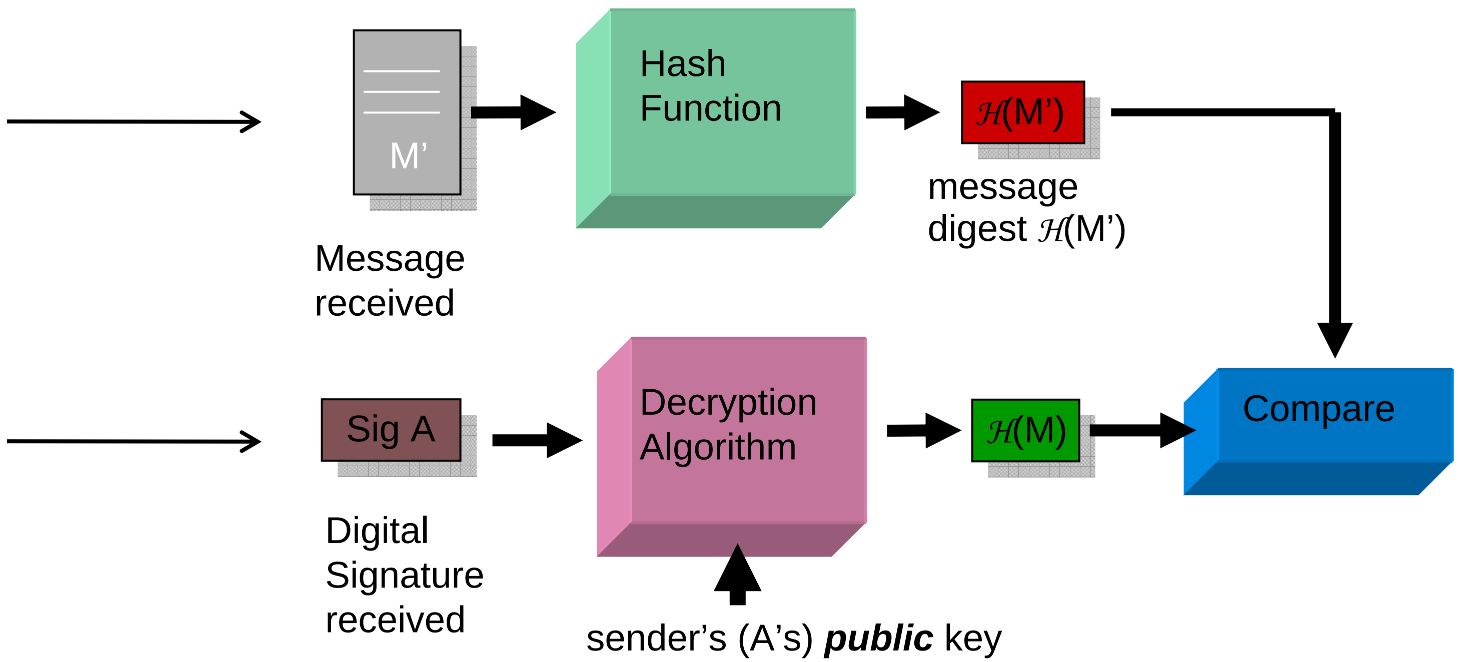
Producing Digital Signatures

Step 1: A produces a one-way hash of the message.
Step 2: A encrypts the hash value with its private key, forming the signature.
Step 3: A sends the message and the signature to B.



Verifying Digital Signature

Step 4: B forms a one-way hash of the message.
Step 5: B uses A’s public key to decrypt the signature and obtain the sent hash.
Step 6: compare the computed and sent hashes



Security of Digital Signatures

If the hashes match then we have guaranteed the following:

- Integrity: if m changed then the hashes would be different
- Authenticity & Non-repudiation: A is who sent the hash, as we used A's public key to reveal the contents of the signature A cannot deny signing this, nobody else has the private key.

Satisfies the requirements of a Digital Signature

If we wanted to further add confidentiality, then we would encrypt the sent m + signature such that only B could reveal the contents (encrypt with B's public key)

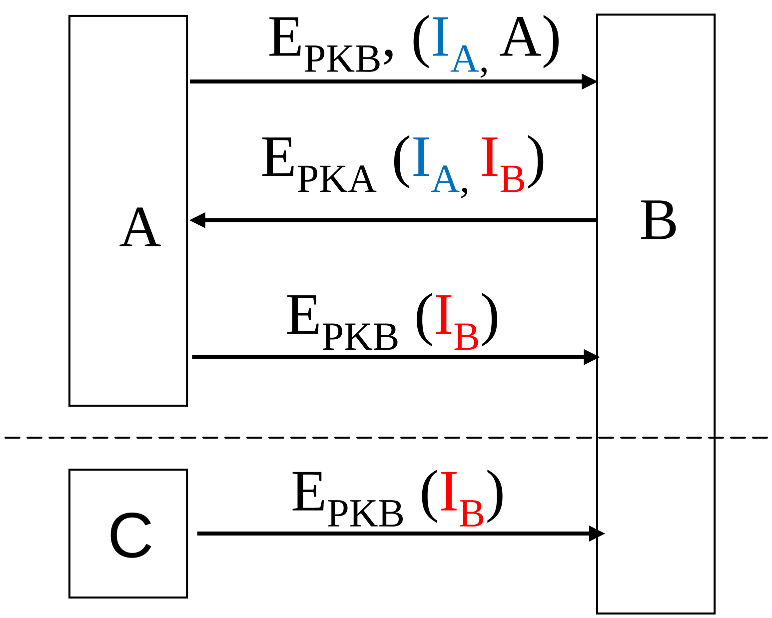
Possible problem: If signing modulus > encrypting modulus

-> Reblocking Problem

Secure Communication (Public Key)

Handshaking

I_A, I_B are “nonces”
nonces can be included in each subsequent message
PKB: public key of B; PKA: public key of A;



← If B sees the same nonce at a later time, then it should suspect a replay attack.

Questions?