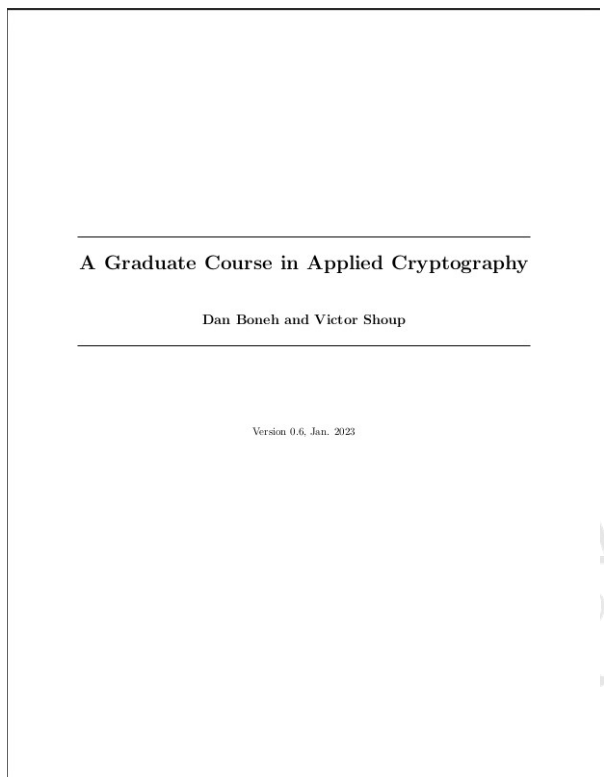
Cryptography I

DTU Compute
Department of Applied Mathematics and Computer Science

Books recommendation

- A Graduate Course in Applied Cryptography by Dan Boneh   and Victor Shoup: https://toc.cryptobook.us/



A Graduate Course in Applied Cryptography

Dan Boneh and Victor Shoup

Version 0.6, Jan. 2023

What Cryptography can do

- Cryptography is just one of the tools in the security tool box
  - But a very versatile and powerful tool

- Cryptography can help solve important problems:
  - Keeping secrets
    - *Messages on the network*
    - *Data stored on disks, memory cards, USB sticks, …*
  - Ensuring integrity
    - *Messages on the network (Message Authentication Codes)*
    - *Data stored on disk, …*
  - Authentication
    - *User authentication*
      - Protecting shared secrets during communication
    - *Message authentication*
      - Sender authentication
      - Message authentication
        combine sender authentication and message integrity

Keeping Secrets
Steganography, Codes and Ciphers

- Steganography
  - Hide the existence of a secret message
  - Inconspicuous message (invisible ink, micro dots)
  - If the message is found, then the secret is revealed

- Codes
  - Replace symbols in the message with “codes”
  - Transformation must be agreed in advance
  - Inconspicuous if code is well defined

- Ciphers
  - Hide the meaning of the secret message
  - Scrambling according to an agreed algorithm
  - Conspicuous message (obviously a secret)

Cryptography
Fundamental principles

Secret writing

Steganography (hidden)

Cryptography (scrambled)

Substitution

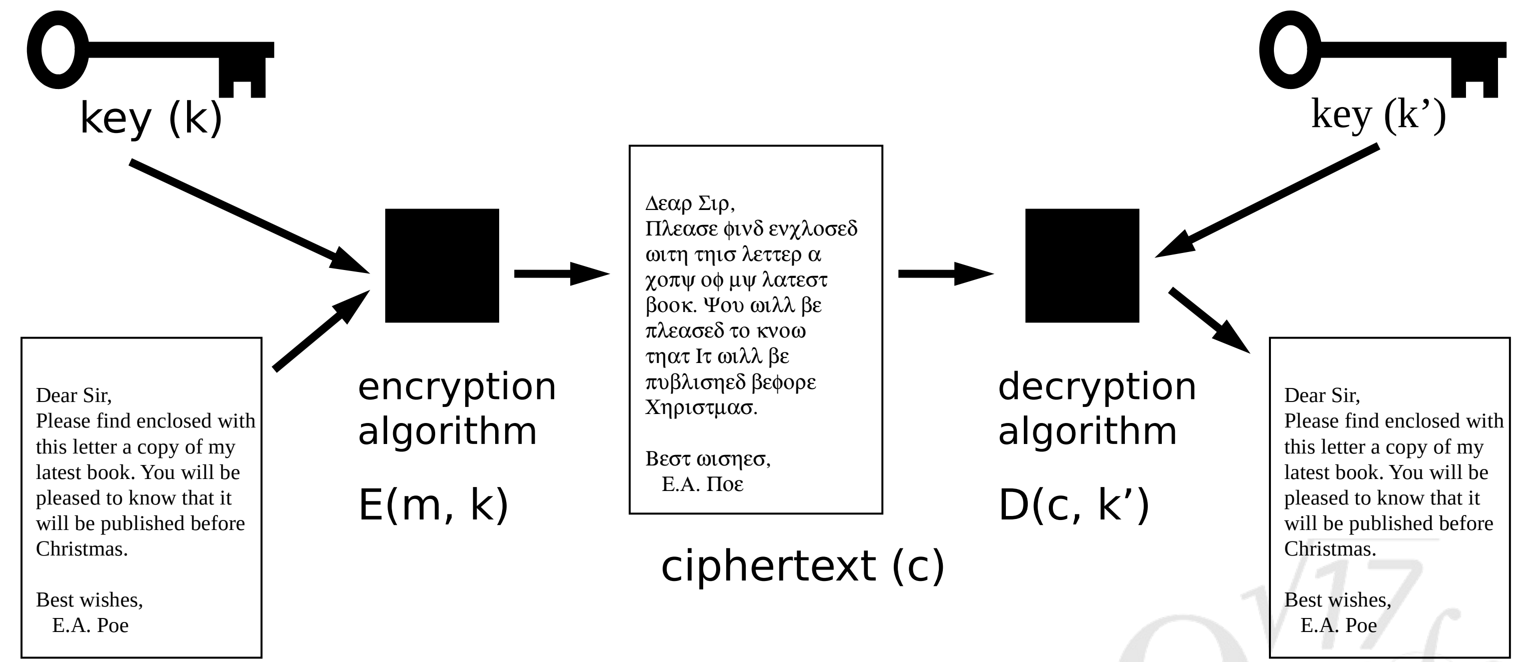
Transposition

Code (replace “words”)

Cipher (replace “letters”)

Source: https://arstechnica.com/information-technology/2012/05/steganography-how-al-qaeda-hid-secret-documents-in-a-porn-video/

Cipher = Algorithm + Key



plaintext (m) plaintext (m)

$$c = E(m, k)$$
$$m = D(c, k') = D(E(m, k), k')$$

No cipher should rely on the secrecy of the algorithm!

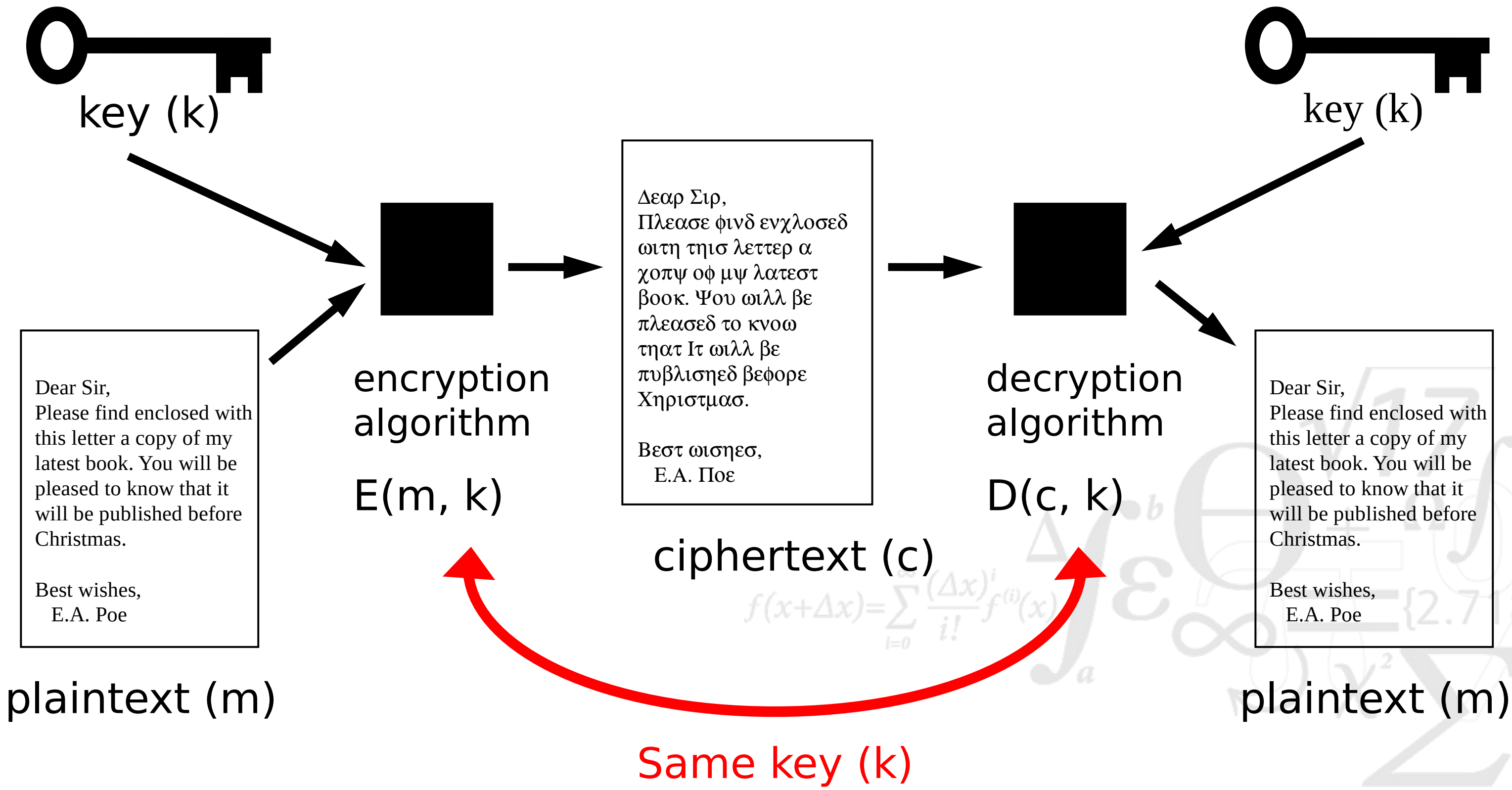
A. Kerckoffs, "La Cryptographie Militaire", 1883

Basic Building Blocks of Cryptography

- Symmetric ciphers (1 key is used)
  - Same key is used for encryption and decryption
- Asymmetric ciphers (2 keys are used)
  - One key is used for encryption
  - Another key is used for decryption
    - *It is not computationally feasible to derive one key from the other*
- Cryptographic Hash Functions (no keys are used)
  - Scrambles input in a unique way
    - *Generates fixed length output from variable length input*
    - *Scrambles input ("decryption" is infeasible)*
- Digital signatures
  - One key signs data (private key)
  - Another key is used to verify signature (public key)
- Advanced algorithms, protocols and constructs not covered here

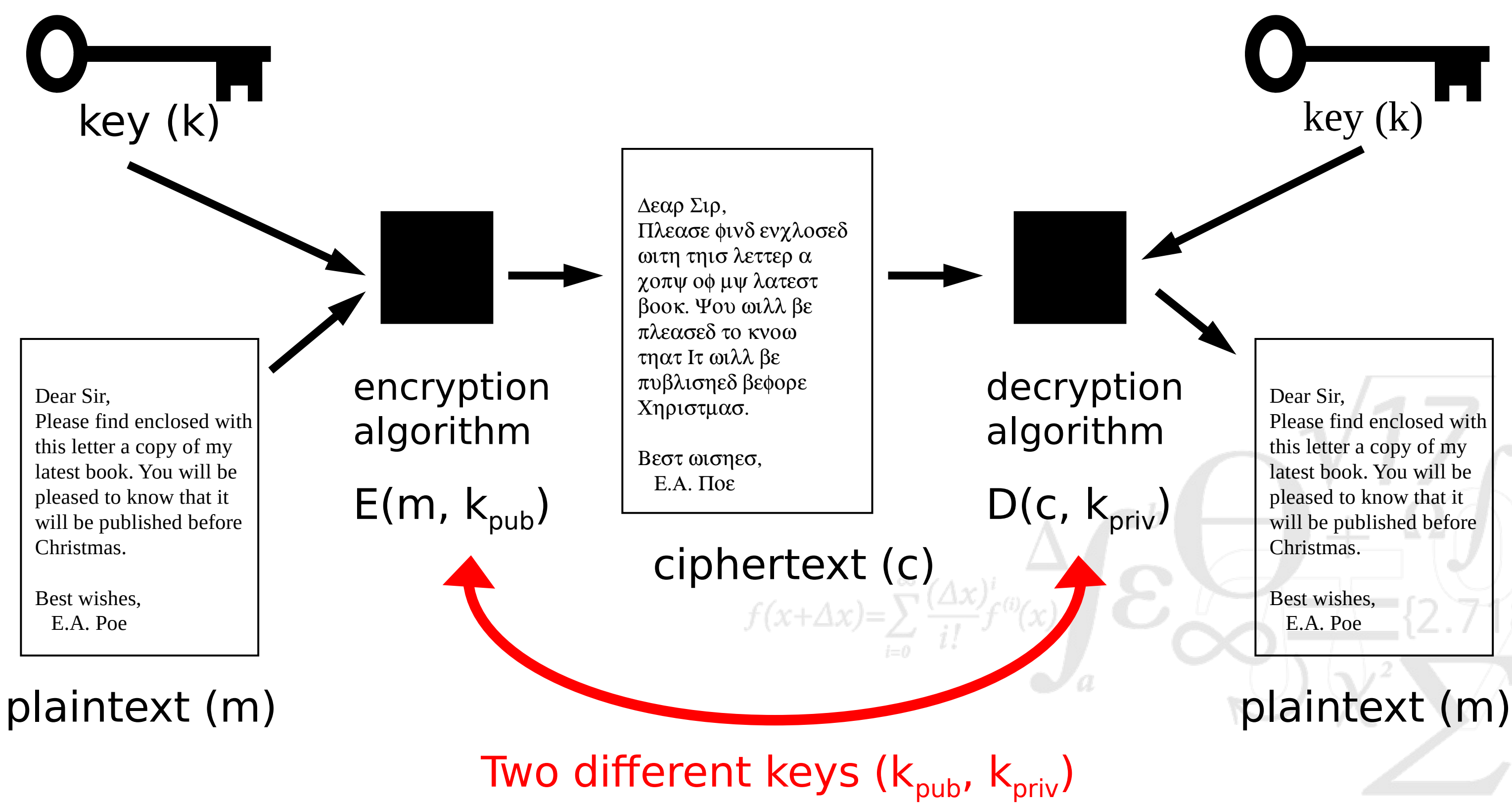
Symmetric Cryptography

- Decryption-key is identical to Encryption-key (or easily derived)

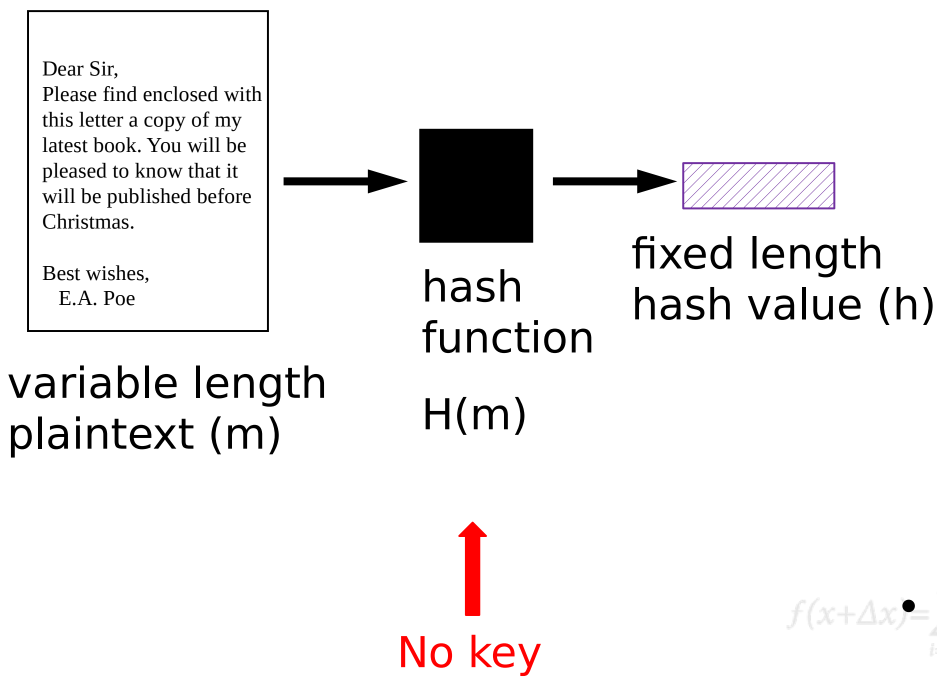


Asymmetric Cryptography

- Decryption-key cannot be derived from Encryption-key



Cryptographic Hash Functions

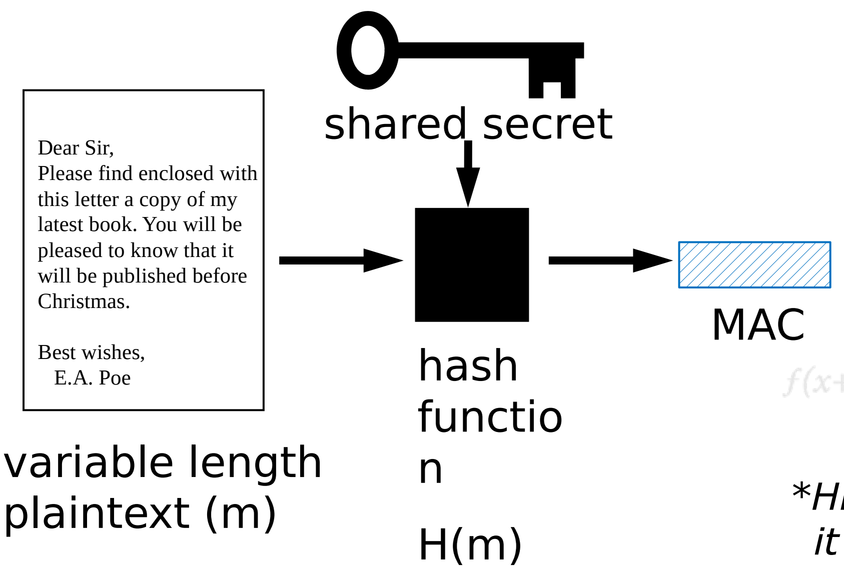


- Cryptographic hash functions must also satisfy:
  1. Given M, computation of h is easy
  2. Given h, it is intractable to compute M such that $H(M) = h$
  3. It is intractable to find M and M' such that $H(M') = H(M)$

- Hash value h is often called a "fingerprint" or "digest" of M

Message Authentication Codes (MAC)

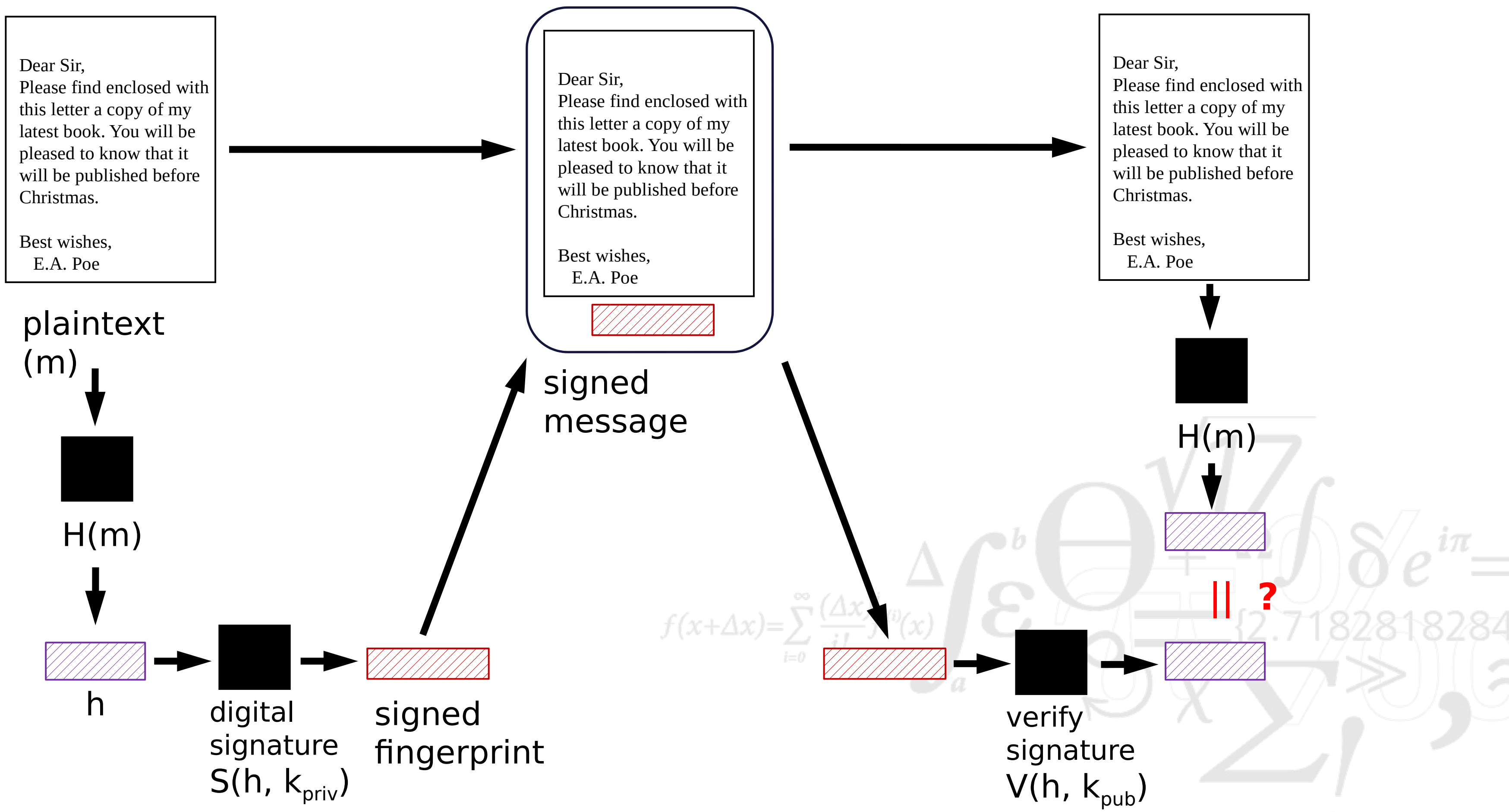
- Message Authentication Codes protect integrity of messages
- Hash-based MAC uses a cryptographic hash function and a shared secret
  - Shared secret (K) is prepended to the message (M) before applying H
  - MAC(K,M) = H(K || M || K)*



**HMAC is a frequently used MAC function; it is defined in RFC 2104*

Digital Signatures

- Operation is expensive, so we normally sign the fingerprint



Security Properties

- Symmetric cryptography
  - Confidentiality of messages
  - Both parties know the shared key (may be more than two parties)
- Asymmetric cryptography
  - Confidentiality of messages
  - Only one party knows the secret key (only party who can decrypt *M*)
- Hash functions
  - Hash value corresponds to given *M* (with very high probability)
- Message Authentication Codes
  - Integrity of message (hash function)
  - Authenticity of message (shared secret)
- Digital signatures
  - Integrity and Authenticity (as with MAC)
  - Non-repudiation of message (assuming security of private-key)

Protection Goals

- Before considering a cryptographic solution, get clear about its protection goals:
  - Confidentiality
  - Integrity
  - Authenticity
  - Non-Repudiation

- People often say: “Our system provides secure communication” or “we need a secure communication channel”
  What does this really mean?

- Cryptography is expensive, so we should only use the building blocks that we need

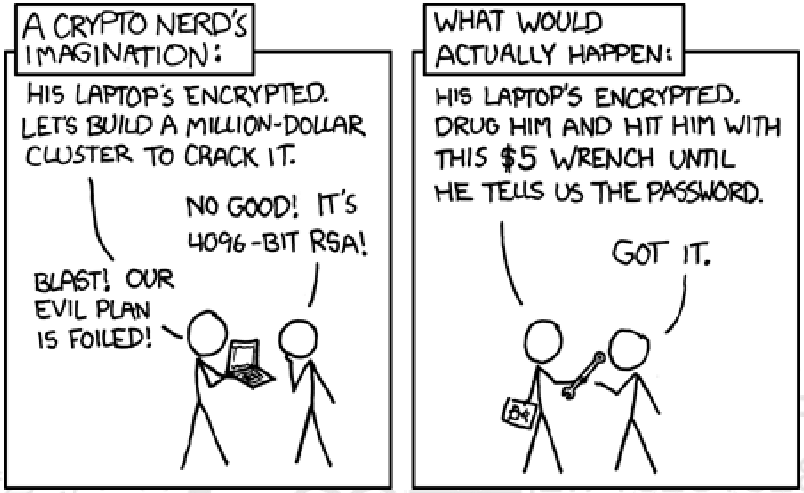
Security of Cryptographic Solutions

- There are 3 classes of attacks on crypto-systems
  - Attacking the cipher (algorithm and mode)
  - Attacking the key (key space, key generation, key management)
  - Attacking the cryptographic protocol

- Most crypto-systems are ”computationally secure”
  - Security often measured in number of operations required by the best known attack (this is known as the “workload” of an attack)
    - *Strong algorithm, brute force attack* => *workload = key-length/2*
  - Hardness of many algorithms is easily overcome with quantum computers

Cryptanalysis
attacking ciphers

- Cryptanalysis attempts to recover plaintext without access to key, but with full knowledge of algorithm
- There are four general types of cryptanalytic attacks:
  - Ciphertext-only attack
  - Known-plaintext attack
  - Chosen-plaintext attack
  - Chosen-ciphertext attack
- Other attacks are equally effective
  - Rubber-hose cryptanalysis aka. purchase-key attack



https://xkcd.com/538

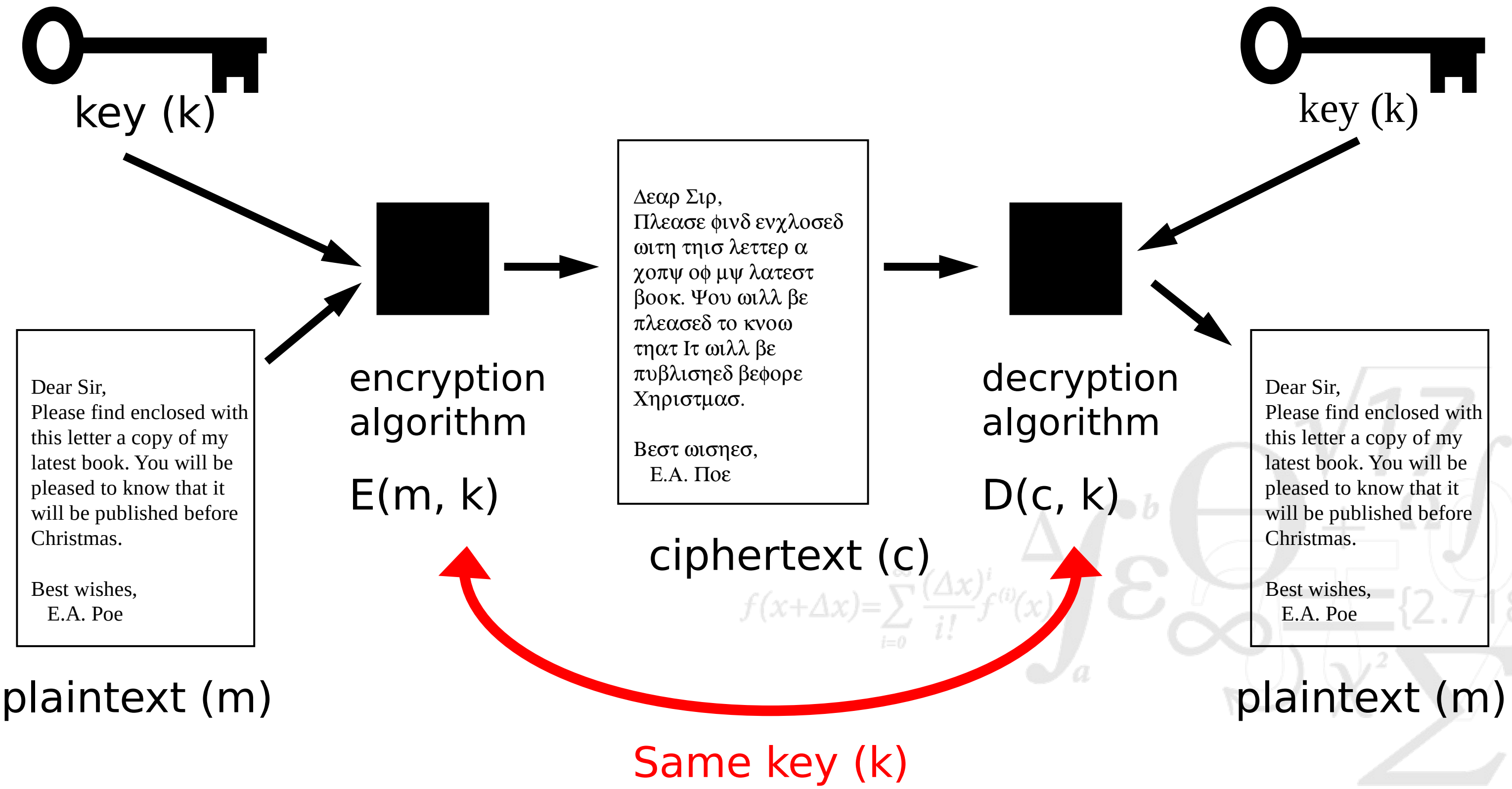
Cryptanalysis
attacking keys

- Key space attacks
  - Exhaustive key search (brute force) attacks are possible with short keys
    - *DES has fixed 56 bit keys, which are cracked very quickly*
    - *AES uses 128, 192 or 256 bit keys (algorithm allows longer keys)*
  - Asymmetric algorithm keys are an order of magnitude longer
    - *Elliptic Curve crypto. achieves equivalent security with shorter keys*
- Key generation attacks
  - 128 bits = 16 random bytes are hard to remember
  - Key derivation functions (KDF) take a password and generates a key
    - *Guessing the password => knowing the key*
- Key management
  - Keys must be stored, shared, distributed, …
    - *All these steps may be attacked*

Coffee Break

Symmetric Cryptography

- Decryption-key is identical to Encryption-key (or easily derived)



One Time Pads

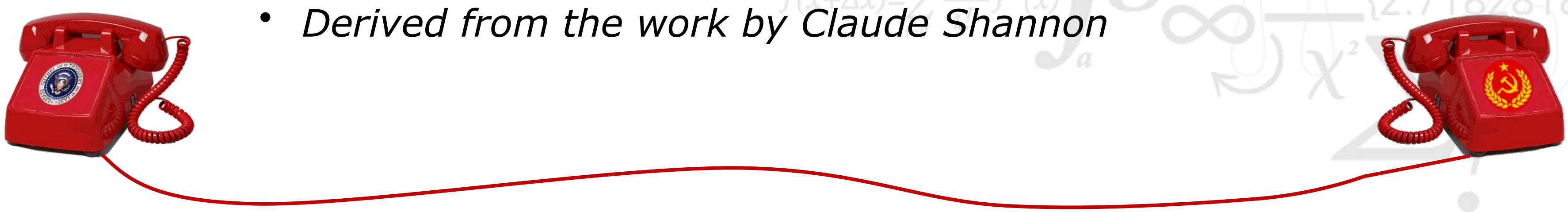
- A One Time Pad consists of a large non-repeating sequence of *truly random* characters
- Encryption xor each letter in the plaintext with the corresponding letter from the one time pad

$$C = P \oplus K$$

- Decryption xor each letter in the ciphertext with the corresponding letter from the one time pad

$$P = C \oplus K$$

- One time pads produce perfectly secure encryption
  - Known as information-theoretic security, cryptosystem cannot be broken by attacker with unlimited resources
    - *Derived from the work by Claude Shannon*

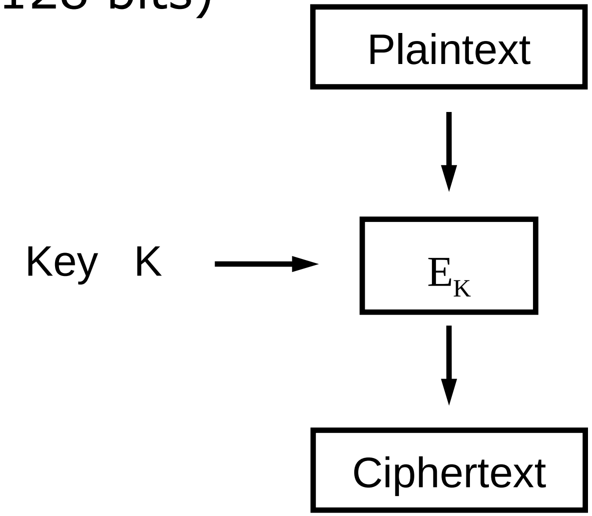


Symmetric Cryptography

- Two main classes of symmetric cryptography:
  - Block Ciphers
  - Streams Ciphers
- Well known Block Ciphers include:
  - DES (badly broken, but found in older textbooks)
    - *Keys are too short to protect against brute force*
  - Triple DES (3DES, still in use, but has been retired by 2023)
    - $C = E_{K3}(D_{K2}(E_{K1}(P)))$
  - AES (current standard adopted by NIST)
- Well known Stream Ciphers include:
  - A5/1 (used in GSM networks, essentially broken)
  - RC4 (difficult to use securely)
  - Block ciphers can be used to construct stream ciphers
    - *This is often the better option*

Block Cipher Algorithms

- Block ciphers operate on blocks of plaintext and ciphertext (usually 32, 64 or 128 bits)



- Algorithm describes how to transform one block of a certain size
  - What happens when plaintext is shorter than block size of algorithm?
  - What happens when plaintext is longer than block size of algorithm?
- Introduces encryption schemes/modes

Algorithms and Schemes/Modes

- Cryptographic algorithms
  - Describe transformation of plaintext/ciphertext to ciphertext/plaintext
  - Do not describe how cryptography is used in systems

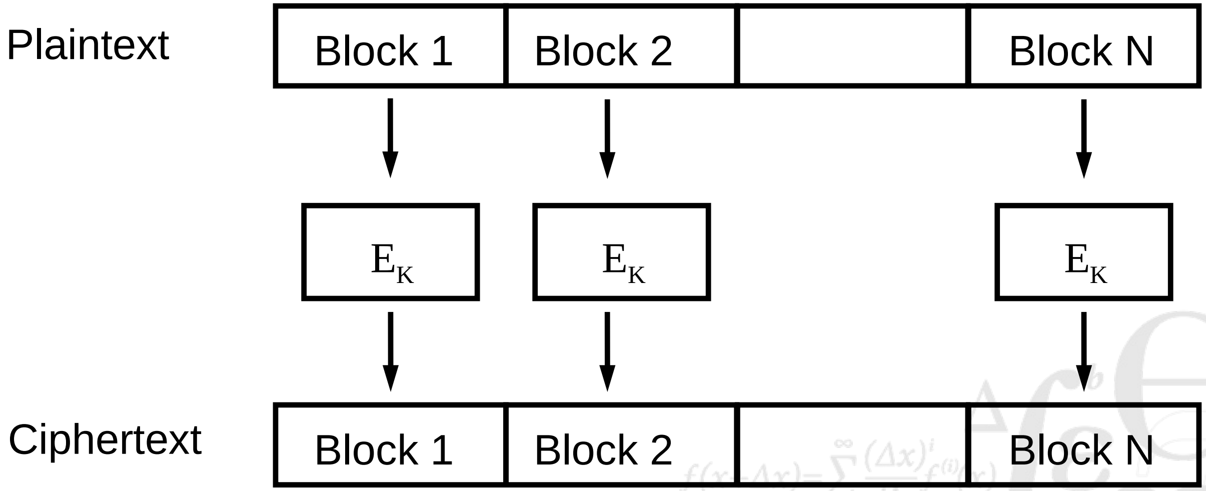
- Algorithms operate on fixed sized data
  - Blocks, stream units (bits/bytes)

- What happens when message size does not fit algorithm size?
  - Message is padded to fit algorithm size

- Cryptographic schemes/modes define:
  - How to encrypt plaintexts of arbitrary lengths
  - How to use initialisation vectors to make each encryption unique

Encrypting Long Plaintexts

- Divide the plaintext into blocks of the defined block size
  - Add padding to the end if necessary

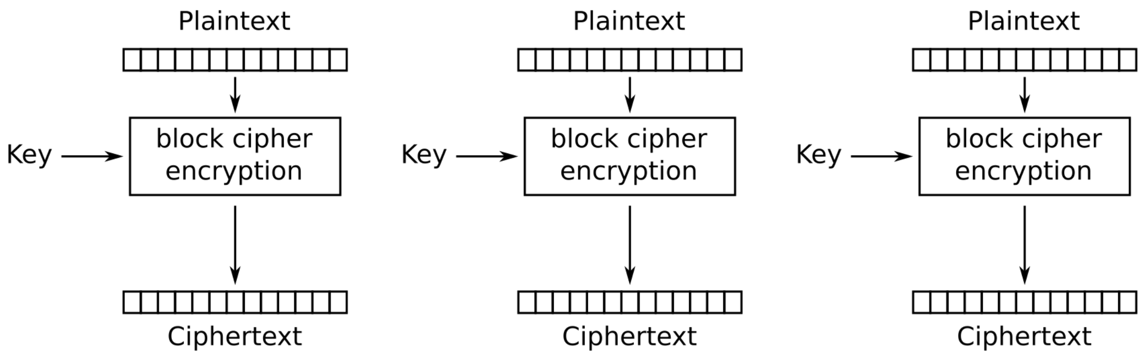


Electronic Codebook Mode

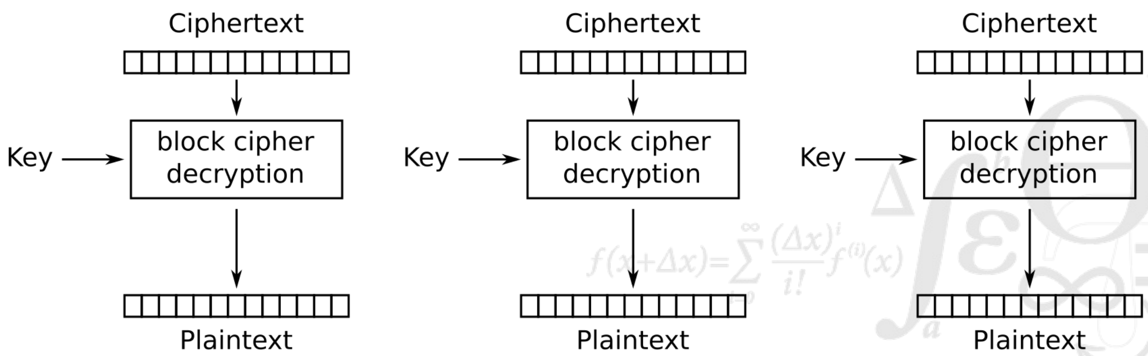
- ECB is the simplest application of a symmetric block cipher
  - plaintext blocks are separately encrypted into ciphertext blocks
    - *Figure on the previous slide*
  - the same plaintext block is always encrypted into the same ciphertext block (with the same key)
    - *Substitution of one "letter" for another "letter" => cipher*
      - Size of AES alphabet is 2^{128}

- Properties of ECB encryption
  - blocks can be en-/decrypted in random order
    - *ECB used in encrypted file systems means that the "seek" operation works*
      - Does require block alignment, but disk blocks are typically multiples of 32, 64 or 128 bits (e.g., 512B, 1024B or 4096B)
  - blocks can be en-/decrypted in parallel
    - *Useful in high speed network communication*

Electronic Codebook Mode



Electronic Codebook (ECB) mode encryption

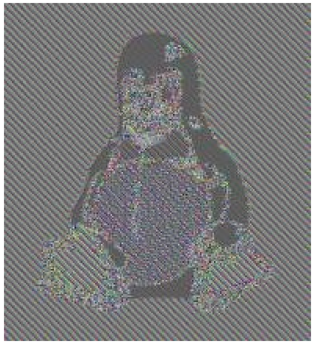


Electronic Codebook (ECB) mode decryption

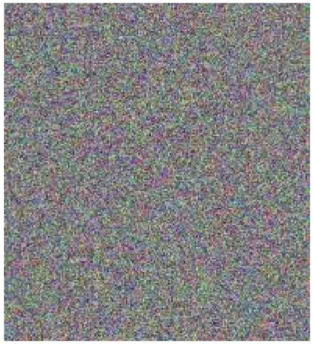
Problems with ECB

- Structure in the plaintext is reflected in the ciphertext

Tux



ECB encoding



Non-ECB encoding

- Messages often have stereotyped beginnings (Dear Foo) and endings (Best regards, Bar)
- Cryptanalyst who learn the encryption of a plaintext block can decrypt the corresponding ciphertext block in all messages encrypted with the same key
- ECB is also vulnerable to *block replay attacks*

Block Replay Attack

- Mallory has access to the network
- He deposits money several times in the Receiving bank (and looks for duplicates)
- He deduces the sending banks encoding of his name, account no. and the amount
- He can now modify other people’s money transfers

Block number

1	2	3	4	5	6	7	8	9	10	11
Time-stamp	Sending bank		Receiving bank	Account holder’s name				Account number		Amount

Field

NB! You don’t need the key to modify the message

Encryption does not protect integrity

Cipher Block Chaining Mode

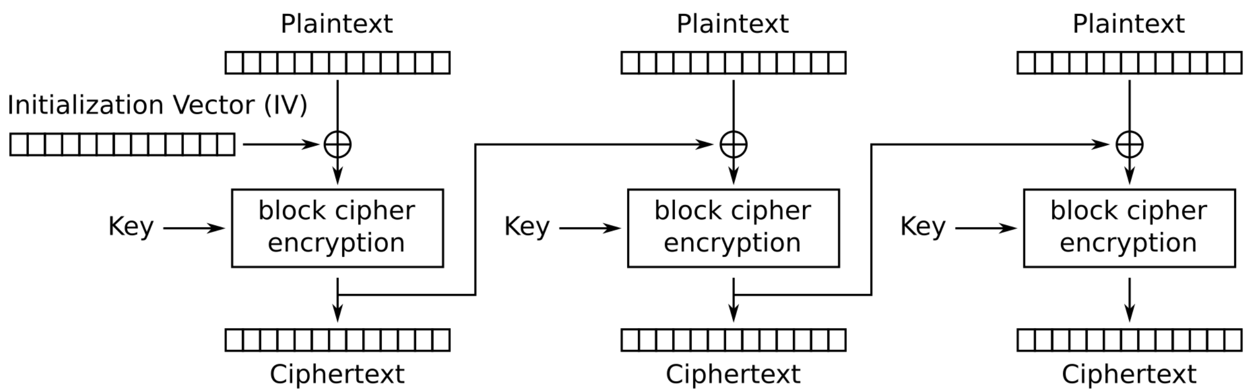
- Introduce a feedback mechanism
  - include the previous ciphertext block in the encryption of the current plaintext block

$$C_i = E_k(P_i \oplus C_{i-1})$$

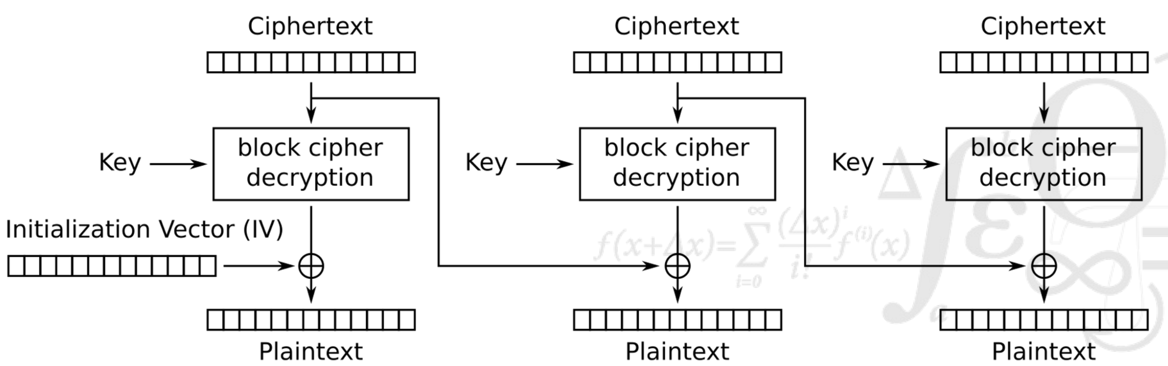
$$P_i = C_{i-1} \oplus D_k(C_i)$$

- An initialization vector (IV) is used to start the process (the IV need not be secret, but must not be reused)
  - Nonce, sequence number, date, …
  - IV may be public, i.e., known to the attacker

Cipher Block Chaining Mode



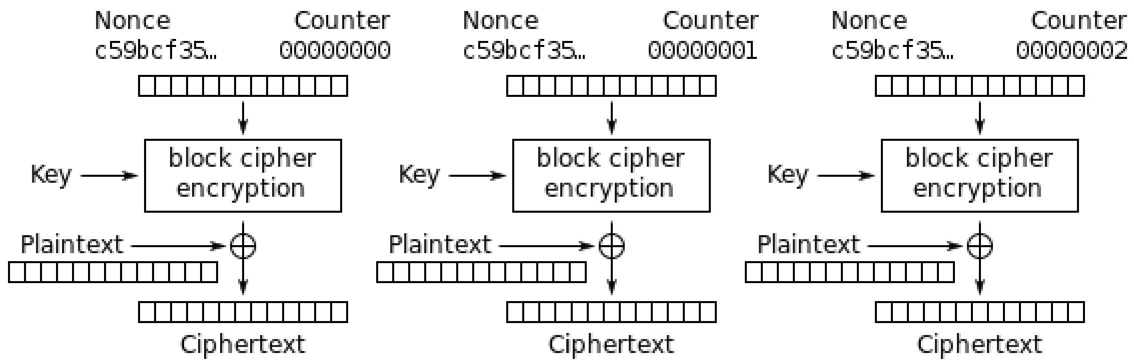
Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

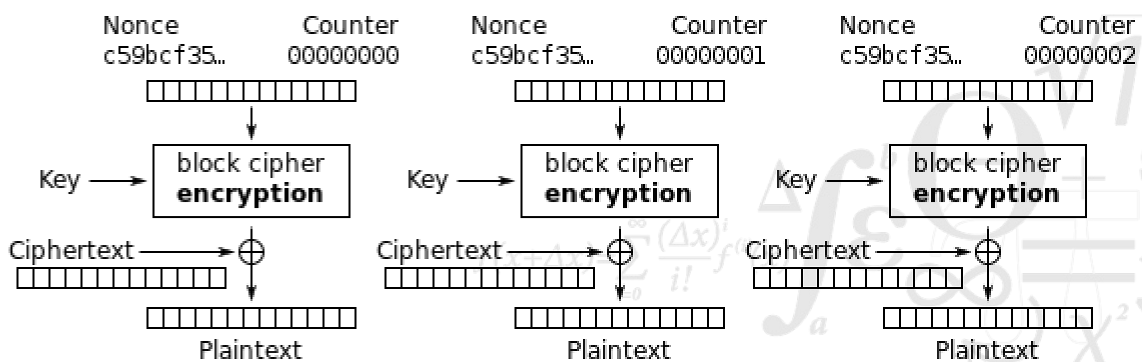
Counter Mode

- Encryption



Counter (CTR) mode encryption

- Decryption

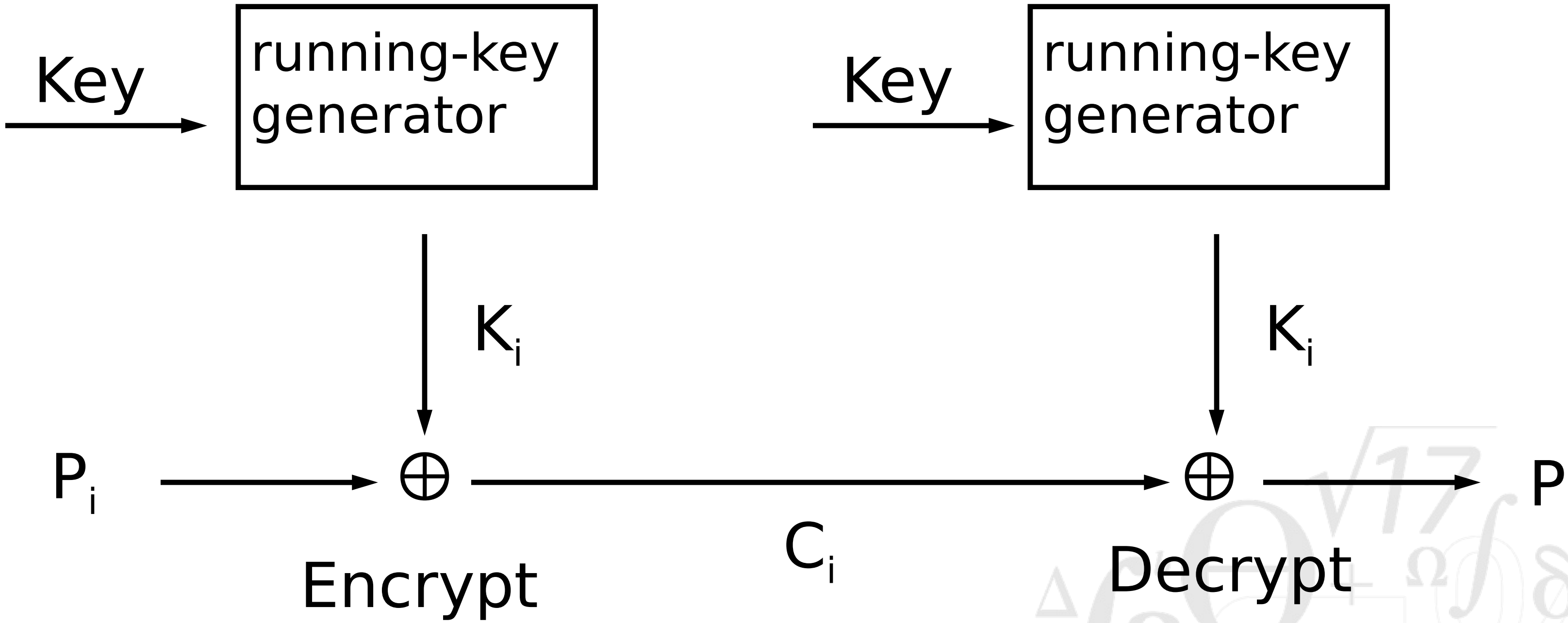


Counter (CTR) mode decryption

Stream Ciphers Algorithms

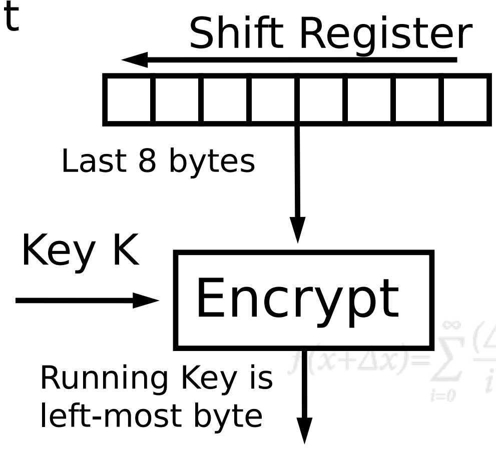
- Stream ciphers work on 1 bit/byte at the time
- A running-key generator generates a pseudo-random string of bits used for encryption
  - same running-key every time means that cryptanalysis is trivial
  - completely random running-key is equivalent to a one time pad (complete security, but not possible)
  - the key is used to seed the running-key generator
- Stream ciphers do not require an entire block to work
  - Good for character based applications

Stream Ciphers II



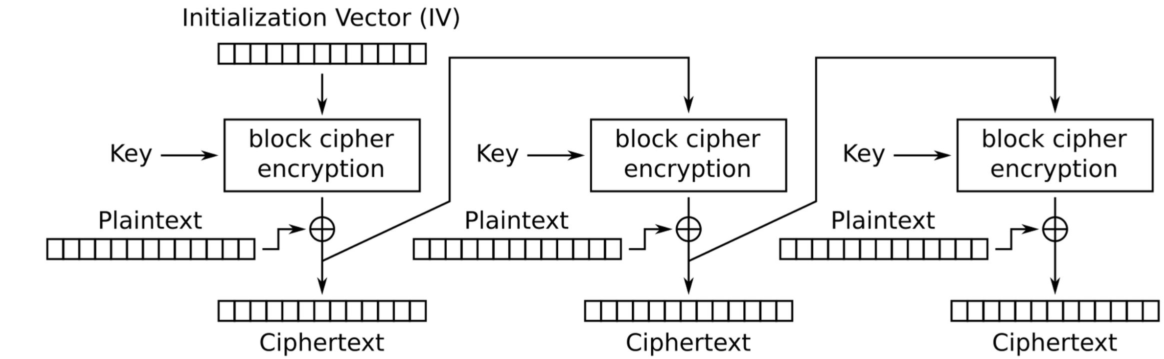
Creating Stream Ciphers from Block Ciphers

- Block ciphers can be used to implement stream ciphers
- Example: a 64bit block cipher is used to implement a byte stream cipher
  - A "shift register" (64bit) is encrypted and the leftmost byte is XORed with the plaintext byte
  - the shift register is shifted 1 byte to the left and the ciphertext byte is added to the right

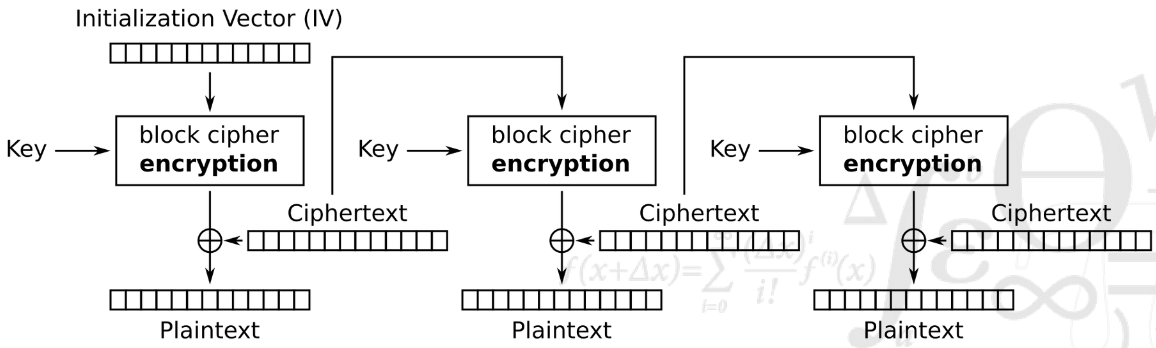


- The IV must be unique, e.g., a serial-number

Cipher Feedback Mode



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

Authenticated Encryption

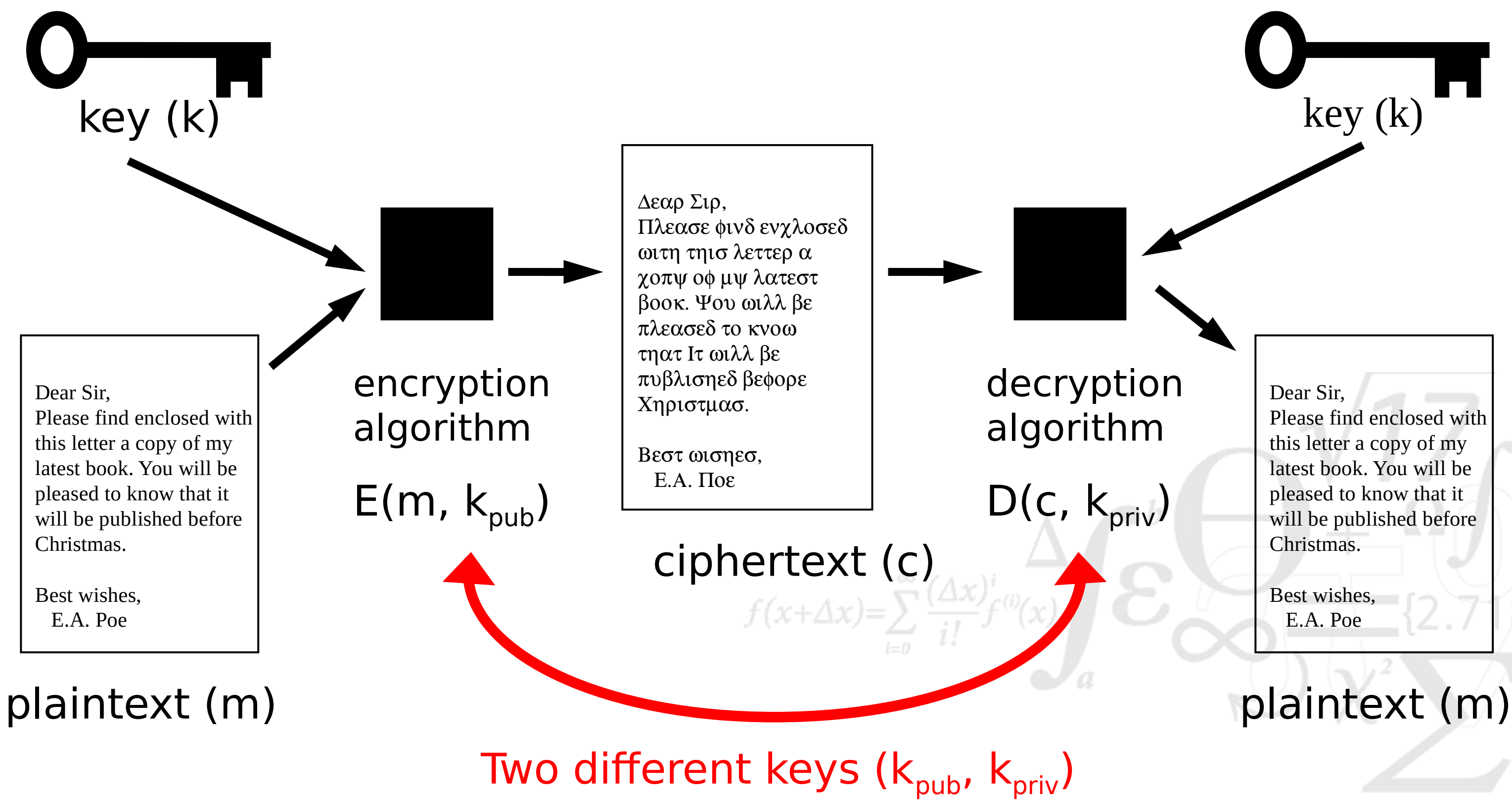
- The goal is to ensure both data secrecy (confidentiality) and data integrity
- In practice, even to achieve only data secrecy, authenticated encryption is preferred
- The main idea is to combine a cipher secure against Chosen-Plaintext Attack (CPA) and a secure MAC

Authenticated Encryption = CPA-Secure + secure MAC

- In the past, developers needed to combine these two primitives
- Now, standard exists such Galois Counter Mode (GCM) that combine a random counter encryption mode and a secure MAC

Asymmetric Cryptography

- Decryption-key cannot be derived from Encryption-key



Limitations

- Compared to symmetric cryptography, asymmetric cryptography is inefficient (slow, long keys)
  - In practice it is not used to encrypt large amounts of data
    - *Building block for key exchange*
    - *Building block for other protocols*

Sample Asymmetric Encryption Algorithms

- Asymmetric encryption is based on computationally hard problems
  - Problems that we cannot solve efficiently on computers
- Important groups of asymmetric encryption algorithms
  - RSA (prime factorization of large numbers)
  - Diffie-Hellman (modular arithmetic)
  - Diffie-Hellman (elliptic curves)
- Asymmetric encryption schemes define how algorithms are used in practice, i.e., arbitrary length messages, padding, etc.
- Particularly important: Semantic Security
  Encryption scheme has to randomize the message, since otherwise, the following attacks become possible
  - Guess the message's content (guess M that corresponds to given C)
  - Use (known) public key to check whether the guess is correct

RSA

- Most popular public-key cryptosystem
- 1977: Rivest, Shamir, Adleman (RSA)
- Both encryption and authentication
- Survived years of attacks (cryptanalysis)
  - Probably secure
- Part of many official standards worldwide
  - Interoperability with existing code base

RSA overview

- Based on the difficulty of factorization
- Pick two random large primes: p and q
- Calculate $n = pq$ (n is called the modulus)
- Chose random encryption key e (e is called the exponent) such that:

  e and $(p-1)(q-1)$ must be relatively prime
- Compute decryption key d (extended Euclidian algorithm), such that:

$$ed \equiv 1 \bmod (p-1)(q-1)$$

- Public key = (e, n) Private key = d

Using RSA

- Encryption of a plaintext M
  - divide M into numerical blocks $m_i < n$

  $c_i = m_i^e \pmod{n}$

- Decryption of ciphertext block c_i

  $m_i = c_i^d \bmod n$

  because

  $c_i^d = (m_i^e)^d = m_i^{ed} = m_i^1$ (all mod n)

ElGamal

- Based on the difficulty of calculating discrete logarithms in a finite field

$$y \equiv g^x \bmod p$$

  - p is prime
  - g and x are random numbers less than p

- Public key = (y, g, p)
- Private key = x

ElGamal Encryption

Encryption of message *M*

- Select random k relatively prime to $p - 1$

$$a = g^k \bmod p$$

$$b = y^k M \bmod p$$

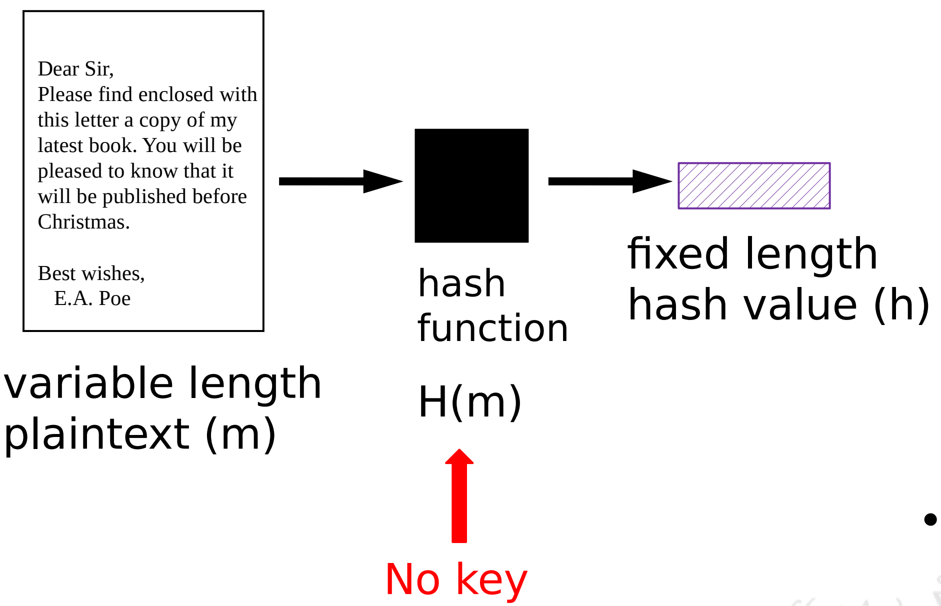
- The pair (a,b) is the ciphertext

Decryption of a and b

$$M \equiv \frac{b}{a^x} \pmod{p}$$ *Because:*

$$a^x \equiv g^{kx} \pmod{p} \wedge \frac{b}{a^x} \equiv y^k \frac{M}{a^x} \pmod{p} \equiv g^{kx} \frac{M}{g^{kx}} \pmod{p} \equiv M \pmod{p}$$

Cryptographic Hash Functions



- Cryptographic hash functions must also satisfy:
  1. Given M, computation of h is easy
  2. Given h, it is intractable to compute M such that $H(M) = h$
  3. It is intractable to find M and M' such that $H(M') = H(M)$
- Hash value h is often called a "fingerprint" or "digest" of M

Hash Functions

- The “work horses” of cryptography

- Main uses include:
  - Condensing long strings into short strings (collision resistant hash function)
  - Making an irreversible transformation without a key (one-way function)

- Prominent Examples:
  - MD4, MD5, SHA-0 (badly broken)
  - SHA-1 (broken, still found in standards, but must be retired before 2030)
  - SHA-2 (current standard)
  - SHA-3 (newest standard – since 2015)

Collision Resistance

- Intractable to find random M and M' such that

$$H(M) = H(M')$$

- Collisions invalidates the use of hash functions in digital signatures
  - Alice creates to contracts M and M', where M is fair, but M' is favourable to the Alice
  - Bob signs M: Sign(H(M), Bob_{Priv})
  - Since H(M) = H(M') Alice can present M' as if it was signed by Bob

Birthday Paradox and Birthday Attack

- Standard statistics problem

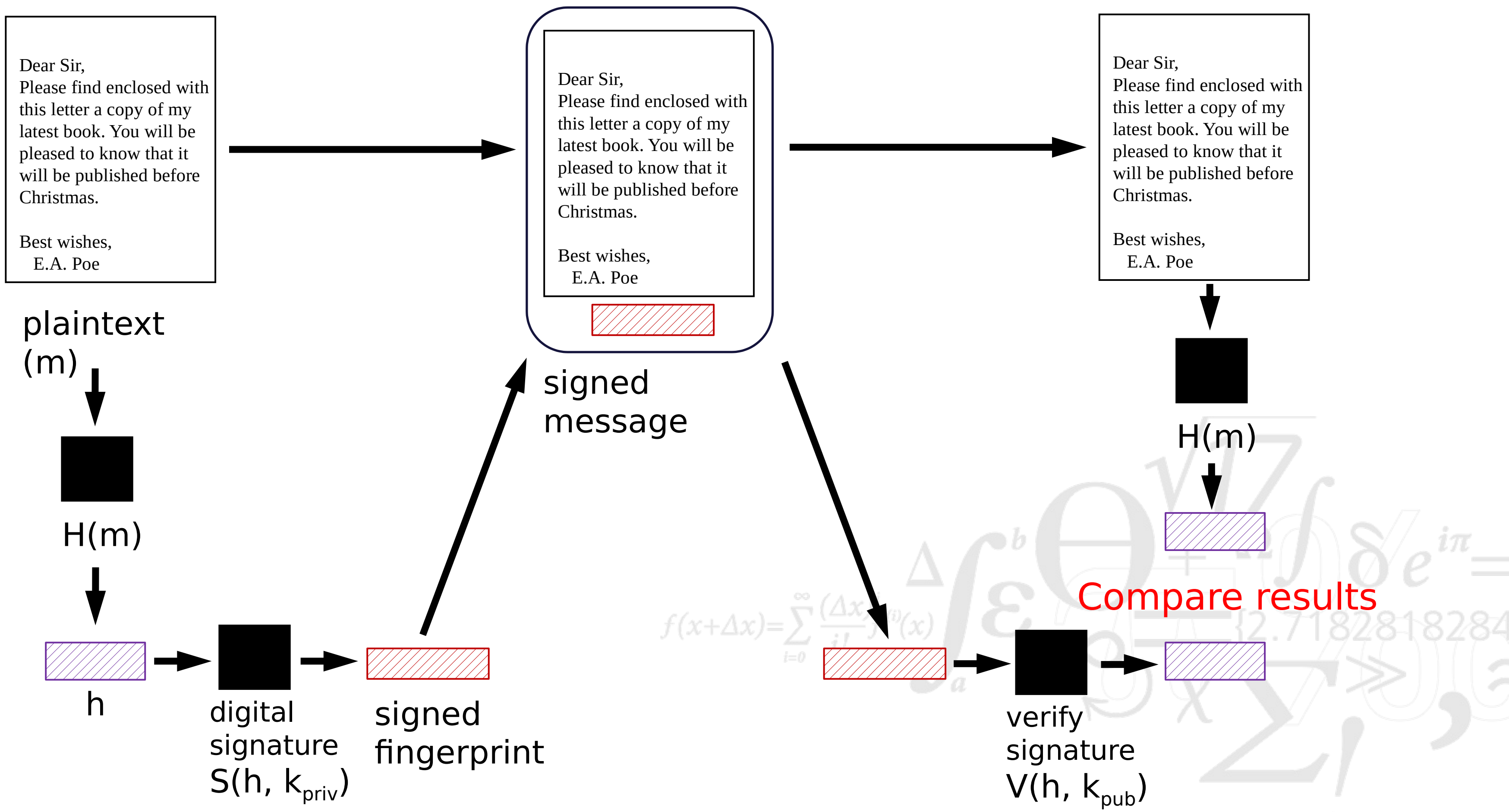
  How many people do you need in a room to have better than 50% chance of two persons with the same birthday?

  - someone with your birthday 253
  - any two with the same birthday 23
  - Any two born of the same day of the month 10

- *m* bit hash function
  - 2^m random hashes needed to find a particular *h*
  - $2^{m/2}$ random hashes needed to find two messages with the same hash value

Digital Signatures

- Operation is expensive, so we normally sign the fingerprint



Security Level

- If the best known attack is equivalent to running the cryptographic algorithm 2^n times, then we have a security level of n bit

- Typical 80 bit (too low), 128 bit (decent), 256 bit (high)

- Often, the security level is equal to the key length

- Important exceptions:
  - Hash functions (hash size $\geq 2 *$ security level)
  - Asymmetric cryptography (e.g. RSA: 1024/2038/4096 bit)

Summary on Building Blocks

- Please remember the following
  - Clarify your security goals
  - Don’t design your own cryptographic primitives and protocols
    - *Always rely on well known (and analysed) standards*
  - Make sure that you understand the primitives and protocols you use:
    - *Security goals*
    - *Security level*
    - *How to apply them*
    - *Known problems and pitfalls*

Lab next week

- Create an account on https://cryptohack.org
- Select ”Courses”
- This is based on self study and there is no graded assignments