Cryptographic Services

Segurança Informática em Redes e Sistemas
2024/25

David R. Matos, Ricardo Chaves

Ack: Miguel Pardal, Miguel P. Correia, Carlos Ribeiro

Roadmap

- Cryptography
- Criptographic services
  - What we want to provide
- Criptographic building blocks
  - Primitive and composite functions
  - No technology details (yet)
- How to provide the services using the functions

Cryptography: terminology

- Cryptography
  - Art or science of writing in a concealed form
    - from Greek: kryptós, hidden + graph, r. de graphein, write
  - Used to ensure confidentiality of data (until the 1970s)
  - Steganography
    - from Greek: steganós, covered + graph, r. de graphein, write

- Cryptanalysis
  - The art or science of breaking cryptographic systems or ciphered data

- Cryptology
  - Cryptography + Cryptanalysis

Steganography in ancient history

*In ancient Greece, **Histiaeus**, the ruler of Miletus, shaved a slave's head, tattooed it with a message, and waited for the hair to grow back.*

He then sent the messenger on the long journey from Persia to Greece to urge revolt.

*Upon arrival, the messenger's head was shaved again to read the message.**

*https://thereader.mitpress.mit.edu/a-brief-history-of-secret-communication-methods/

Cryptography in ancient history

Scytale: used for transposition cipher

It is a Cylinder with a strip of parchment wound around it on which is written a message

*The **key** is a rod with the right diameter*

Cryptography in ancient history

Caeser ciphers are simple substitution ciphers. Each letter in the plaintext is shifted a certain number of places down the alphabet.

Cryptography

- Widespread and dangerous belief:
  - Encrypting everything provides protection against anything
- A simple example to prove the contrary:
  - Money transfer from one bank to the other
    - The bank encrypts the whole message
  - The attacker:
    - Might not be able to understand the message! (or can he?)
    - But he might be able to:
      - Divert the message into his account (maybe not!)
    - Could get rich by:
      - Diverting or stopping debit messages
      - Allow the passage of all credit messages
      - He might be able to distinguish the two merely by looking at their size
    - Crash the bank by:
      - Injecting random messages

Cryptanalysis: what cryptography must protect from

- Basic assumption: the algorithm is known
  - If not public, might be obtained (e.g., stolen)
- Attacks:
  - Ciphertext-only: cryptanalyst has access to ciphertexts
    - Without them, no cryptanalysis is possible
  - Known-plaintext: cryptanalyst has a set of ciphertexts to which he knows the corresponding plaintext
    - Often easy to get at least partial plaintext, e.g., message beginning
  - Chosen-plaintext: cryptanalyst can obtain the ciphertexts corresponding to plaintexts of his choice; or:
    - Chosen-ciphertext: cryptanalyst can obtain the plaintexts corresponding to ciphertexts of his choice

Easier for attacker but harder to get

Attacks on information

- We want to protect the information against:
  - Unauthorized insertion of information
  - Unauthorized modification of information in transit
  - Unauthorized replay of information
    - From an earlier legitimate data transmission
  - Unauthorized access to information

- Which cryptographic services can we use to prevent this?

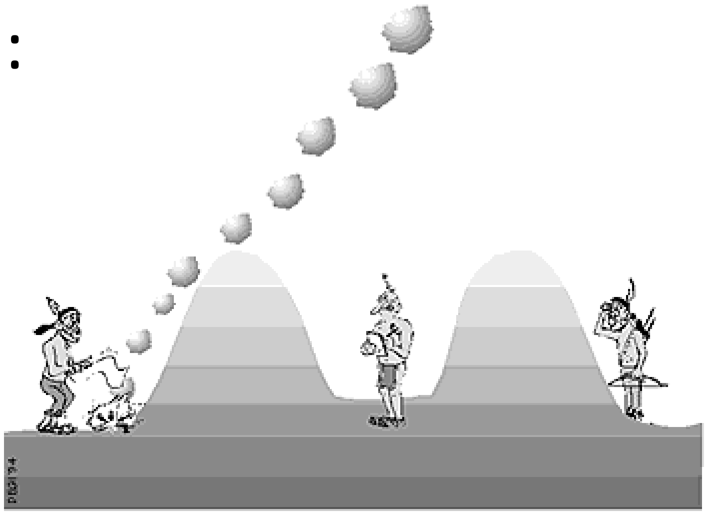
CRYPTOGRAPHIC SERVICES

Cryptographic services

- We need the following cryptographic services:
  - Confidentiality
  - Integrity
  - Authenticity
    - Entity authentication
    - Data origin authentication
    - Non-Repudiation

1 - Confidentiality

- Is a service used to keep the content of the information from all, but those entities authorized to have it
  - i.e. making the information unintelligible to all but those who possess some secret
- Typically achieved by encryption:
  - Process of converting plaintext to ciphertext using
    - Cryptographic algorithms
    - Cryptographic key



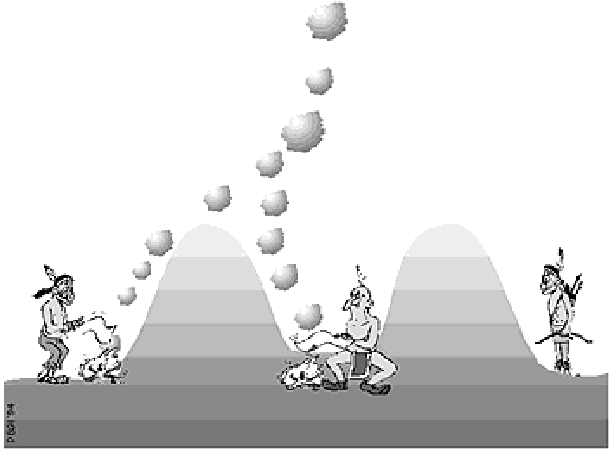
Confidentiality challenges

- Makes debugging harder
  - Software
  - Systems
  - Protocols
- Information loss
  - If the key is permanently lost, so is the information
- Sometimes misused
  - Other, more appropriate, services can be used

2 - Integrity

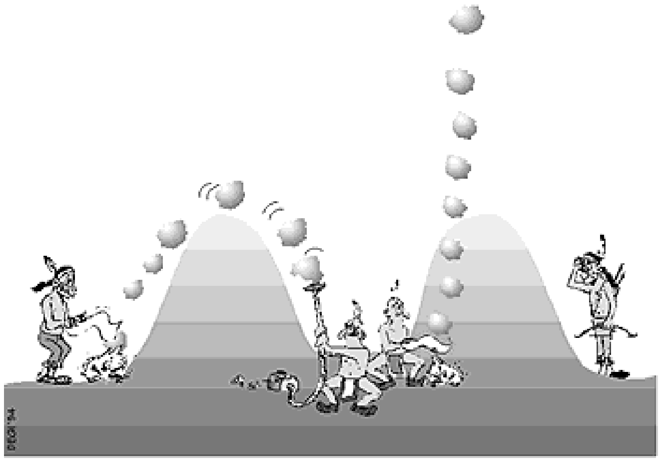
- Is a service that detects data manipulation by unauthorized entities
  - An intruder should not be able to substitute a false message for a legitimate one

- Not the same thing as error detection codes
  - e.g. Cyclic Redundancy Codes (CRC) are not cryptographically strong
    - do not protect against intentional alterations of the message



3 - Authenticity

- Is a service used to ascertain the identity or the origin of a message:
  - Guarantees that entities are who they claim to be
    - Verified identification
    - Data origin authentication
  - Requires *message integrity* and *freshness*
    - Tamper detection
    - Replay detection



Authentication

- Entity authentication
  - Verify the identity of an entity
  - Ensure legitimacy of parties involved in a communication
    - Sender authenticates itself to Receiver
    - Receiver checks evidence and decides to accept identity
  - Spoofing/impersonation must be infeasible
- Data origin authentication
  - Confirm the originator/creator of the message
  - Detect message tampering and replay
- All these features must remain true
  - Even after a large number of honest message exchanges

Device versus User Authentication

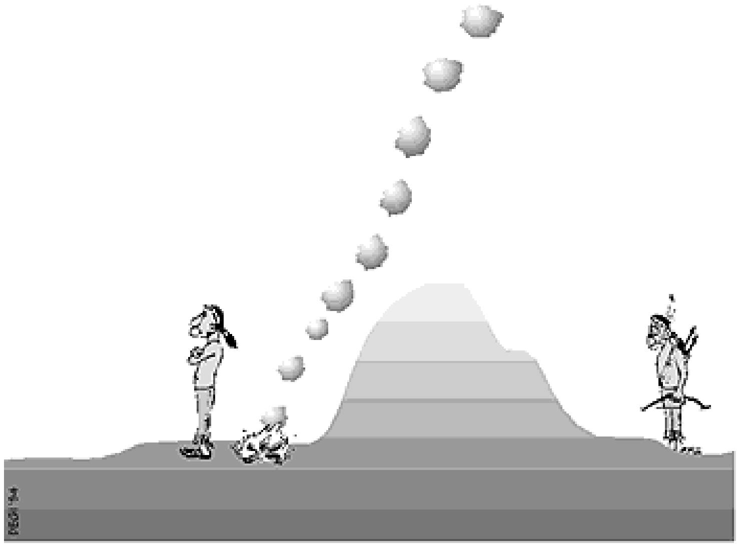
- Device Authentication
  - Validating a specific device
  - Often using digital certificates, network addresses, or device-specific keys
  - Not bound by human limitations

- User Authentication
  - Verifying the identity of an individual user
  - through credentials
    - Passwords
    - Biometrics
    - Hardware tokens

- We will come back to this later in the course

Non-Repudiation

- It is a service which prevents an entity from denying previous commitments or actions
  - Such as:
    - a sent message
    - a signed document
    - …



CRYPTOGRAPHIC BUILDING BLOCKS

Primitive Building Blocks

- Cipher
  - Symmetric
  - Asymmetric
- Hash

Cipher and decipher functions

- Defines two functions: cipher, decipher
- Cipher function
  - Receives data and key
  - Outputs cryptogram
- Decipher function
  - Receives cryptogram and key
  - Outputs original data, if the key is correct
    - Otherwise, returns something else

Symmetric Cipher

- Uses the same key to cipher and decipher

- In cryptographic function notation:
  - E(M, K) – cipher message M with key K
    - Produces cryptogram C
  - D(C, K) – decipher cryptogram C with key K
    - Produce message M’
    - M’ = M if the key is correct

  - D(E(M, K), K) = D(C, K) = M

Asymmetric Cipher

- Instead of a key, we have a key pair:
  - One part we call the private key – KR
    - Only known by one entity
  - The other part we call public key – KU
    - Can be known by everybody else
- In cryptographic function notation:
  - AE() – Asymmetric Encryption
  - AD() – Asymmetric Decryption
- We can cipher with one key and decipher with the other

Cipher with private, decipher with public

- In cryptographic function notation:
  - AE(M, KR) – cipher message M with private key KR
    - Produce cryptogram C
  - AD(C, KU) – decipher cryptogram C with public key KU
    - Produce message M’
    - M’ = M if both keys belong to the same pair

- What do we know about the cryptogram C?
  - Only the owner of the private key KR can produce it
  - Anyone with the public key KU can decipher it

Cipher with public, decipher with private

- In cryptographic function notation:
  - AE(M, KU) – cipher message m with public key KU
    - Produce cryptogram C
  - AD(C, KR) – decipher cryptogram c with private key KR
    - Produce message M’
    - M’ = M if both keys belong to the same pair

- What do we know about C in this case?
  - Anyone with the public key KU can produce it
  - Only the owner of the private key KR can decipher it

Cryptographic Hash

- A cryptographic hash function receives an input message and returns a digest of the data
  - Does not use a key

- In cryptographic function notation:
  - H(M) – hash message M
    - Produce digest DT

Digest value produced by hash

- What do we know about the digest value DT?
  - Deterministic
    - The same input always produces the same digest value
  - Fixed Size
    - Digest values are of a fixed length, independent of the input size
  - Unique Representation
    - Ideally, each input produces a unique digest, though collisions can occur
  - Non-reversible
    - Hash is a one-way function
    - It is computationally infeasible to derive the original input from digest
  - Sensitive to Input Changes
    - Small changes in input significantly alter the digest (avalanche effect)

Composite Building Blocks

- Hybrid Cipher
- Integrity Check
  - Message Integrity Code
  - Digital Signature

Hybrid Cipher

- Typically, symmetric ciphers are 100 to 1000 times faster than asymmetric ciphers
  - Mathematical operations used in symmetric cryptography are simpler
- How can we have the best of both?
  - Generate random symmetric key KM
  - Cipher (large) message M with symmetric cipher
  - Cipher (small) key KM with asymmetric cipher
  - We get the same properties of asymmetric cipher with the performance of symmetric cipher
- Functions:
  - HE (Hybrid Encryption) and HD (Hybrid Decryption)

Hybrid Cipher in detail

- In cryptographic function notation:
  - Generate random key for message: RND()
    - Produce message key KM
  - Cipher the message key with public key of receiver: AE(KM, KU)
    - Produce cryptogram of key CK
  - Cipher the message: E(M, KM)
    - Produce cryptogram of message CM
  - Transmit CK, CM
  - Decipher the message key with receiver private key: AD(CK, KR)
    - Obtain received key KM’
  - Decipher the message: D(CM, KM’)
    - Obtain received message M’

Message Integrity Code

- Is it possible to detect changes to a message?
  - Using a hash function H and a secret K
  - Compute a value that can be used to detect changes in received message M’
- Function: MIC (Message Integrity Code)
  - With freshness, can be used to provide authenticity, so, very often, it is called a MAC (Message Authentication Code)

MIC in detail

- In cryptographic function notation:
  - E(H(M), K) – digest the message and cipher result
    - Produces the MIC value
  - Transmit message M and MIC value
  - To verify:
    - Compute E(H(M’), K) and compare with received MIC
    - Same? Then the message did not change
  - Another approach, using decryption:
    - Compute DT’ from D(MIC’, K) and compare with H(M’)
    - Same? Then the message did not change

HMIC

- HMIC stands for Hash-based Message Integrity Code
  - Also called HMAC
- Is another approach, without using ciphers
  - Better performance
- Function MIX combines the data with the secret
  - For example, with XOR or some specific concatenation
- How to use the HMIC?
  - Compute H(MIX(M, K)) and compare with H(MIX(M’, K))
  - Same? Then the message did not change

Digital Signature

- Is it possible to detect changes to a message and confirm the sender?
  - Still using a hash function H but now with asymmetric keys KR KU
  - Compute a value that can be used to detect changes in received message M’

– Function: DS (Digital Signature)

DS in detail

- In cryptographic function notation:
  - AE(H(M), KR) – digest the message and cipher result with the private key
    - Produces the DS value
  - Transmit message M and DS value
  - To verify:
    - Compare deciphered hash with recomputed hash
      - Compute AD(DS’, KU) to obtain DT’ and compare with H(M’)
    - Same? Then the message did not change and was sent by a holder of the private key

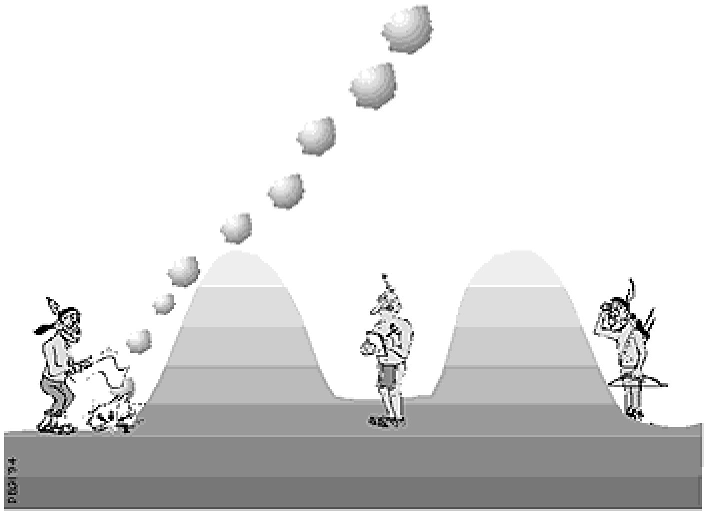
PROVIDING CRYPTOGRAPHIC SERVICES

Cryptographic services (revisited)

- We can now design cryptographic services:
  - Confidentiality
  - Integrity
  - Authenticity
- To protect against:
  - Unauthorized insertion of information
    - Loss of authenticity
  - Unauthorized modification of information in transit
    - Loss of integrity
  - Unauthorized replay of information
    - Loss of authenticity
  - Unauthorized access to information
    - Loss of confidentiality

1 - Confidentiality

- Use symmetric cipher
  - If a secret is shared
- Use asymmetric cipher
  - If public keys are shared
  - More efficient with hybrid cipher



Confidentiality in detail

- Alice wants to send a message to Bob that cannot be read by anyone else

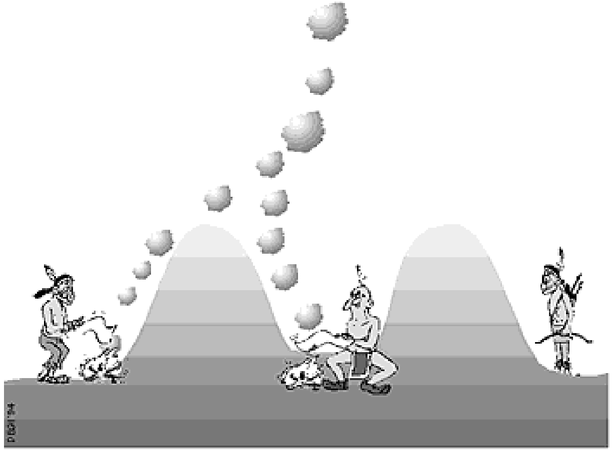
Alice

Bob

- How?
  - With shared secret key
  - E? K?

2 - Integrity

- Use MIC
  - If a secret is shared
- Use DS
  - If public keys are shared



Integrity in detail

- Alice wants to send a message to Bob that cannot be written by anyone else without detection

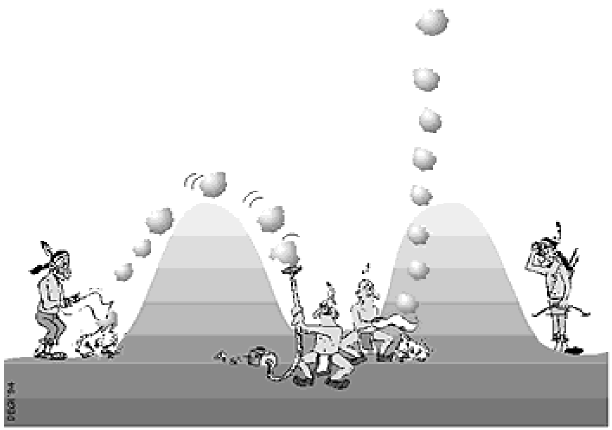
Alice

Bob

- How?
  - With shared public keys
  - AE? KRa? KRb?

3 - Authenticity

- *Integrity* assured with MIC or DS
- Freshness requires adding a nonce N to the message
  - Number used Once
    - Random number RN
      - But… receiver needs to memorize them to detect replays
    - Counter CTR
      - But… messages must be received in order
    - Timestamp TS
      - But… clocks must be synchronized
    - Combination of two of the above



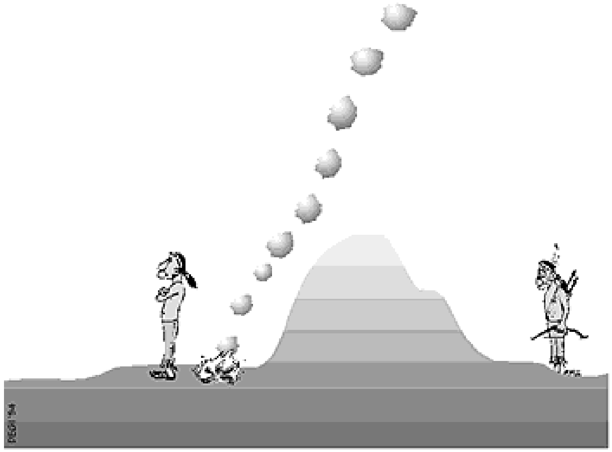
Authenticity in detail

- Alice wants to send an authentic message to Bob

- How? Also need freshness
  - Add nonce
    - Which one?
  - AE? KRa? KRb?

Non-Repudiation

- A digital signature can provide non-repudiation, if…
  - the signer is only entity that knows the private key



Confidentiality + Authenticity?

- Alice wants to send a confidential and authentic message to Bob

- KUa? KUb?

Summary

- Cryptography allows us to protect information
  - With ciphered data to prevent reads
  - With digests to allow detection of writs
- We can use cryptographic functions to provide cryptographic services for:
  - Confidentiality
  - Integrity
  - Authenticity
- Next: cryptographic technology