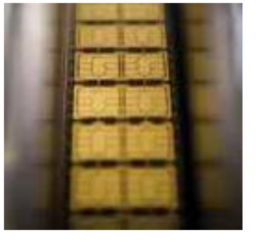
Technological foundation

Carte à puce et Java Card

2010-2011

Jean-Louis Lanet

Jean-louis.lanet@unilim.fr



Université de Limoges

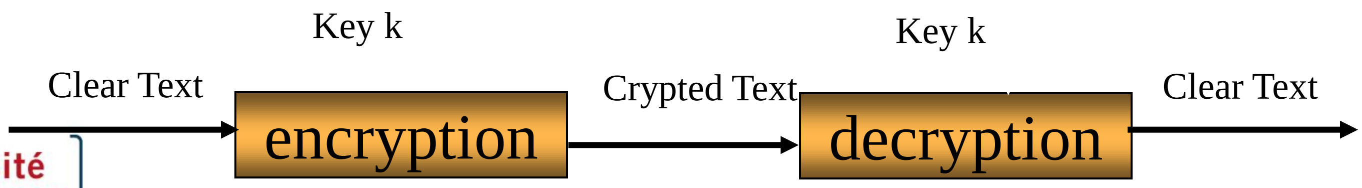
SUFOP Service Universitaire de Formation Permanente

Agenda

- Cryptology
- Authentication
- Secure upload

Cryptology

- Cryptography / Cryptanalysis,
- Smart Cards considered as Authentication media and Encryption module,
- Expected properties : confidentiality, integrity, authenticity and binding ness,
- Modern crypto is not based on obscurity, only the key is confidential.
  - $E_k(M) = C$
- Cryptanalysis = exhaustive search of the secret key
- The key space as high as possible 256 bits key = 2^{256} possibilities
- One way function: encryption and decryption must be effective in polynomial time but but impossible without the key.



Université de Limoges

SUFOP Service Universitaire de Formation Permanente

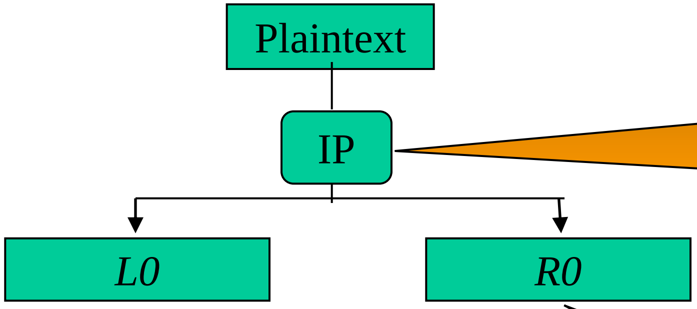
Cryptographic techniques used in smart card

- Techniques
  - Functions
    - Rnd. Generator
    - Block enc. mod.
    - Hash function
    - Key generator
    - One way func.
  - Algorithms
    - Symmetric
    - Asymmetric
  - Protocols
    - -Challenge Response
    - -Diffie Hellman
    - -Fiat Shamir
    - -Zero knowledge
    - -Blind signature

DES

- Symmetric => encryption and decryption share the same secret,
- Characteristics
  - Input = 64 bits block
  - Output = 64 bits block
  - Key = 64 bits but only 56 are used (the 8th bit of each byte serves as a parity bit)
  - Symmetric algorithm : E = D, $K_E = K_D$
  - Operations : substitutions, permutations and XOR at each round
- Exhaustive key search is possible :
  - Computer machine can find a key in some hours.

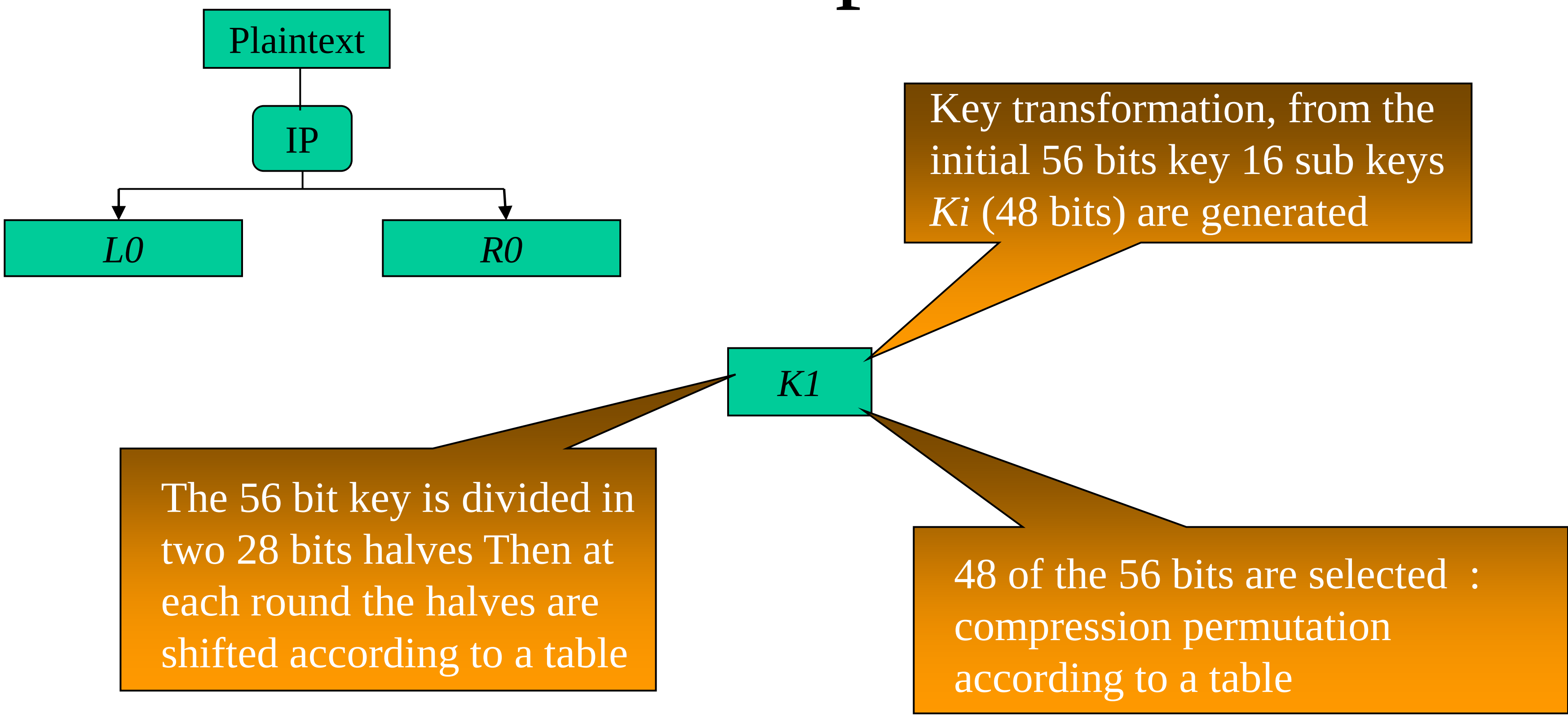
DES descri



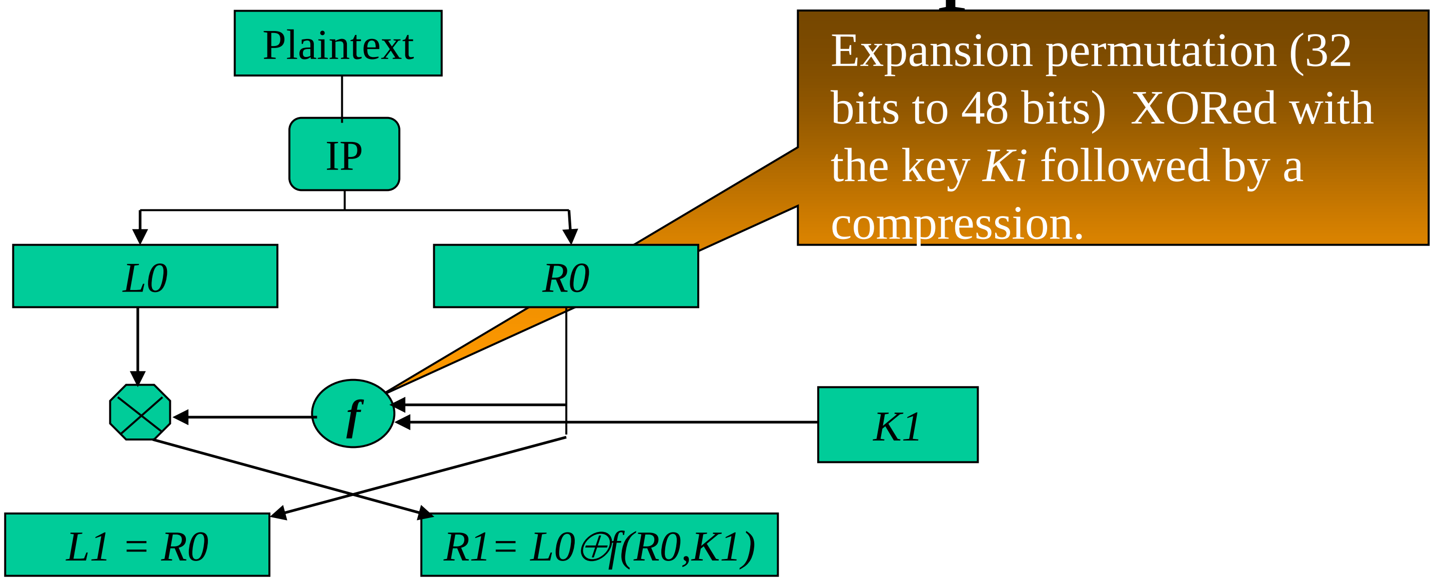
Initial permutation : the plain text is broken in two 32 bits half according to a table (bit 58 to bit 1, bit 50 to bit 2…)

16 rounds noted 0 to 15, all identicals

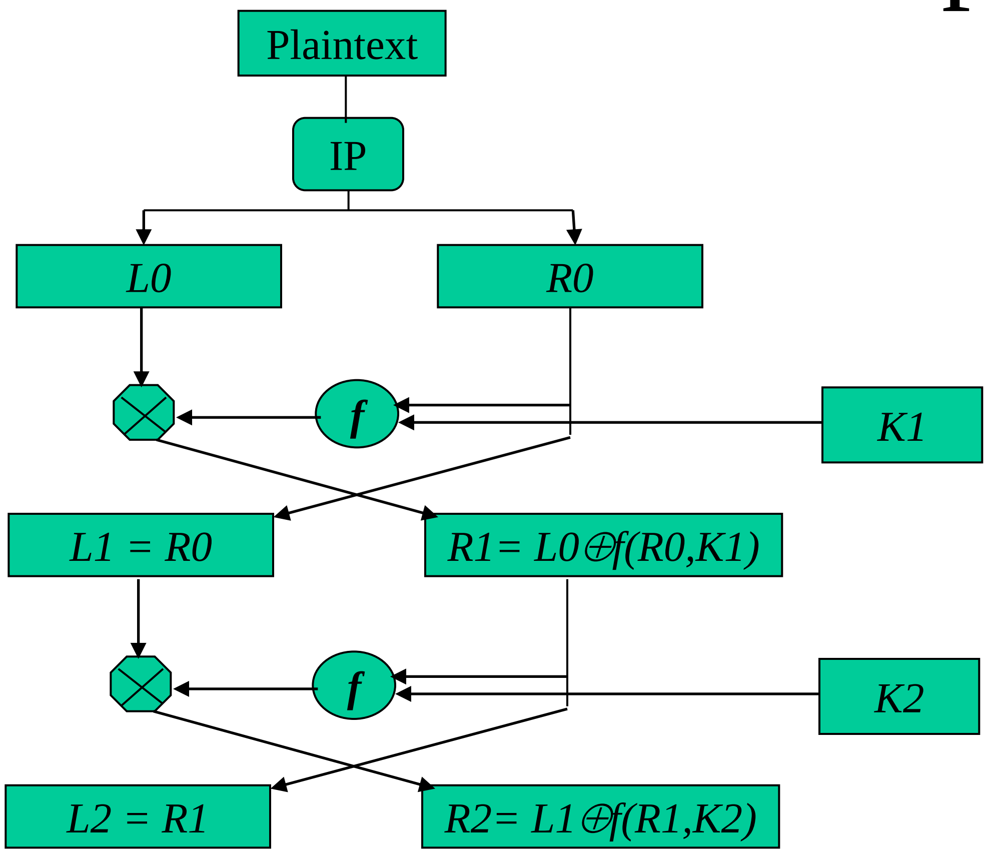
DES description



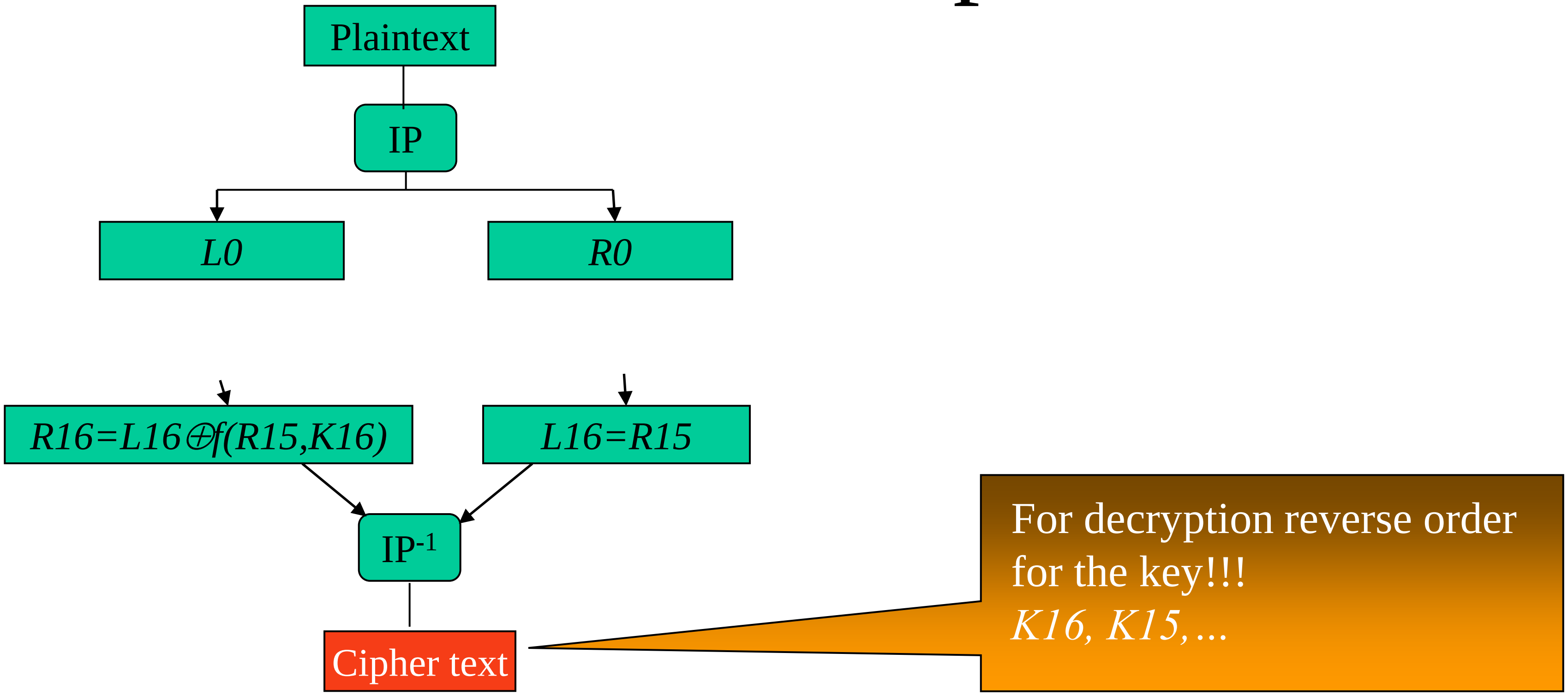
DES description



DES description



DES description

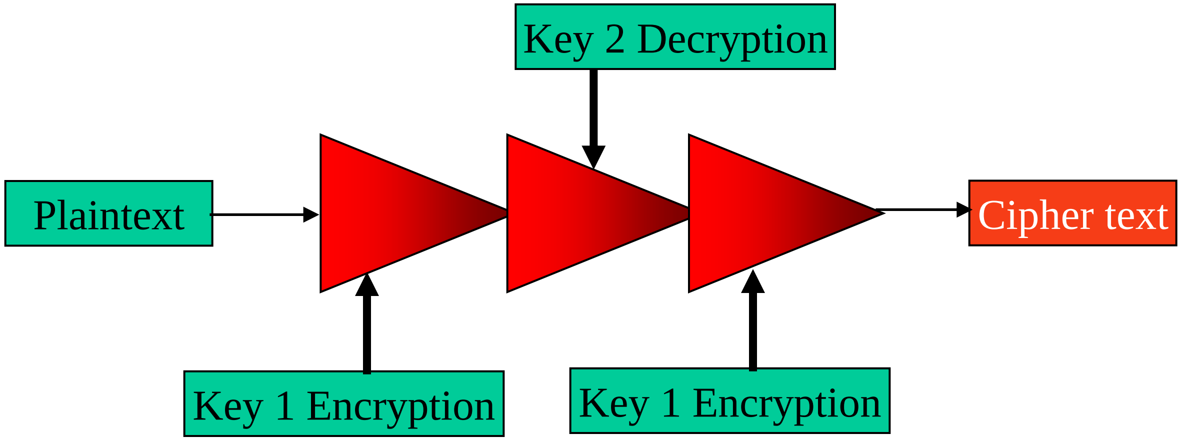


Practical issues

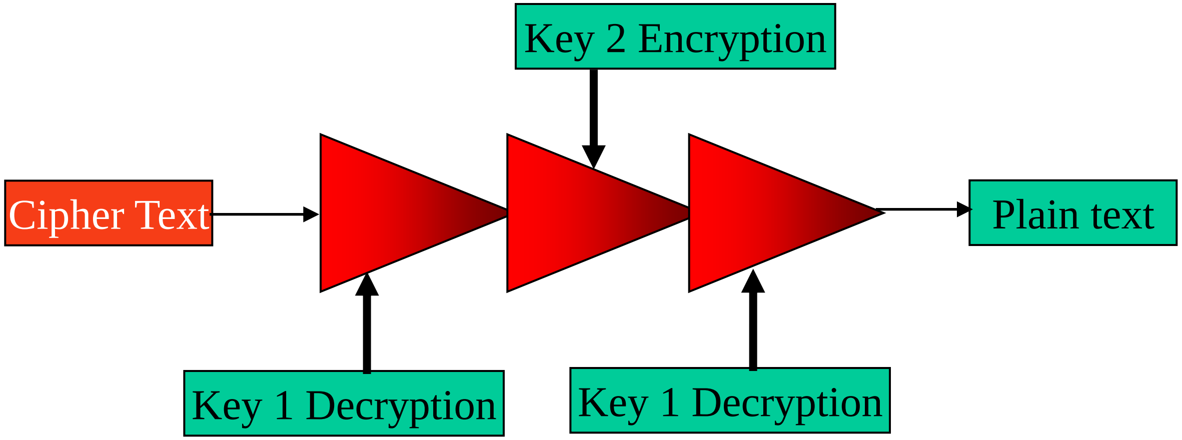
- Due to the parity bit the key space is 2^{56},
- If a pair (plaintext, cipher text) is obtained from a terminal it costs around 64 hours to find the correct key,
- No smart card microcontroller with a hardware DES module,
- It costs 1KB of code and 256 byte of RAM and with a 3,5 Mhz clock runs in 17 ms per 8 byte block.

Triple DES

- Chaining of 3 DES
  - Larger key size: 2x56 = 112 bit-key, often key1=key3 but it remains possible to use 3 keys 3x56=168 bits.
  - Data compatibility with DES, with no additional cost
  - Input = 64 bit block
  - Output = 64 bit block



Triple DES



- How do I distribute and store my keys ?
- Do not reuse the same key for several sessions,
- Symmetric algorithms are more resistant to cryptanalysis.

Asymmetric algorithms

- Before using a secret key encryption algorithm,
  - How do the principals get the key ?
  - Notion of a Public Key Cryptosystem : Diffie, Hellman(1976).
  - First application : RSA (1977).
- Other public key cryptosystems have emerged since ; their security relies on some hard mathematical problem.
- Idea : find a system where D_k is very difficult to compute from E_k then E_k can be made public.
- Public Key algorithms = asymmetric algorithms
  - Encryption key : Bob's public key (in the yellow pages)
  - Decryption key : Bob's private key (secret).
  - Two key pairs are necessary for a dialogue

RSA

- Based on the arithmetic of large integers
  - Easy to compute the public modulus of the two prime numbers,
  - Very difficult to decompose the modulus into two prime factors, no effective algorithm for this operation
- Key are generated from two large prime numbers,
  - Encryption : $y = x^e \textbf{ mod } n$
  - Decryption : $x = y^d \textbf{ mod } n$
  - Where
    - x = clear text, y encoded text,
    - e public key, d private key,
    - n public modulus, p and q secret prime number; $n = p \bullet q$

Simple example

1. Select two prime numbers p and q,
2. Calculate the public modulus
3. Calculate the temporary variable z
4. Calculate a public key *e* that satisfies the conditions *e*<z and **gcd** (*z*,*e*) = 1. There are several numbers that meets these condition, select one
5. Calculate a private key d that satisfies the condition (*d*•*e*)mod z=1
6. Discard p and q, publish e and n

- p=47; q=71
- n= p•q =3337
- z= (p-1) • (q-1)=3220
- e = 79
- d=1019

Université de Limoges
SUFOP Service Universitaire de Formation Permanente

Encryption

- To encrypt the message x = 6882326879666683
  - Break into blocks e.g. three-digit block
    - M1 = 688, M2 =232, M3=687, M4=966, M5 =668, M6= 003 Padding !!!
  - The first block is encrypted as
    - $y1 = 688^{79}$ **mod** 3337=1570
  - On all the blocks:
    - y = 1570 2756 2091 2276 2423 158
  - Decrypting the first block using the private key 1019
    - $x1= 1570^{1019}$ **mod** 3337 = 688

Université de Limoges

SUFOP Service Universitaire de Formation Permanente

Issues with asymmetric algorithm

- Some structure of the plaintext should remain in the ciphered text, more sensible to cryptanalysis.
- Specific to smart cards:
  - Relatively slow technology, difficult to encrypt huge messages,
  - Generation of p, q and e => random number generator, primarily test (Eratosthenes can't be used, storage of prior prime number).
  - Fast exponentiation algorithms (encryption an decryption)
  - Ram usage: $(x \bullet x)$ **mod** n is never evaluated directly but: $((x$ **mod** $n) \bullet (x$ **mod** $n))$ **mod** n.
  - Public and private keys may have different length,
    - Private key as large as possible to avoid to break the code,
- Public key as short as possible (time required to verify a digital signature)
    - 512 bits is the lower limit, 2048 bits is the upper bound for SC

Issues

- Smart cards:
  –

Implementation	Mode	512 bits	1024 bits
SC no hw	Signing	20 min	
SC w hw	Signing	308 ms	2000 ms
SC no hw	Key gen	3s to 40s	10s to 180s

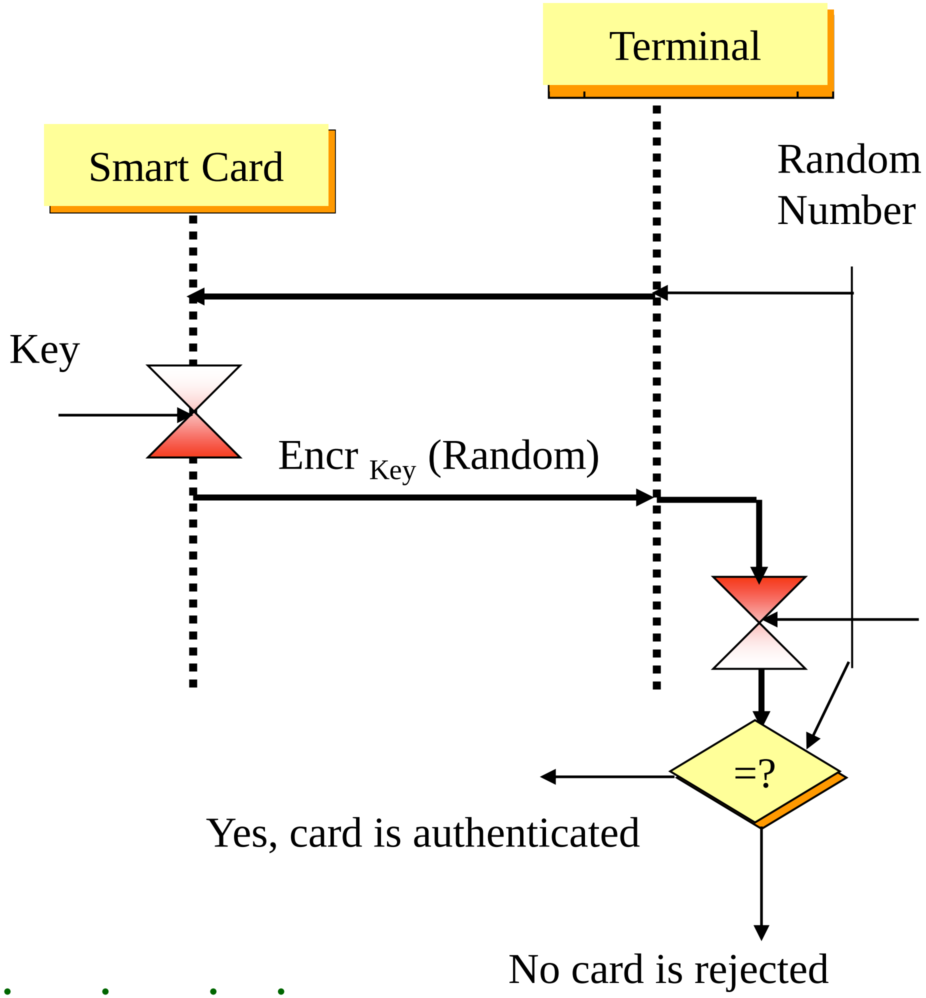
  – Code size with hardware-supported (fast exponentiation) 512 bits RSA is around 300 bytes and 1 k bytes for 1024 key.
  – RSA algorithm very secure but rarely used to encrypt data
  – Import export restriction, patent issues.

Agenda

- Cryptology
- Authentication
- Secure upload

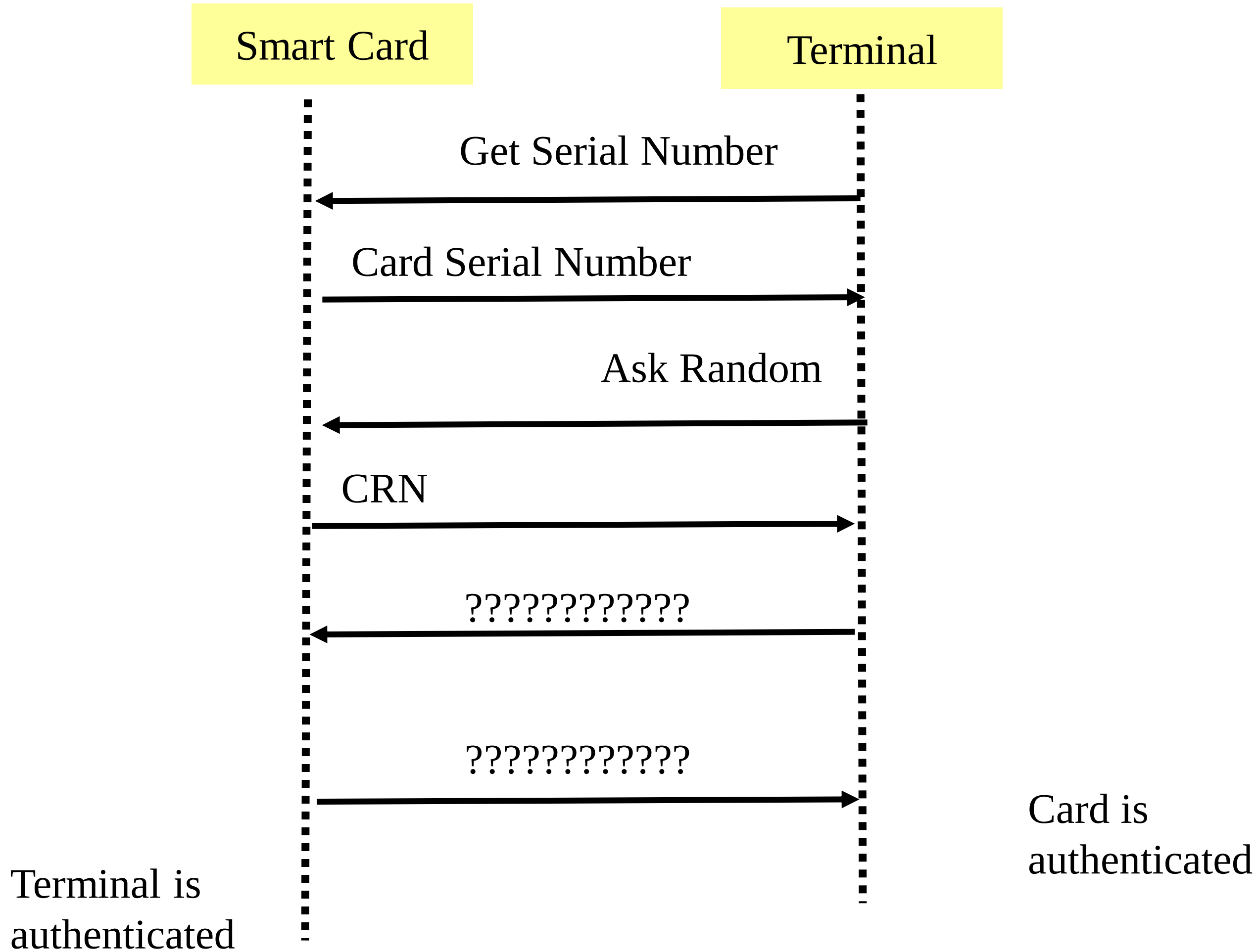
Unilateral authentication

- Random=> no replay
- Attack= pairs of plain ciphered text
- Used of a derived key…
- DES or triple-DES can be used for encryption
- Use of command "internal authenticate" cf. ISO7616-4



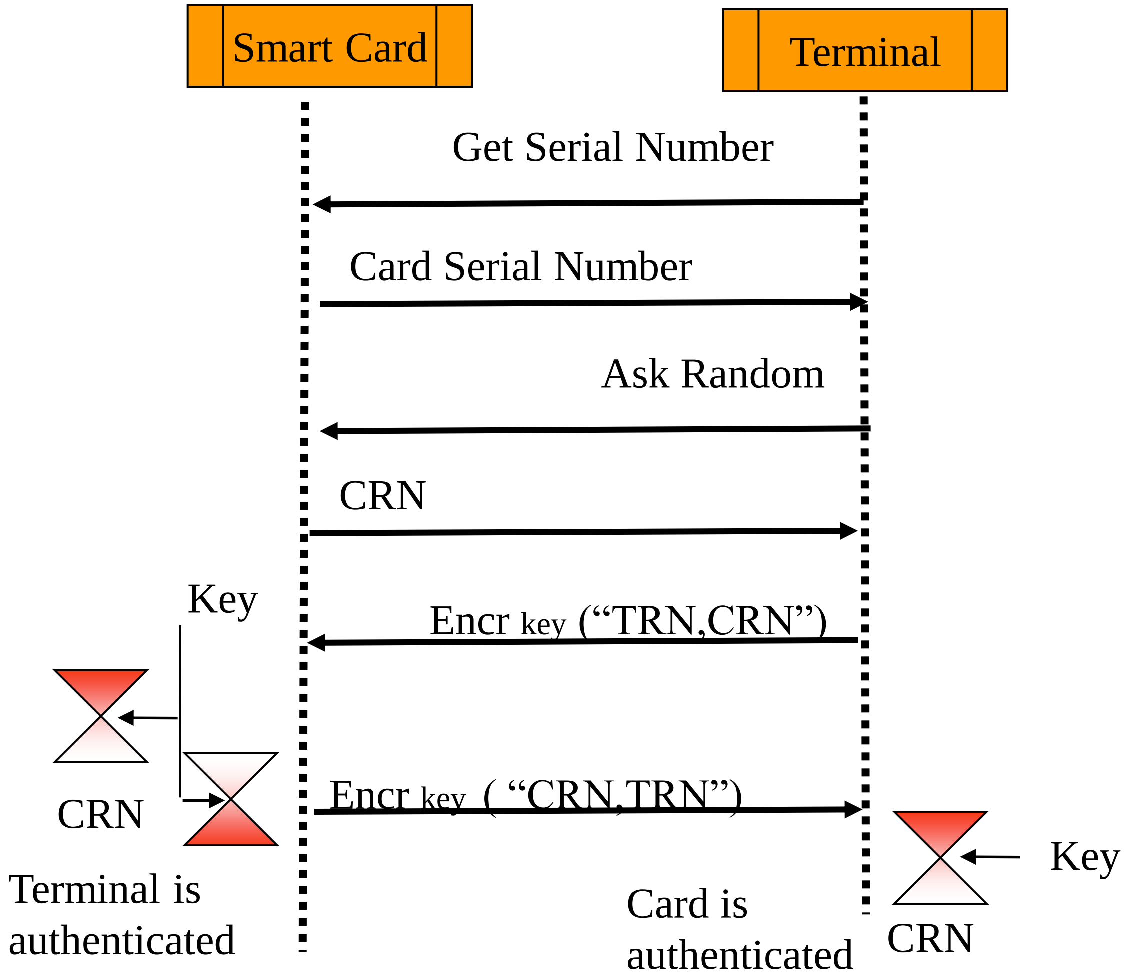
Something is missing …

Mutual Authentication



Université de Limoges
SUFOP Service Universitaire de Formation Permanente

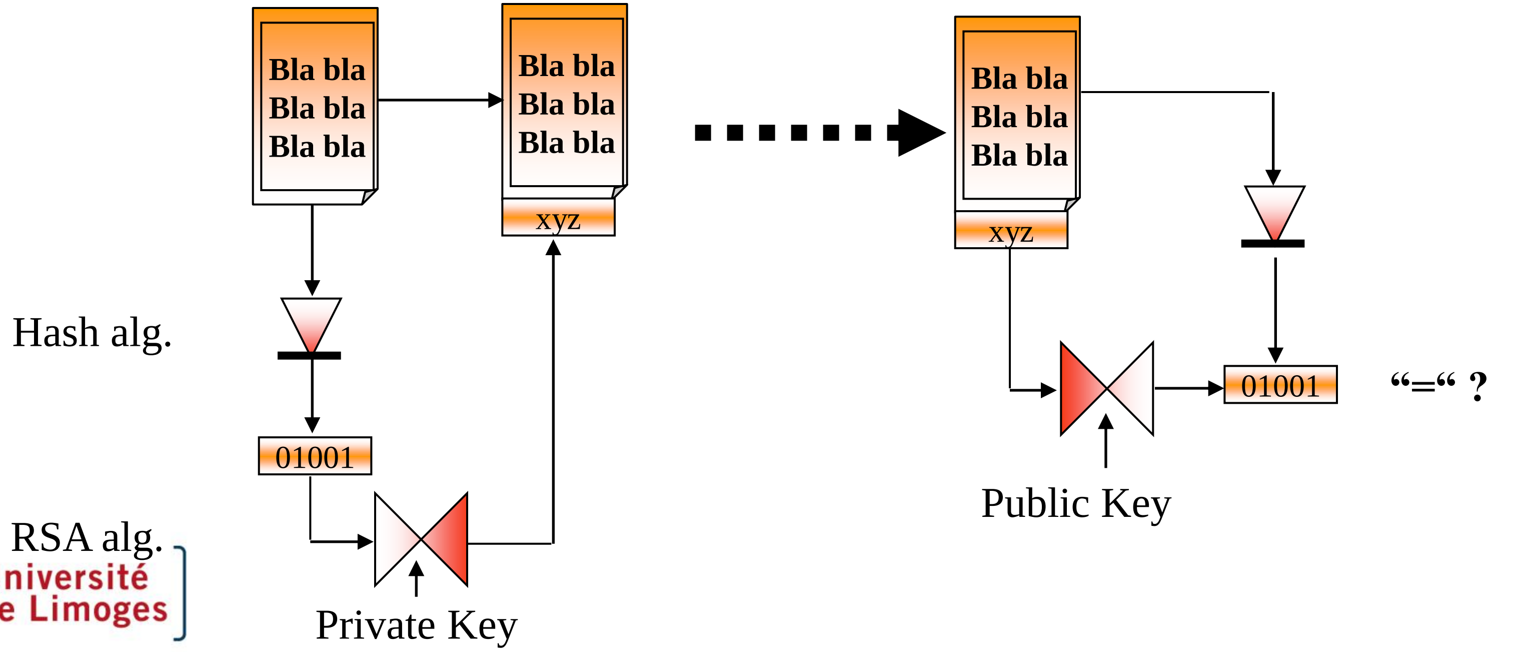
Mutual Authentication



Université de Limoges
SUFOP Service Universitaire de Formation Permanente

Signature

- Authenticity of electronically transmitted documents,
- Produced by one single individual, verified by anyone => asymmetric algorithms,
- Not computed on the entire data but on a hash value using a one way compression function,



Université de Limoges

SUFOP Service Universitaire de Formation Permanente

Agenda

- Cryptology
- Authentication
- Secure upload

Global Platform

- The Global Platform provides specifications to define security policies and cryptographic mechanisms to protect download and delete of applications.
- Each applet can be securely loaded and removed using either Public Key or Symmetric key cryptography.
- Need to embed the Card Manager as an applet or as native code
- Need to implement the GP API
  - Services: Cardholder verification, personalization, security services,…
  - Card Content management services : card locking, Application life cycle state update,…

GP Card Domain

- Issuer representative
- Provides card global services:
  - installation of applets on the card
  - management of the applet life cycle
  - personalization and reading card global data (such ICC serial number)
  - management of the card life cycle
  - blocking card service
  - auditing services when the card is blocked
- Acts as the security domain for the issuer's applets

GP functionalities

- Applet & life cycle management,
  - Need authentication and integrity for :
    - Load, Install and Make Selectable
    - Delete,
    - Set Status.
- Secure communication protocol,
  - Entity authentication,
    - Current Security level = `Authenticated` (SCP01, SCP02) `Any_Authenticated` (SCP10)
  - Confidentiality and/or Integrity and authentication,
    - Integrity or Confidentiality and Integrity of a command sent to the card,
    - Integrity of the sequence of APDU command sent to the card

SCP 01

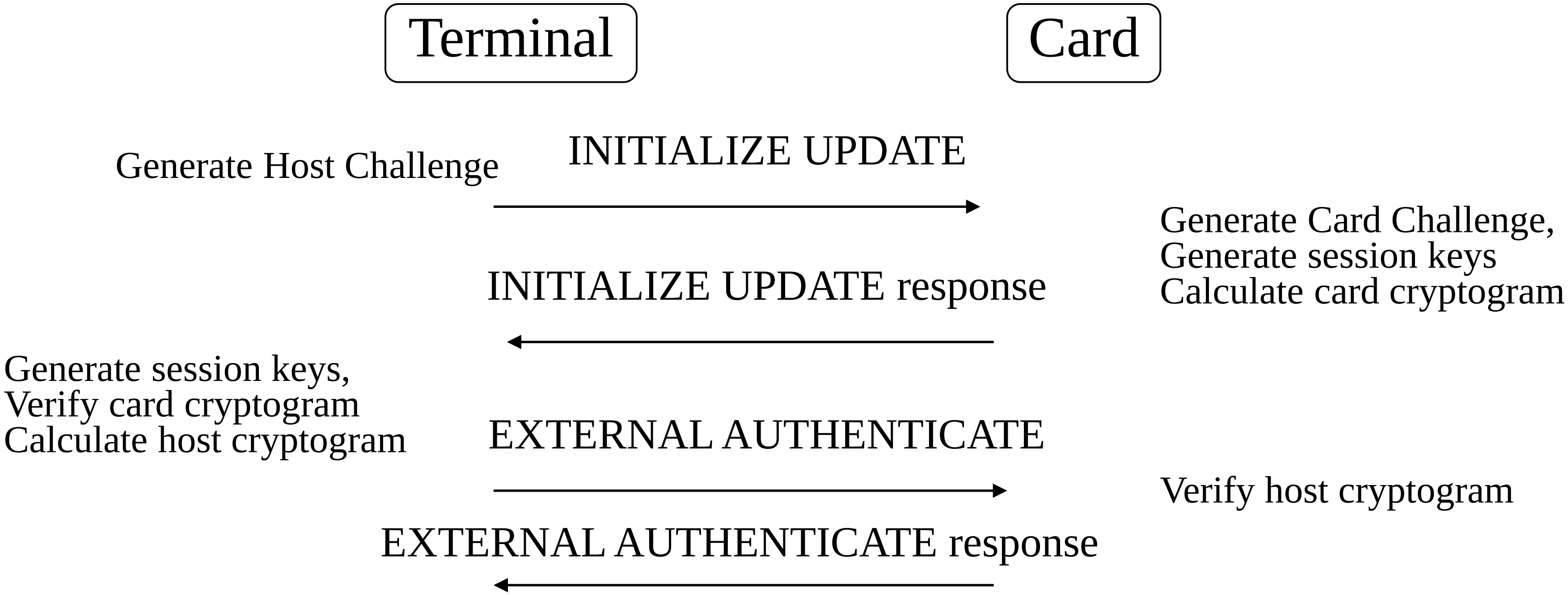
- This protocol modifies APDUs, using some pre-established symmetric keys on both sides, to secure the original APDUs with MAC checks and optional encryption.
- Symmetric key, SCP01 is deprecated, backward compatibility with GP 2.0.1
  - Replaced by SCP 02 symmetric key protocol,
  - Three levels of security
    - Mutual authentication
    - Integrity and data origin authentication
    - **Confidentiality** : **in which data being transmitted from the off-card entity to the card, is not viewable by an unauthorized entity**

SCP01 versus SCP02

- SCP02:
  - **Confidentiality : in which data being transmitted from the sending entity (the off-card entity or card) to the receiving entity (respectively the card or off-card entity) is not viewable by an unauthorized entity**.
- For SCP01, data from host to card is not susceptible to sniffing but no mention of the reverse to be true. For SCP02, both directions are not susceptible to sniffing.
- SCP01 supports mutual auth while for SCP02, only the card authenticates the host, with an option for the reverse.
- There is no encryption from the card side. Be aware that R-MAC is optional, depending on the security policy of the issuer.
- Another differences between SCP01 and SCP02:
  - The DEK in SCP02 is a session key, and in SCP01 it is static,
  - The INITIALIZE UPDATE command is different regarding the P2 parameter and the structure of the response

Mutual Authentication

- It provides assurance to the card and the terminal they are communicating with authenticated entity.



Initialize Update

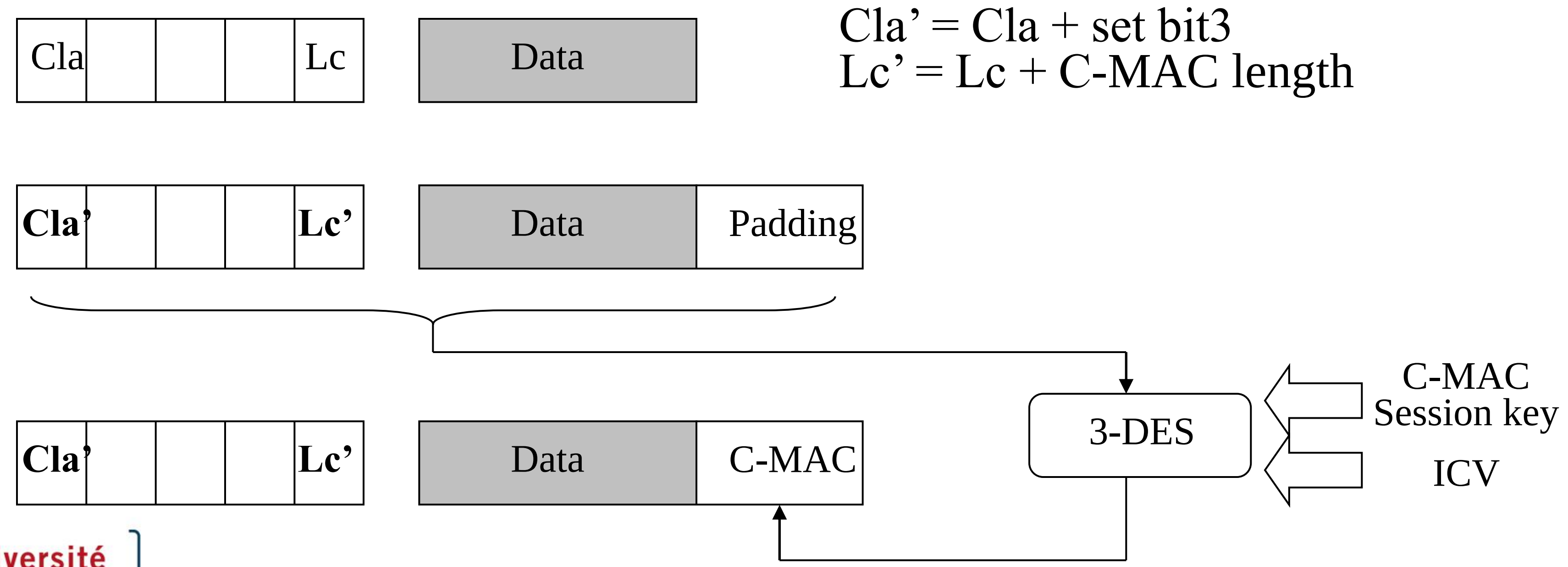
- APDU INIT_UPDATE
  - P1 key version number
  - P2 key set index
  - Data : host challenge
- RESPONSE
  - Card cryptogram + Card Challenge
  - Or 0x6A88 Referenced data not found

External Authenticate

- APDU EXTERNAL_AUTHENTICATE
  - P1 Security Level
    - 0x00 No Secure messaging
    - 0x01 C-MAC
    - 0x03 C-DECRYPTION and C-MAC
  - Response:
    - DATA Host Cryptogram and MAC
    - Or 0x6300 Authentication failed

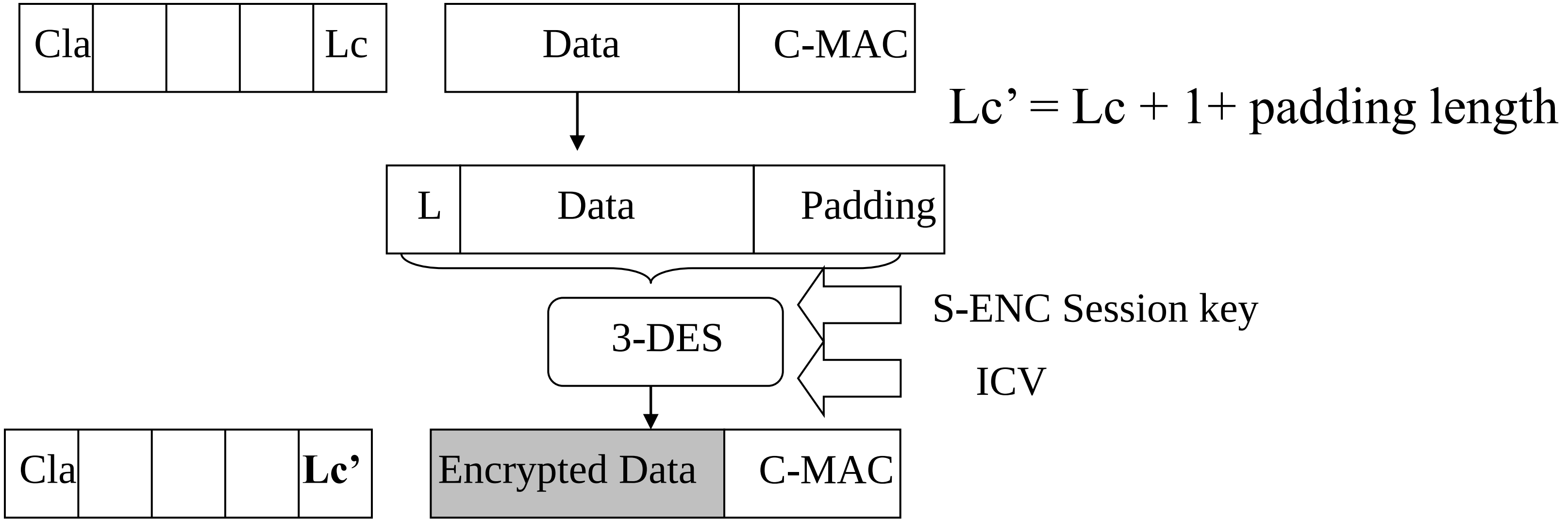
APDU Command MAC generation

- A C-MAC is generated by an off-card entity and applied across the full APDU command being transmitted to the card including the header (5 bytes) and the data field in the command message



APDU data field encryption

- If confidentiality is required, the off-card entity encrypts the “clear text” data field of the command message being transmitted to the card.



The Cryptographic Keys

- S-ENC, Secure Channel Encryption Key
  - A static key to generate a session key: 16 bytes
  - Used to Authentication and encryption (DES)
- S-MAC, Secure Channel Message Authentication Code Key,
  - A static key used to generate a session key: 16 bytes,
  - Used to MAC verification (DES),
- DEK, Data Encryption Key
  - Used as a static key, 16 bytes,
  - For decrypting sensitive data (e.g. secret key)

SCP 01 drawbacks

- The main drawback of SCP01 is the lack of protection of the card response : no MAC no sequence number.
- SCP02 includes a sequence number and a complete response including a R-MAC.
- Expected weakness scenario:
  - There is no proof that a transaction finished correctly,
  - E.g : while loading an applet, an attacker can modify the Load-Install-MakeSelectable response (9000 -> 6xxx).

Any question ?