Overview (1)

- Introduction
- Goal
- Cryptography Overview
  - Symmetric crytography
    - Modes of encryption
      - ECB: Electronic Code Book
      - OFB: Output Feed Back
      - CFB Cipher Feed Back
      - CBC: Cipher Block Chaining
      - Padding before encryption
    - Secret Key $\leftrightarrow$ Public Key

Overview (2)

- Key-Agreement
  - Diffie-Hellman Key-Agreement Protocol
  - One Way Functions
  - Modular Exponentiation

– Secret Key derivation

– Public Key Cryptography
  - Secret Key ↔ Public Key
  - RSA Public Key Algorithm

– Data Authentication
  - Hash Functions
    - MDC: Hash function without key
    - MAC: Hash function with Secret Key
    - retail MAC: MAC based on Block Cipher

Overview (3)

    - Digital Signature
        - Digital Signature with RSA
    - Station to Station Protocol
- Protocol Details: Key agreement
- Protocol Details: Data Broadcast
- Expectations
- References

Goal

- Setup a secure bidirectional communication channel between A and B
- Secure: data integrity and confidentiality are guaranteed
- Confidentiality: Encryption (AES or DES in CBC mode)
- Integrity: retail-MAC based on the AES or DES
- Secret keys (shared keys) negotiated through key agreement
- MAC and encryption keys are derived from the shared secret (after application of hash function SHA-1)
- Digital signatures for the authenticated key agreement use RSA or ECC

Protocol Details: Key agreement

A

- **1:** Send PING
  - **get input:** x, $Priv_1$, Pub_1, Pub_2
  - **compute msg:** $\alpha^x \bmod p$ or xP
  - **send msg**
- **4:** Receive PONG
  - **get msg**
  - **check signature**
  - **compute shared secret:** $(\alpha^y)^x \bmod p$ or xyP
  - **derive secret keys**

B

- **2:** Receive PING
  - **get msg** ($\alpha^x \bmod p$ or xP)
- **3:** Send PONG
  - **get input:** y, $Priv_2$, Pub_1, Pub_2
  - **compute shared secret:** $(\alpha^x)^y \bmod p$ or yxP
  - **derive secret keys**
  - **compute msg:** $E_{ENC}(S_{Priv2}(\alpha^y,\alpha^x))||\alpha^y$ or $E_{ENC}(S_{Priv2}(yP,xP))||yP$
  - **send msg**

Protocol Details: First Data Cell of the Data Broadcast

- First data message sent from A to B contains in the data field:
  - $E_{ENC}(Data||\mathbf{S_{Priv1}}(\boldsymbol{\alpha}^{\mathbf{x}},\boldsymbol{\alpha}^{\mathbf{y}}))$ or $E_{ENC}(Data||\mathbf{S_{Priv1}}(\mathbf{xP},\mathbf{yP}))$
- Next data messages will only contain $E_{ENC}(Data)$ in that field.

Protocol Details: Data Broadcast

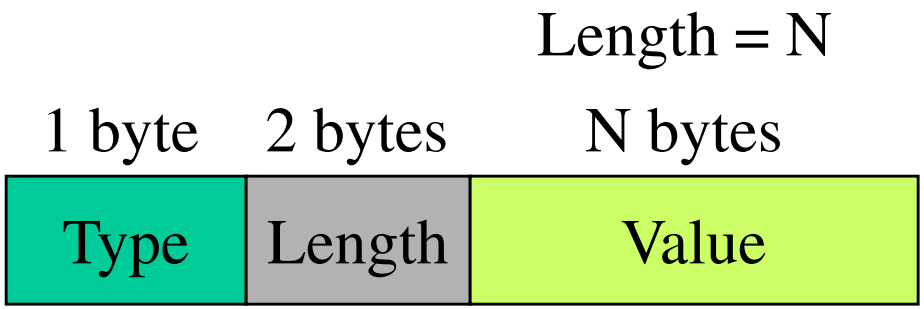
A B

- **5:** Get data
  - **pad the data**
  - **encrypt the padded data**
  - **compute MAC on the encrypted data**
- **6:** Send data to B

- **7:** Receive data from A
  - **check integrity (MAC)**
  - **decrypt data**
  - **strip padding**

Implementation Hints: Suggested message format

- 4 types of messages
- Structure TLV (Type, Length, Value)



Key agreement: shared secret

- $sec_{DH} = \alpha^{xy}$ *mod* p or xyP
- 2 secret keys are derived from the shared secret [H(•) = SHA-1]:
  - Encryption key (confidentiality protection):
    - $sec_{ENC} = H(sec_{DH} || 0x0F) \rightarrow E_{ENC}(\bullet)$
  - MAC-Key (integrity protection):
    - $sec_{MAC} = H(sec_{DH} || 0xF0) \rightarrow M_{MAC}(\bullet)$
    - MAC is computed on the encrypted information

Secure broadcast: A

- A: produce *Padded* = Data || padding
- A: compute the ciphertext: $E(sec_{ENC})(Padded)$
- A: compute the MAC (using sec_{MAC}) on the ciphertext
- A: broadcast info to B

Secure broadcast: B

- When B receives information:
  - Retrieve the MAC-Key
  - Compute the MAC on the incoming info
  - Compare the computed MAC and the incoming MAC
  - IF both MACs have the same value:
  - B: Decrypt information (using sec_{ENC})
  - B: Validate padded information
  - IF ok, info ready for further processing

Expectations

- IMPLEMENTATION of:
  - AES or DES
  - SHA-1
  - **Protocols**: **key agreement** and **data broadcast**
    - **Retail MAC** based on AES or DES
    - **General modular exponentiation** or point multiplication (Diffie-Hellman and RSA or ECC)
    - **Padding** (adding and stripping)
    - **Encryption/Decryption** in **CBC mode**