Identification, authentication, authorization

1 Access Control

Access to the resources:

- Claim the identity
- Verify the credentials
- Check permissions
- Grant access

2 Identification

Definition: Subject

An active entity within a system (physical person, script, etc)

Definition: Principal

An entity that can be granted access (represented by a username, userid, pin etc)

The subject identifies itself to the system as a principal

3 Authentication

The system verifies the identity of the user

Definition: Object

Resource that some principals may access or use

The system checks that the principal has the permissions to access an object

4 Credentials

- What do you know? Passwords, PINs
- What do you have? Authentication key, passport, ticket, mobile phone
- Who you are? Biometrics

5 Passwords

Common Security Guidelines:

- Adopt long passphrases
- Avoid easy to guess passwords
- Use combination of a-z, A-Z, 0-9 and symbols
- Do not write down passwords
- Avoid using the same password for multiple services

However - when internet users log on to as many as 25 password-protected sites per day, remembering a different and secure password for each one is very difficult.
Passwords should be stored in a password manager so you only have to remember one secure password

6 Authentication Keys

Authentication keys (e.g. SSH keys)

- Similar to passwords, but
- RSA-based keys
- Subject create private/public key
- Share the public key with services
- Per device RSA key

Advantages:

- Public key leak are inconsequential
- Compromised device access can be revoked

7 Security Keys

Authentication keys weakness: Compromised client

Solution: Physical security keys:

- Static password token (not recommended)
- Asynchronous tokens (one-time passwords)
- Challenge-response tokens

8 Biometrics

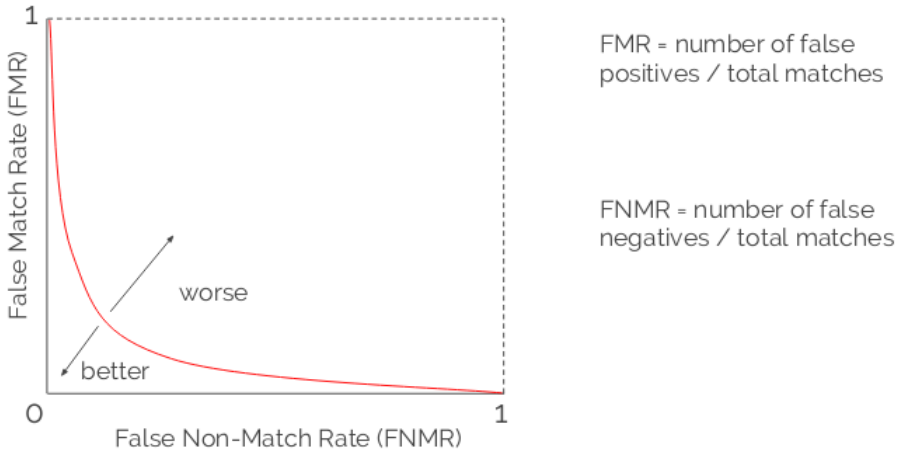
Advantages:

- Non-repudiation: a way to guarantee that an individual who accesses a certain facility cannot later deny using it

Disadvantages:

- Uncertainty: Compromise between false-positive and false-negatives

Receiver Operating Characteristic (ROC) curve



Performance policy

- Prefer low FMR. E.g. automatic border control. Refer to human on negative result
- Prefer low FNMR. E.g. suspect recognition on CCTV. Refer to human on positive result

9 Two-factor authentication

Two-factor authentication.

Combine two authentication factors from:

- What you know: password, pin
- What you have: mobile phone, authentication key

10 Zero-Knowledge Password Proof

Objective: Do not reveal anything in the client/server communications about the password. Otherwise we are vulnerable to replay attacks.

Most common ZKPP approach: Challenge-response auth:

- Serer generate unique challenge value: nonce
- Server send nonce to the client
- Client computer response = hash(nonce + password)
- Client send response back to server
- Server compare the response with hash(nonce+stored password)

Nonce properties:

- Nonce: Random or pseudo-random unique value
- Uniqueness: Prevent replay attacks
- Susceptible to PRNG flaws

11 Example: EMV Payment

- Standard used for all credit cards
- EMV standard: Initially written in 1993
- Over 3600 pages of protocol specification
- Requirements varies from bank to bank

Card authentication mechanism

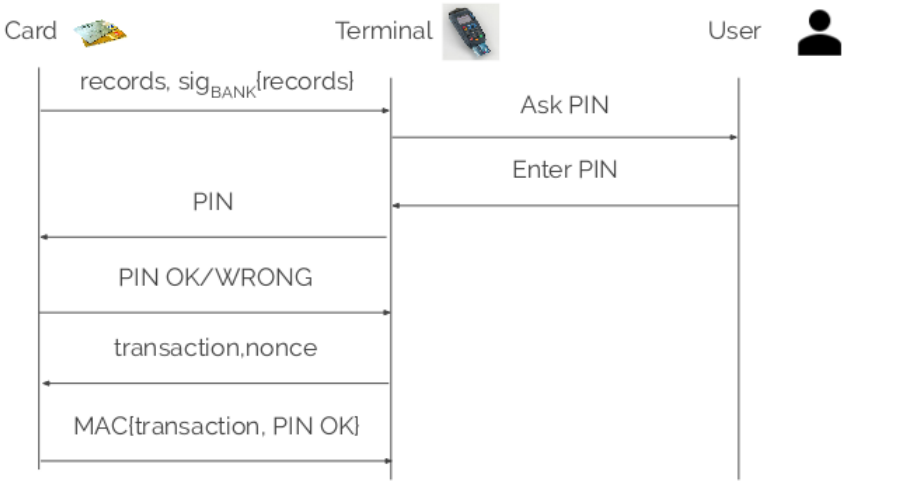
- Static data authentication (offline)
- Dynamic data authentication (offline)
- Combined DDA with application cryptogram generation (offline)
- Cryptogram (online)

Multiple cardholder verification mechanism (CVM)

- No CVM required (e.g motorway toll)
- Signature (common in some countries, e.g. US)
- Offline pin (no internet, pin verified by the card)
- Online PIN (internet, pin verified by the bank)

11.1 SDA: Static Data Authentication

- Offline card payment
- Used by old card and terminal
- Lowest common denominator
- Vulnerable to skimming attack
- During transaction, terminal records the static data
- A cloned card is created with the same static data



* MAC: Message Authentication Code, computed by the card from a unique key

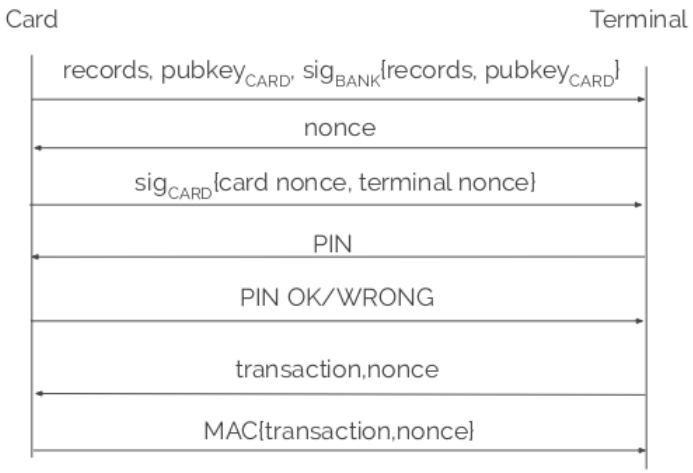
An attacker can get records, $sig_{BANK}[records]$ by listening to a valid transaction

The the attacker can create a fake card using $sig_{BANK}[records]$ and generate an invalid MAC. For offline transaction, the merchant cannot verify the MAC anyway. By the time the merchant sends the transactions to the bank, the attacker will be long gone

Problem: static password

11.2 DDA: Dynamic Data Authentication

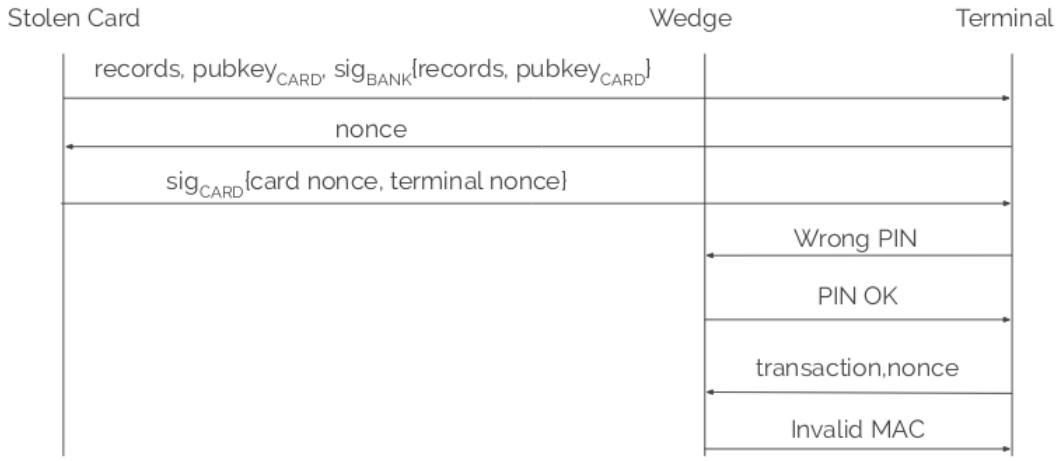
Use challenge-response authentication to generate data unique to the transaction



Card clone is not possible because sig_{card}[card nonce, terminal nonce] is different at every transaction.

However, card answer to PIN check is not authenticated either

A wedge between a stolen card and a terminal can pretend that the password is always correct

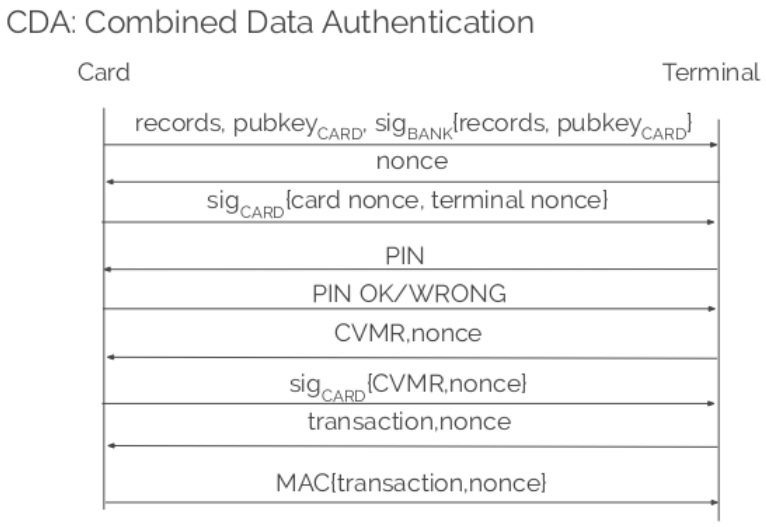


* MAC: Message Authentication Code, computed by the card from a unique key

11.3 CDA: Combined DDA/Application Cryptogram Generation

Solution: Second card authentication step after PIN check

The terminal sends a message called CVMR representing the terminal view of the operation (PIN OK, PIN Wrong, signature etc) for the card to compare with its own point of view



* MAC: Message Authentication Code, computed by the card from a unique key

Takeaway:

- Do not send static auth data (e.g. unencrypted passwords)
- Use challenge-response to prevent replay attacks
- Make sure that authentication is verified at all steps