Chapter 5
Authentication Mechanisms

Hashing vs encryption

Hashing: The process of converting plaintext into ciphertext (one-way encryption).

There is no decoding in hashing

the purpose of Hashing protects data in storage.

Encryption : The process of converting plaintext into ciphertext and from cipher to plaintext(two- way encryption) .

the purpose of Encryption protects data during transmission

Storing passwords

A password could be stored in a system as:

- **Plain password**
- **Encrypted password**
- **Hashed password**
- **Salted password**

Hashed password are stored in separate file(shadow password file)from the user IDs

Improved implementations

- Have stronger, hash/salt variants
- Many systems now use MD5

– with 48-bit salt

– password length is unlimited

– is hashed with 1000 times inner loop

– produces 128-bit hash

Password choices/concerns

- users may pick short passwords
  - e.g. 3% were 3 chars or less, easily guessed
  - system can reject choices that are too short
- users may pick guessable passwords
  - so crackers use lists of likely passwords
  - e.g. one study of 14000 encrypted passwords guessed nearly 1/4 of them
  - would take about 1 hour on fastest systems to compute all variants, and only need 1 break!

Using Better Passwords

- Clearly have problems with passwords
- Goal to eliminate guessable passwords
  - Still easy for user to remember
- Techniques
  - user education
  - computer-generated passwords
  - reactive password checking (periodic checking)
  - proactive password checking (at the time of selection)

Proactive Password Checking

- Rule enforcement plus user advice, e.g.
  - 8+ chars, upper/lower/numeric/punctuation
  - may not suffice
- Password cracker
  - list of bad passwords
  - time and space issues
- Markov Model
  - generates guessable passwords
  - hence reject any password it might generate
- Bloom Filter
  - use to build table based on dictionary using hashes
  - check desired password against this table

Improved implementations

- Have stronger, hash/salt variants

- Many systems now use MD5

– with 48-bit salt

– password length is unlimited

– is hashed with 1000 times inner loop

– produces 128-bit hash

Token-based authentication

- Memory Card
- Embossed card

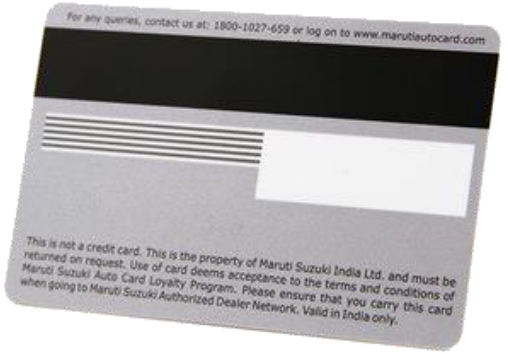
- Smart card
- Magnetic stripe

Magnetic stripe & Memory Card

-Store but do not process data

-Used only for physical access

-some memory card with passwords



Disadvantages :

Need special reader

Loss of token issues

Smart card

-Store and process data

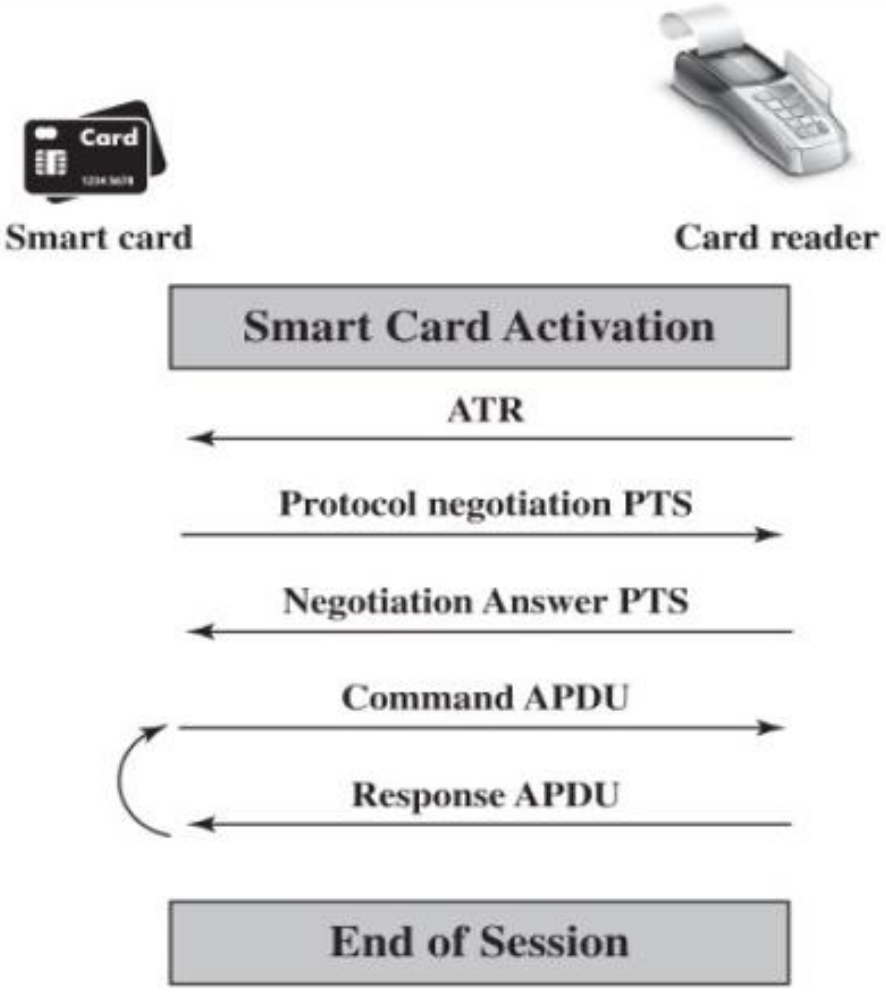
Smart card

Static : uses a specific authenticator .It is called static because the authenticator is reused multiple times and stays the same until you change it

dynamic: passwords created every minute; entered manually by user or electronically.

challenge-response: computer creates a random number; smart card provides its hash

Smart card/reader exchange

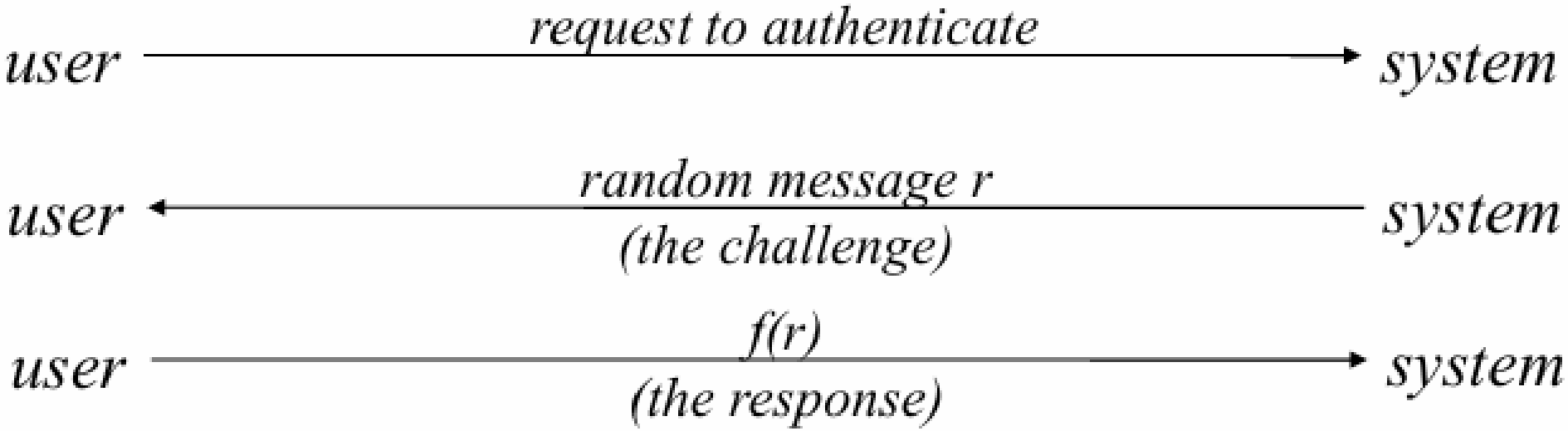


APDU = Application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

Remote User Authentication

- Authentication over network more complex
  - Problems of eavesdropping, replay
- Generally use challenge-response
  - user sends identity
  - host responds with random number *r*
  - user computes *f(r,h(P))* and sends back
  - host compares value from user with own computed value, if match user authenticated
- Protects against a number of attacks

Challenge-Response



Multi – Factor Authentication (MFA)

users provide two types of authentication Such as smartcard with Pin code.

Authentication types :

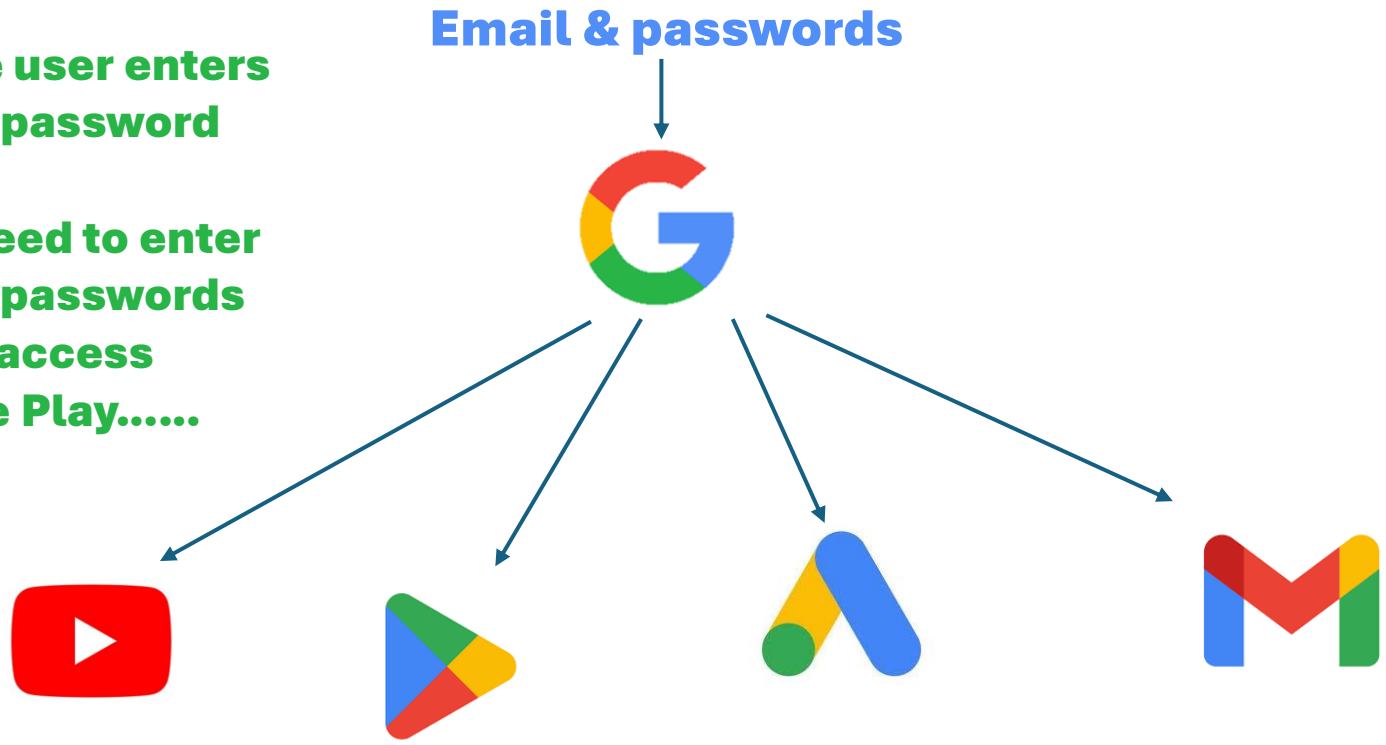
Something you know - something you have – something you are

MFA is more secure than passwords alone but can be more complicated for users

Single Sign-On (SSO)

Single sign-on (SSO) is a technology that allows users to authenticate once and access multiple applications or systems.

In this system, the user enters his username and password only once
The user did not need to enter his username and passwords four times to gain access YouTube & Google Play.....



Single Log-Out (SLO)

single action of signing out terminates access to multiple software systems.

Benefits Single Sign-On (SSO)

Single sign-on (SSO) improve security in our system :

In SSO, the user focuses on creating one strong, random password, the user will put all his effort into making this password strong.

Increased Productivity:

SSO saves time by reducing the number of times users have to enter their credentials.

Simplified Administration:

SSO eliminates the need to manage multiple user accounts and passwords.

Types of Single Sign-On (SSO)

Web-Based SSO:

Authenticates users for web applications.

Enterprise SSO:

Authenticates users for desktop applications and systems.

Federated SSO:

Authenticates users across multiple organizations or domains.

Challenges of SSO

Implementation Complexity:

SSO can be complex to implement, especially for legacy applications or systems.

- **Integration with Legacy Systems:**

Legacy systems may not support SSO, requiring additional work to integrate them.

- **Security Risks:**

SSO can create a single point of failure, making it a prime target for attackers.

Authentication Security Issues

- eavesdropping
- replay
- trojan horse

Authentication Security Issues

- **Eavesdropping**

Eavesdropping: attacker attempts to learn passwords by observing the user, finding written passwords, keylogging

Countermeasures

- **diligence to keep passwords**
- **multifactor authentication**
- **admin revoke compromised passwords**

Authentication Security Issues

- **Replay: attacker repeats a previously captured user response**

– Countermeasure

- **Challenge-response**
- **1-time passcodes**

Authentication Security Issues

- **Trojan horse:**

an application or physical device masquerades as an authentic application or Device

– **Countermeasure:**

authentication of the client

within a trusted security environment

- **Denial of service:**

attacker attempts to disable user authentication service (via flooding)

– **Countermeasure:**

a multifactor authentication with a token

Best Practices for Authentication

- **Use strong passwords and enforce password policies.**
- **Implement MFA whenever possible.**
- **Keep authentication systems up to date with the latest security patches.**
- **Monitor and audit authentication logs regularly.**

Authentication challenges

- User education and adoption
- Interoperability across systems and applications
- Balancing security with user convenience