

Security and Privacy

Antonio Brogi

Department of Computer Science
University of Pisa

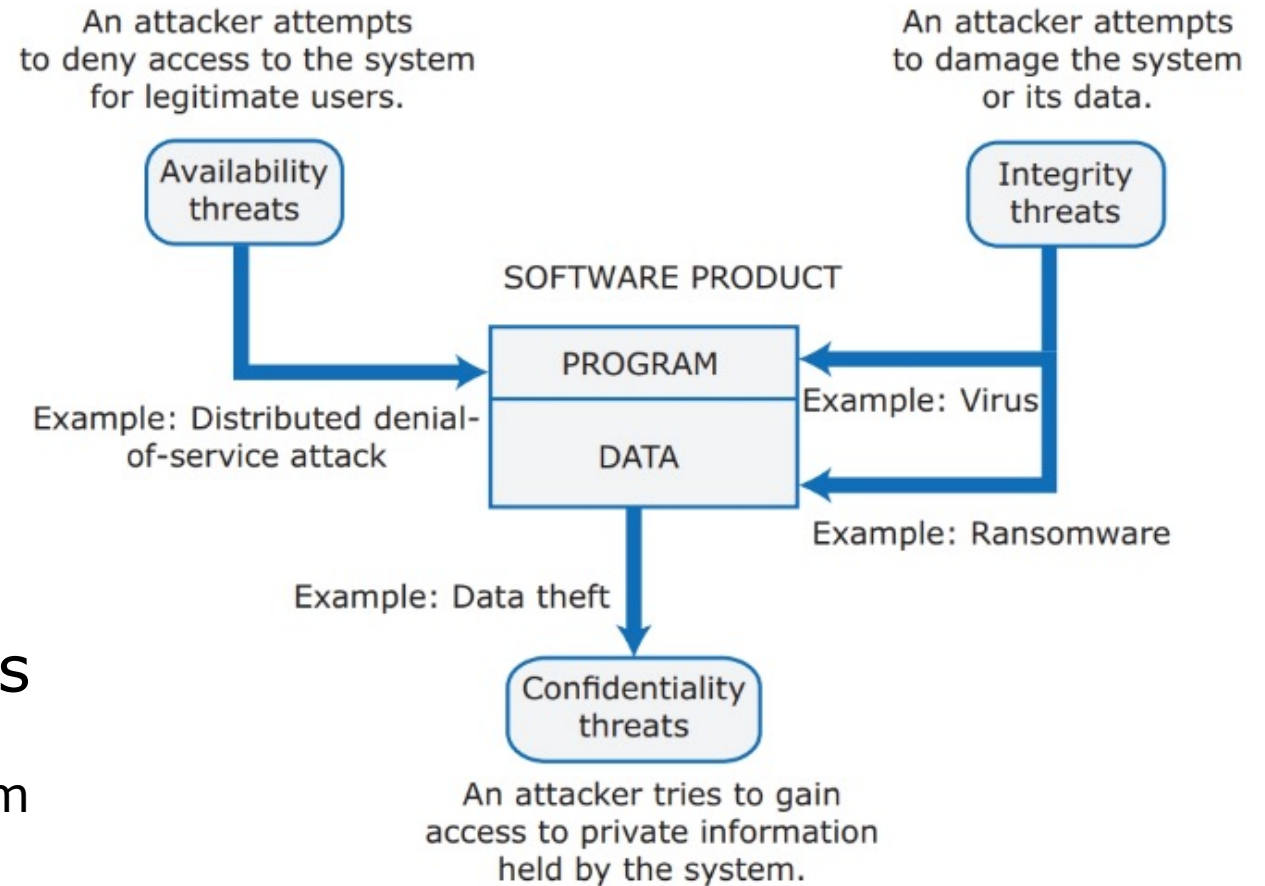


Introduction

Introduction

- Software security: a high priority for product developers and users
 - Malicious attacks can cause losses to both

- Main types of security threats:



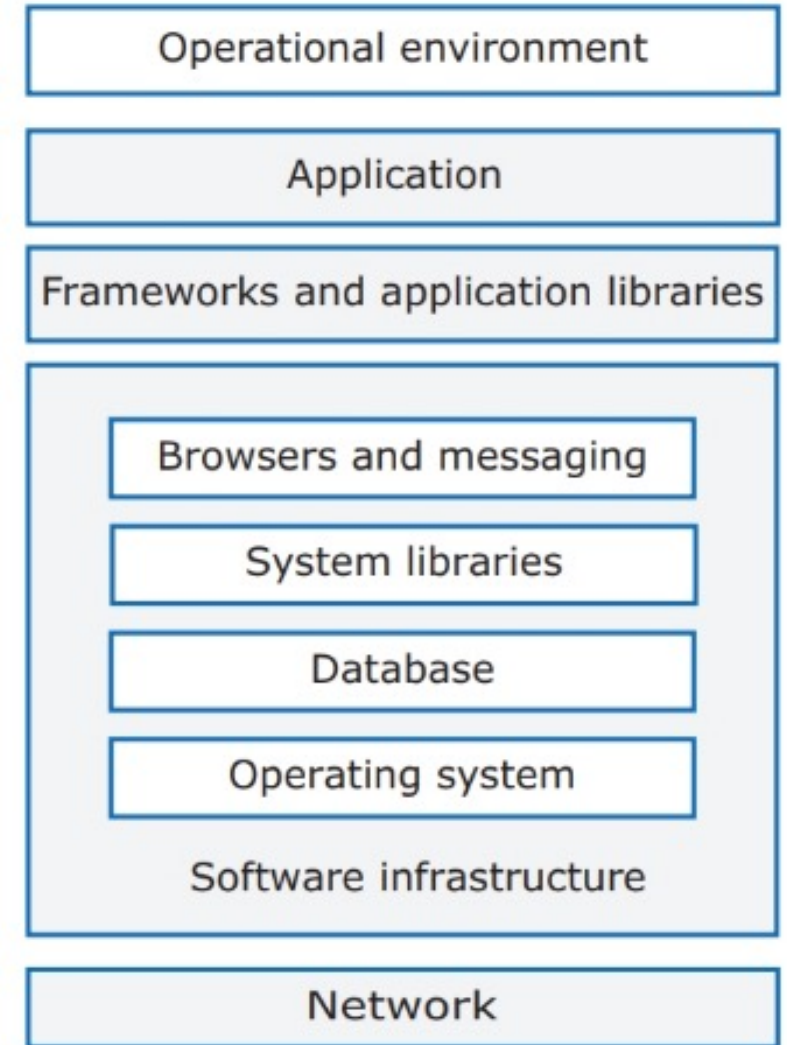
- Some attacks combine these threats

- e.g. ransomware attack threatening the integrity of system data also threatens system availability



Introduction (cont.)

- Security is a system-wide issue
- Application software depends on operating system, web server, language run-time system, database, frameworks and tools ...
- Attacks may target any level of the system infrastructure stack, starting from the network



Introduction (cont.)

System management activities to maintain security:

- ***Authentication and authorization standards and procedures*** to ensure that all users have strong authentication and properly set up access permissions
- ***System infrastructure management*** to keep infrastructure software properly configured and to promptly apply security updates patching vulnerabilities
- Regularly ***monitoring attacks*** to promptly detect them and trigger resistance strategies to minimize effects of attack
- ***Backup policies*** to keep undamaged copies of program and data files that can be restored after an attack

Helping users to maintain security, e.g.

- Multifactor authentication and auto-logout to reduce unauthorized access
- User command logging to help diagnosis and recovery, and to deter

Introduction

Attacks and defenses



Injection attacks

Malicious user uses a valid input field to input malicious code or database commands to damage the system

Buffer overflow attacks

- e.g. on operating systems/libraries written in C/C++, which do not check whether array assignments are within array bounds
- attacker can carefully craft input string that includes executable instructions and overwrites memory
- if a function return address is overwritten, control can be transferred to malicious code



Injection attacks (cont.)

SQL poisoning attacks

- Attacker can do injection attack when user input is part of an SQL command

```
accNum = getAccountNumber ()  
SQLstat = "SELECT * FROM AccountHolders WHERE accountnumber = '"  
+ accNum + "'";" database.execute (SQLstat)
```

Please enter your account number:

'34200645' → SELECT * FROM AccountHolders WHERE accountnumber = '34200645'

'110010010' OR '1'='1' → SELECT * FROM AccountHolders

- Defense: check input validity



Session hijacking attacks

- Session: time period during which user's authentication with a web app is valid
 - Session cookie (token) sent from server to client, client sends session cookie in each HTTP request
 - User doesn't have to re-authenticate for subsequent system interactions
 - Session is closed when user logs out or when system "times out"
- Session hijacking: attacker acquires valid session cookie and impersonates a legitimate user
- Attacker can get session cookie with
 - a cross-site scripting attack (see later)
 - traffic monitoring (easy on unsecured Wi-Fi networks and unencrypted data)



Session hijacking attacks (cont.)

Active vs. passive session hijacking

- Active: attacker carries out user actions on a server
- Passive: attacker simply monitors client-server traffic looking for valuable information (e.g. passwords, credit card numbers)

Defenses

- **Encrypt** client-server network traffic (HTTPS)
- Use **multifactor authentication** to require confirmation of new actions that may be damaging (e.g. require to input code sent to phone)
- Use relatively **short timeouts** on sessions



Cross-site scripting attacks

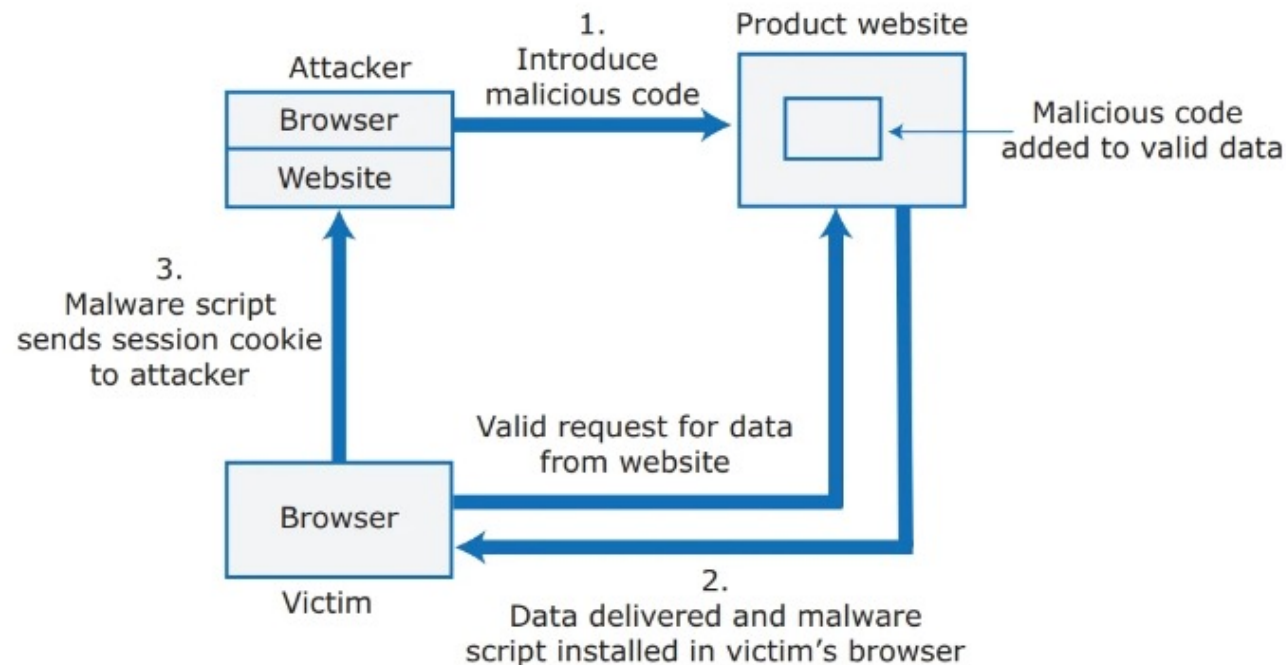
- Another form of injection attack

- Example

- Attacker adds malicious Javascript code to web page returned from server to client
- Malicious script is executed when page is displayed in user's browser
 - Malicious script may steal customer information or direct customers to another website
 - Cookies may be stolen, making a session hijacking attack possible

- Defenses

- (form) input validation
- check input from db before adding it to generated page
- employ HTML "encode" command (info added to web page not executable)





Denial-of-Service attacks

- DoS attacks are intended to make system unavailable for normal use
 - To boycott server provider or to demand ransom payment (or “for fun”)
 - Historical DoS attack: TCP 3-way handshake
- Distributed DoS (DDoS) involve distributed computers -that have usually been hijacked- sending hundreds of thousands of requests for service to a web application
 - Defense (system level): specialised software detecting & dropping incoming packets
- DoS attacks can also target app users
 - e.g. lockout user by repeatedly failing authentication with user email address as login name
 - Defenses
 - Temporary user lockouts
 - IP address tracking (to restrict lockout when failed logins come from unusual IP addresses)



il cerca persone e strutture dell'Università di Pisa.

Cerca persone

Persone trovate

| nome | telefono | e-mail | ruolo |
|---------------|--------------|------------------------|----------------------|
| Brogi Antonio | 050 221 2790 | antonio.brogi@unipi.it | Professore Ordinario |



Brute force attacks

Attacker has (only) some information – e.g. valid login name, not password – and repeatedly tries to guess missing information

Attacker can use string generator to generate all possible combinations of letters and numbers

How Secure Is My Password?

✓ The #1 Password Strength Tool. Trusted and used by millions.

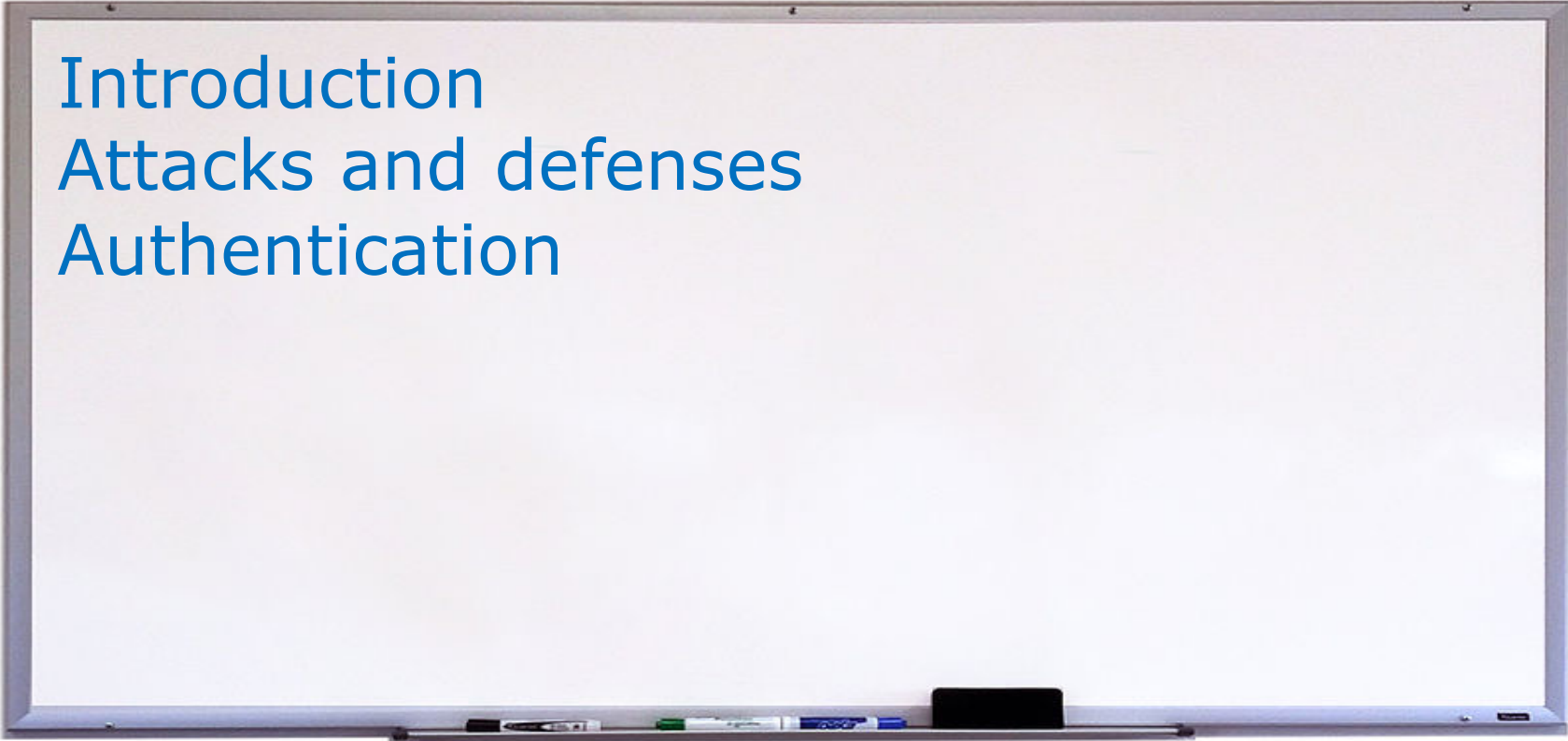
ENTER PASSWORD

<https://www.security.org/how-secure-is-my-password/>

| Password | Time to crack it |
|-------------------|--------------------|
| ASE2223 | 1 sec |
| ASE22-23 | 19 mins |
| ASE22-23# | 16 hours |
| ASE22-23# # | 1 month |
| ASE22-23# # # # # | 12 thousands years |

Defenses

- convince/force users to set long passwords that are not in a dictionary and are not common words
- use two-factor authentication (see later)



Introduction
Attacks and defenses
Authentication

Authentication

Objective: Ensuring that users of your system are who they claim to be

Approaches:

- ***Knowledge-based authentication*** relies on users providing secret, personal information when registering
- ***Possession-based authentication*** relies on users having physical device that can be linked to authenticating system and that generates/displays information known to authenticating system (e.g. system sends code to user's phone number, or special-purpose device that generates one-time codes)
- ***Attribute-based authentication*** relies on a unique biometric attribute of the user (e.g. fingerprint, face)

Many systems now use multi-factor authentication (e.g. password, then code received on mobile phone)

Authentication

Knowledge-based authentication often employed for products delivered as cloud services

Weaknesses of password-based authentication

- Users choose ***insecure passwords*** that are easy to remember
- Users click on email link pointing to fake site that collects login and password → ***phishing attack***
- Users use the ***same password*** for several sites (if there is a security breach at one site ...)
- Users regularly ***forget passwords*** → password recovery mechanism needed → potential vulnerability if credentials have been stolen

Defences

- Force users to set strong passwords
- Add knowledge-based authentication (e.g. user must answer questions)

Authentication

- The level of authentication that you need depends on your product
 - No need to store confidential user information → knowledge-based authentication enough
 - Need to store confidential user information → use two-stage authentication
- Implementing a secure and reliable authentication system is expensive and time-consuming
 - Even if using available toolkits and libraries (e.g. OAuth), there is still a lot of programming effort involved
 - Authentication often outsourced with a federated identity system (see next)

Federated identity

Some websites offer the opportunity to “Login with Google” or “Login with Facebook”

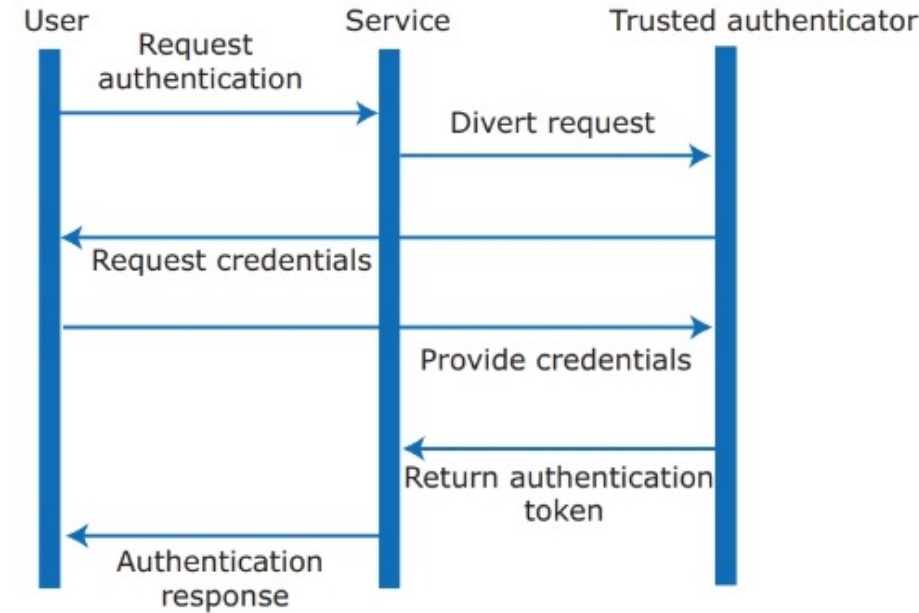
external service is used for authentication

+ user has single set of credentials, stored by a trusted identity service

+ product provider doesn't have to maintain own database of passwords/secrets

+ product provider can get additional user information (if user agrees)

- product provider must share user information with external services



Federated identity verification

ok for products aimed at individual customers

ok for business products, connecting to business's own identity management system

Most federated authentication services use OAuth protocol

Mobile device authentication

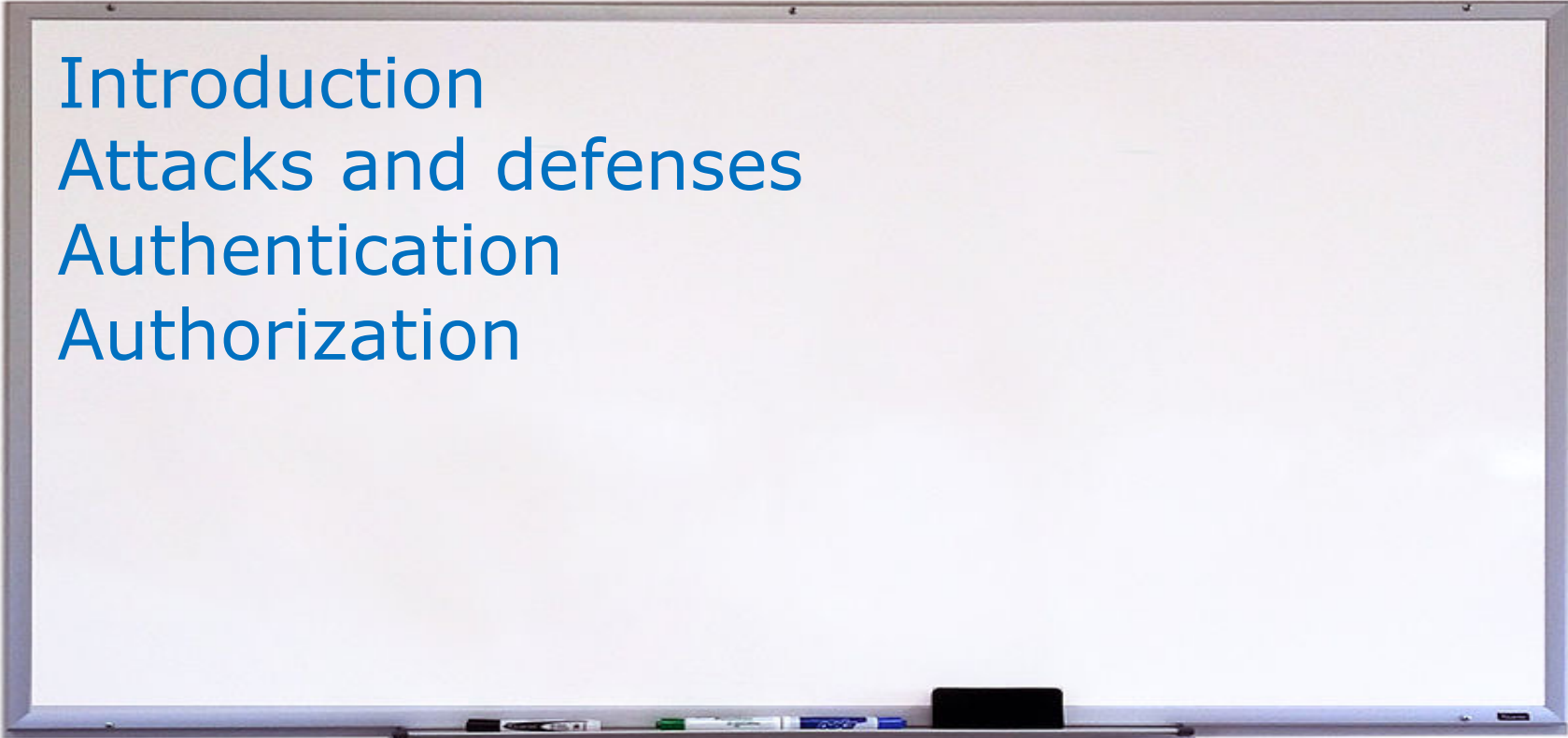


Typing passwords on mobile keyboards is inconvenient

Alternative: install authentication token on mobile device

Potential weakness: if device is stolen/lost, someone else can get access to product

Alternative²: use individual users digital certificates (issued by trusted providers)

A whiteboard with a silver frame and a white surface. On the left side, there is a list of four topics written in blue text. At the bottom of the whiteboard, there is a black eraser and several markers (black, green, blue, and white).

Introduction
Attacks and defenses
Authentication
Authorization

Authorization

Authentication

≠

Authorization

ensure that user is who she claims to be

control that user can access resources



Events
Blog

Events -rwxr-r--
Blog -rwxrwxrwx

Authorization

Access control needed for multiuser products

Access control policy must reflect data protection rules that limit access to personal data

to prevent legal actions in case of data breach

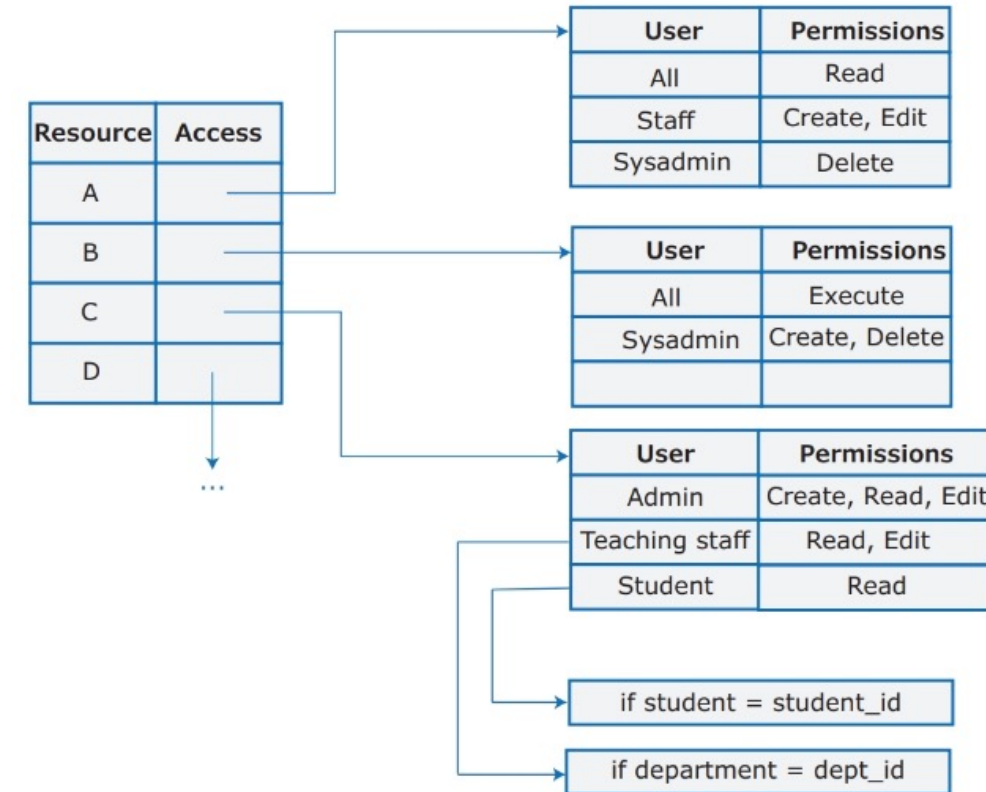
Access Control Lists (ACLs) widely used to implement access control policies

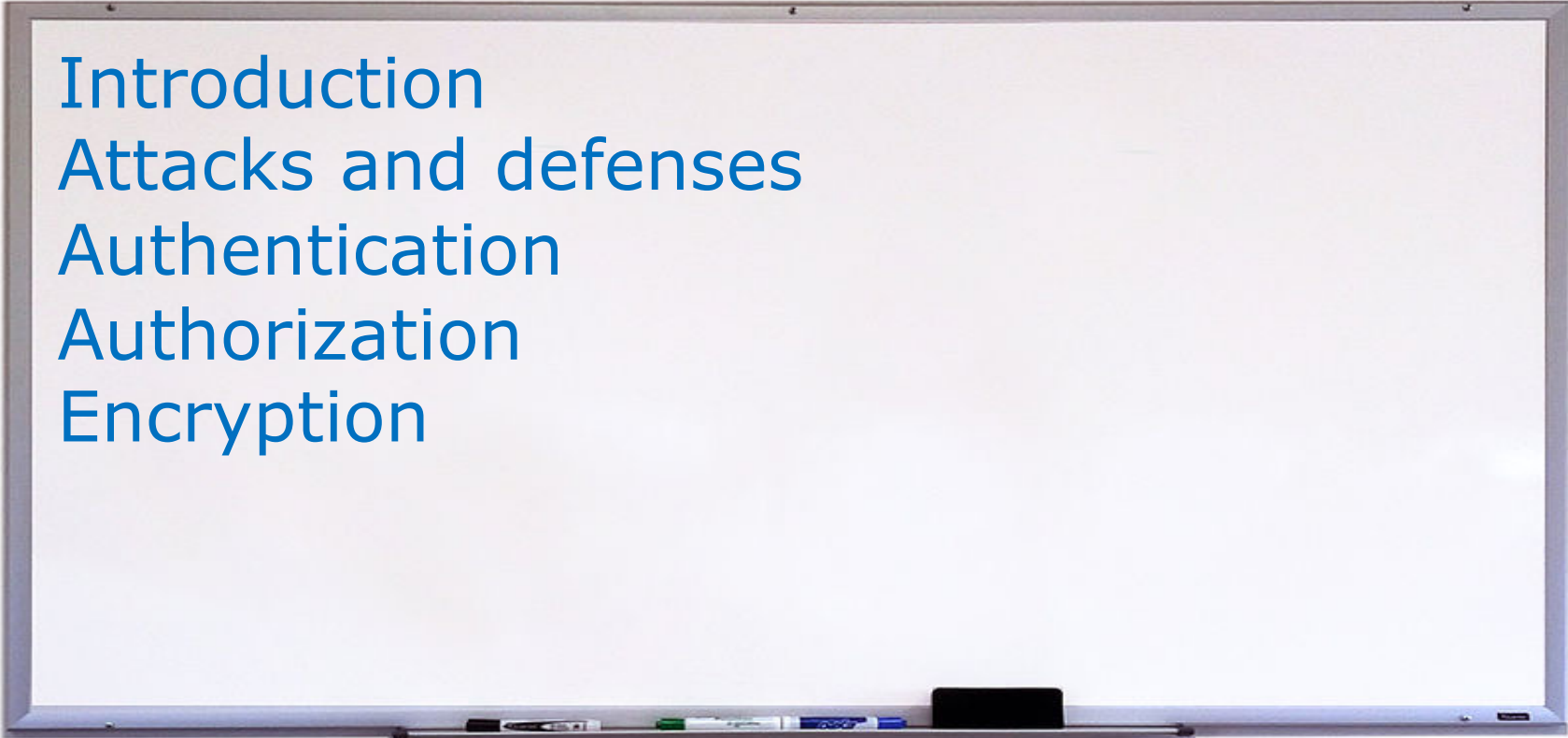
classifying individuals into groups dramatically reduce ACLs size

different groups can have different rights on different resources

hierarchies of groups allow to assign rights to subgroups/individuals

ACLs often realized by relying on ACL of underlying file or db system

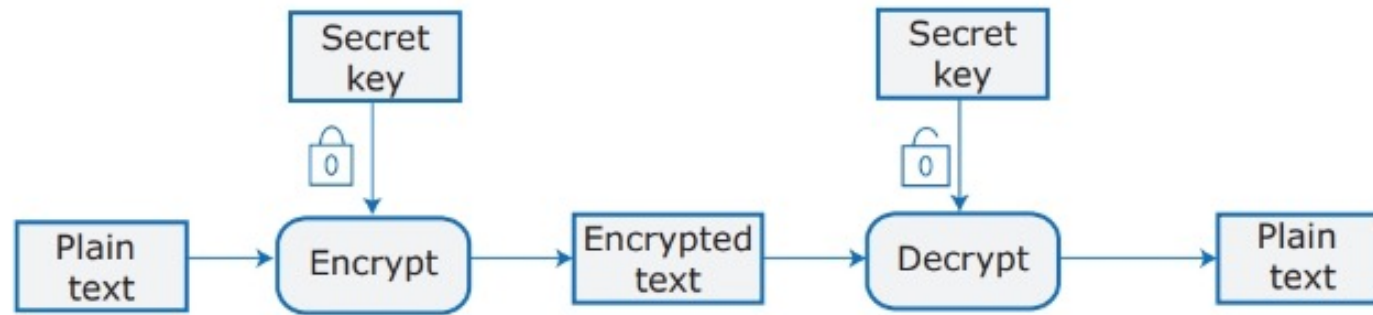


A whiteboard with a silver frame and a white surface. On the left side, a list of five topics is written in blue text. At the bottom of the whiteboard, there is a black eraser and several markers in black, green, and blue.

Introduction
Attacks and defenses
Authentication
Authorization
Encryption

Encryption

Making a document unreadable by applying an algorithmic transformation to it



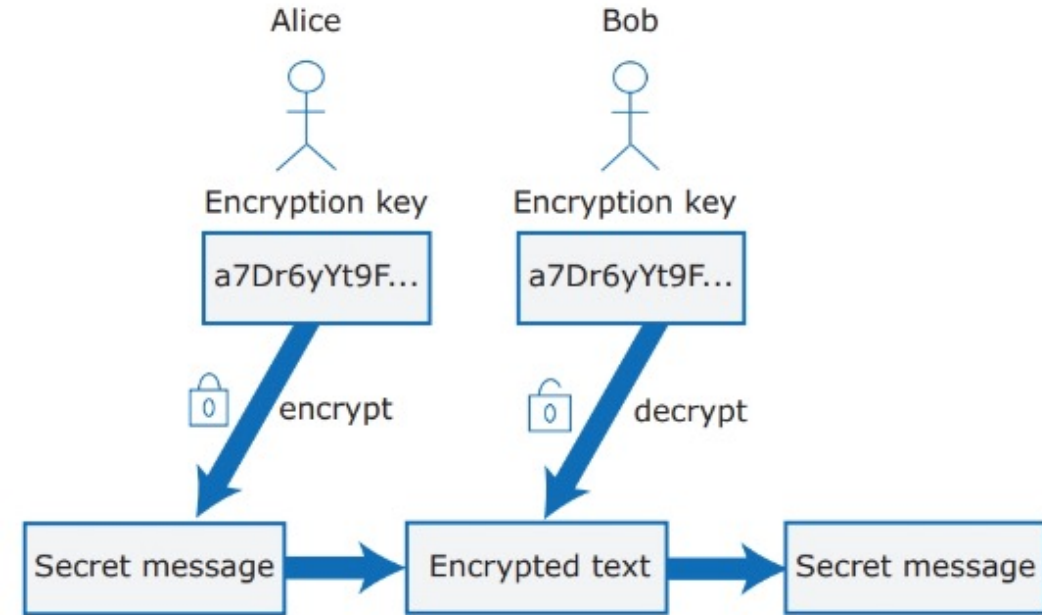
Modern encryption techniques are considered “practically uncrackable” using currently available technology

History tells us that apparently uncrackable encryption may become crackable when new technology becomes available

What if/when quantum computers will become commercially available one day?

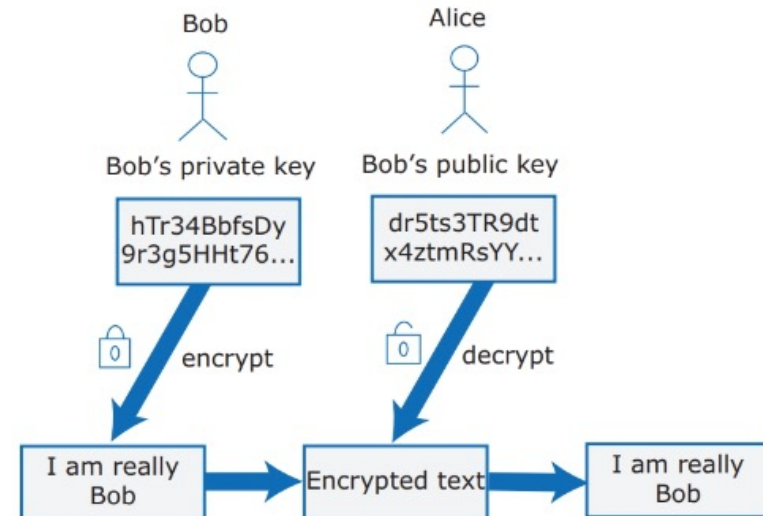
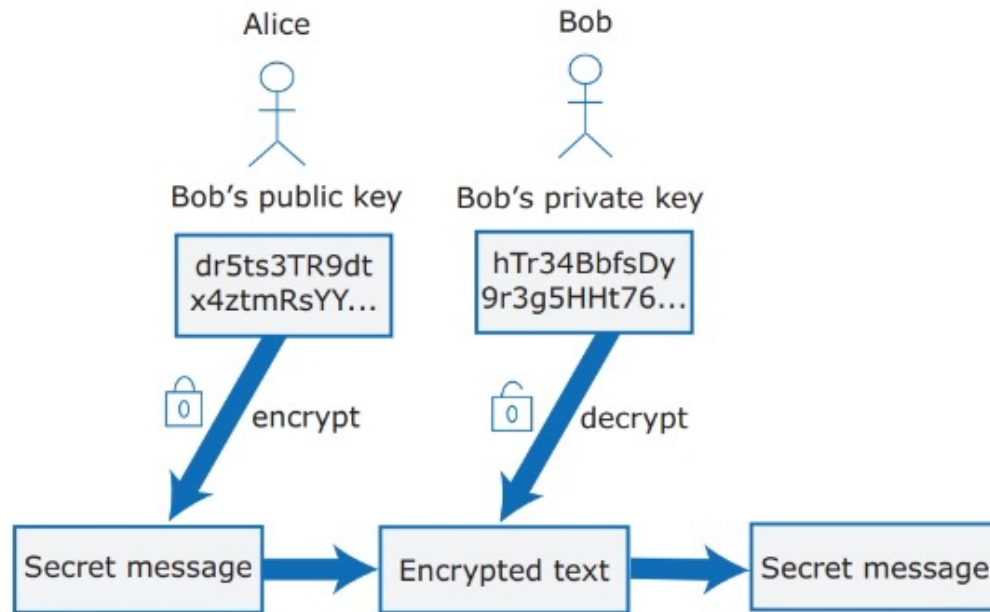
Symmetric encryption

- used for centuries
- same key used for encoding and decoding
- weakness: how to securely share key



Asymmetric encryption

Different (public and private) keys for encrypting and decrypting



TLS and digital certificates

HTTPS = HTTP + ~~SSL~~ TLS (Transport Layer Security)

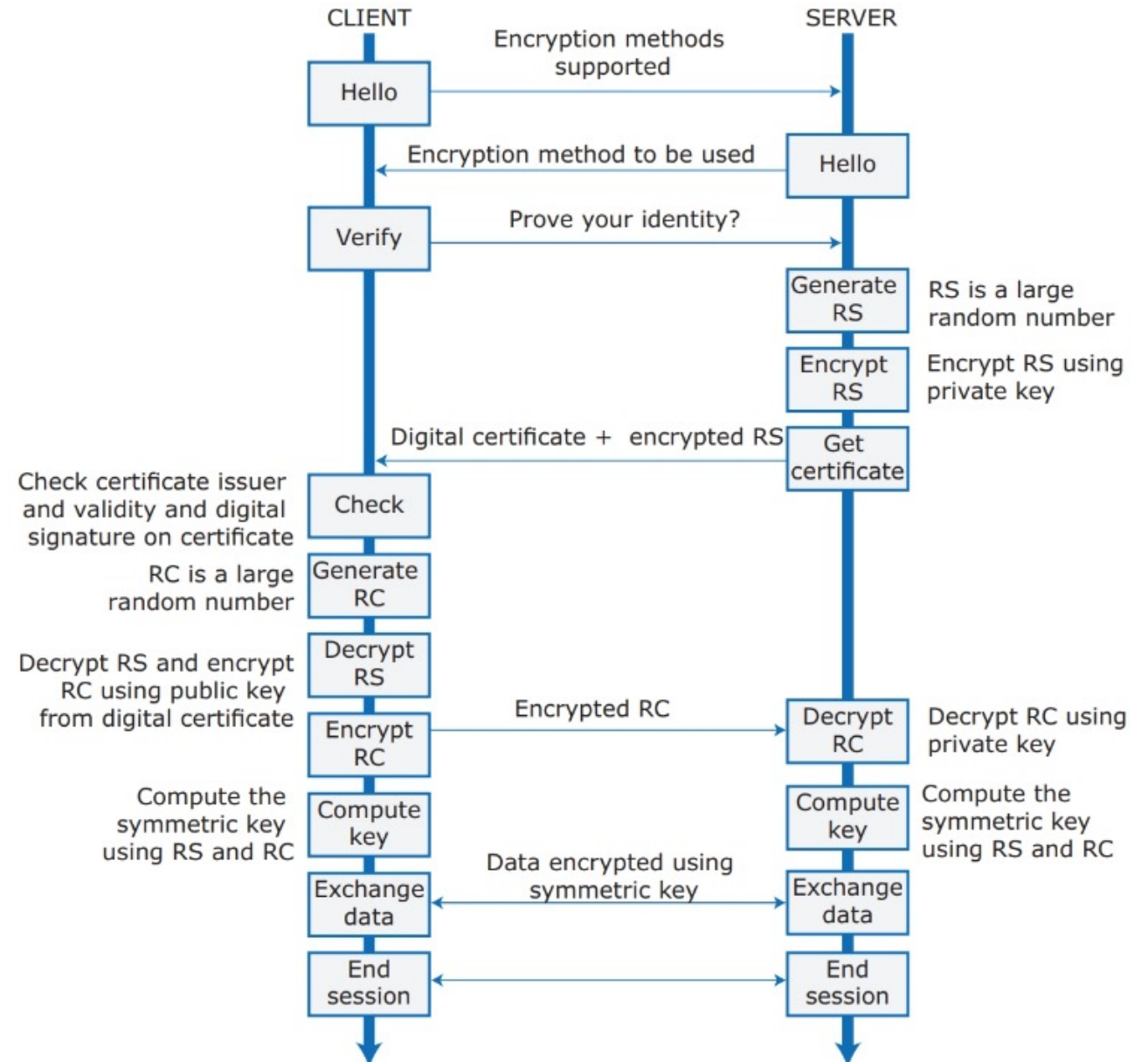
TLS to verify identity of web server and to encrypt communications

TLS encryption uses a digital certificate sent from server to client

Digital certificates issued by trusted identity verification service (CA)

TLS and digital certificates

TLS client-server
interaction to generate
symmetric key to
exchange data



Data encryption

You should encrypt user data whenever it is practicable to do so

- Data ***in transit*** should always be encrypted
- Data ***at rest*** (stored) should always be encrypted
- Encrypting and decrypting ***data in use*** (i.e. actively processed) slow down system response time

Data encryption

Encryption of data is possible at four different levels in the system

| | | |
|-------------|--|--|
| Application | The application decides what data should be encrypted and decrypts that data immediately before they are used. | performance issues need key management (see next) |
| Database | The DBMS may encrypt the entire database when it is closed, with the database decrypted when it is reopened. Alternatively, individual tables or columns may be encrypted/decrypted. | |
| Files | The operating system encrypts individual files when they are closed and decrypts them when they are reopened. | |
| Media | The operating system encrypts disks when they are unmounted and decrypts these disks when they are remounted. | (useful for stolen/lost laptops) |

Key management

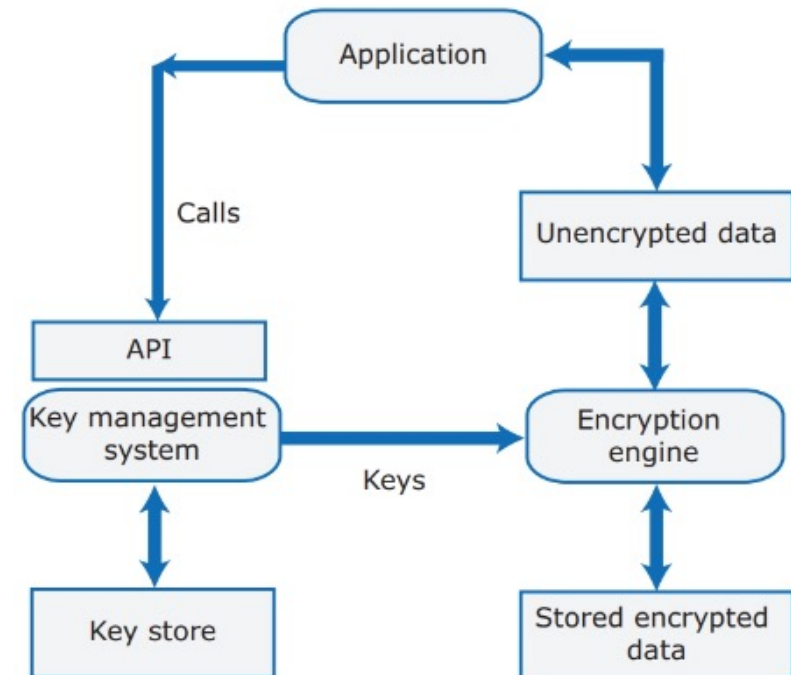


If keys get lost, encrypted data become permanently inaccessible!

Data protection regulations may require that data copies are kept for years and stored securely

Keys should be changed periodically, must maintain multiple timestamped versions of keys

Use a Key Management System (KMS) to make sure that keys are securely generated, stored, and accessed by authorized users



A whiteboard with a silver frame and a white surface. On the left side, a list of topics is written in blue text. At the bottom of the whiteboard, there is a black eraser and several markers (black, green, blue, and white).

Introduction
Attacks and defenses
Authentication
Authorization
Encryption
Privacy

Privacy

Privacy is a social concept that relates to the collection, dissemination, and appropriate use of personal information held by a third party



Privacy

Data protection principles

| Data protection principle | Explanation |
|--------------------------------|---|
| Awareness and control | Users of your product must be made aware of what data are collected when they are using your product, and must have control over the personal information that you collect from them. |
| Purpose | You must tell users why data are being collected and you must not use those data for other purposes. |
| Consent | You must always have the consent of a user before you disclose their data to other people. |
| Data lifetime | You must not keep data for longer than you need to. If a user deletes an account, you must delete the personal data associated with that account. |
| Secure storage | You must maintain data securely so that it cannot be tampered with or disclosed to unauthorized people. |
| Discovery and error correction | You must allow users to find out what personal data you store. You must provide a way for users to correct errors in their personal data. |
| Location | You must not store data in countries where weaker data protection laws apply unless there is an explicit agreement that the stronger data protection rules will be upheld. |

Privacy

Business reasons for paying attention to information privacy

- If your conformance to privacy regulations does not match data protection regulations, you may be subject to legal actions / cannot sell your product
- If you sell a business product, your business customers may require privacy safeguards (not to be at risk with their users)
- Leakage/misuse of client information can damage your reputation

Privacy

The information that your software needs to collect depends on the functionality of your product and on the business model you use

Tips:

- Do not collect personal information that you do not need
- Establish a privacy policy defining how personal/sensitive information about users is collected, stored, and managed
- Make clear if you use users' data to target advertising or to provide services that are paid for by other companies
- If your product includes social network functionalities so that users can share information, you should ensure that users understand how to control the information they share

Reference



Chapter 7 – Security and Privacy