Technologies to enforce security

Managing and Sharing of Construction Data

Computing in Construction

Metropolia

Metropolia

Core cybersecurity properties

Often abbreviated as "CIA" or "CIAAN"

1. Confidentiality
   - protect sensitive information from being disclosed to unauthorized parties
   - includes protecting data at rest, in transit, and in use.
   - common techniques: encryption, access controls, data masking
2. Integrity
   - ensure that information has not been tampered with or modified in an unauthorized way
   - includes protecting data from unauthorized modification, deletion or addition
   - common techniques: digital signatures, message authentication codes, data hashing
3. Availability
   - ensure that information and systems are accessible to authorized users when they need them
   - includes protecting against denial-of-service attacks and ensuring high-availability/fault-tolerance of systems
   - common techniques: load balancing, redundancy, disaster recovery planning
4. Authenticity
   - ensure that information and communication come from a trusted source
   - includes protecting against impersonation, spoofing and other types of identity fraud
   - common techniques: authentication, digital certificates, biometric identification
5. Non-repudiation
   - ensure that a party cannot deny having sent or received a message or transaction
   - includes protecting against message tampering and replay attacks
   - common techniques: digital signatures, message authentication codes, timestamps

Metropolia

Security of data in different states

- **Data in use** – data that is actively being accessed and processed by users
  - most vulnerable as it is immediately available: being read, processed, or updated
  - it is exposed to attack or human mistake
  - security: can be increased through proactive access control measures and data encryption

- **Data in transit** – information is traveling from one point to another
  - for instance, email, collaborative tools, instant messengers, and other communication channels
  - often the focus of attacks
  - security: pass data only in encrypted form

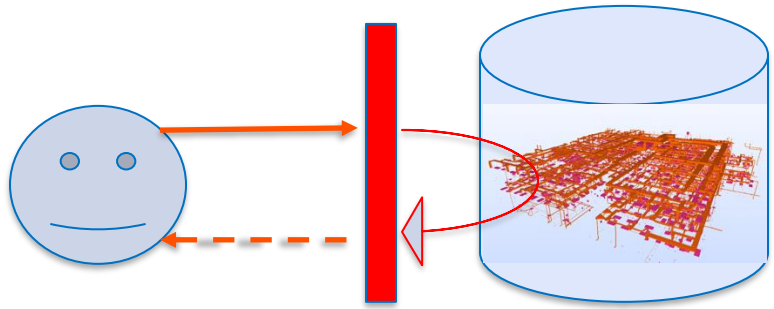
- **Data at rest** – data residing in the storage of servers or user devices
  - for instance, information saved in a database or data kept on a hard drive, computer, or portable device
  - usually least vulnerable
  - security: can be increased by complete disk encryption, data loss prevention solutions, etc.

Metropolia

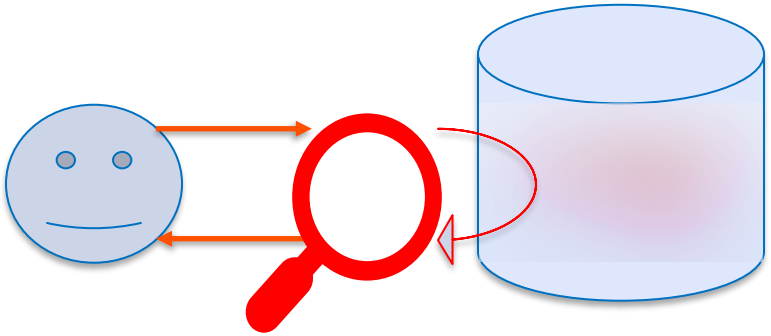
Enforcing information security

Approaches to enforce information security

- Controlling access to the system
  - objective: only authorized person can access the system
    - requires authentication (to know who the person is)
    - requires authorization for access (what use is allowed to do)



- Encrypting the data
  - objective: only authorized persons can read the data
    - those that have a proper decryption key
    - requires the process of key management



Metropolia

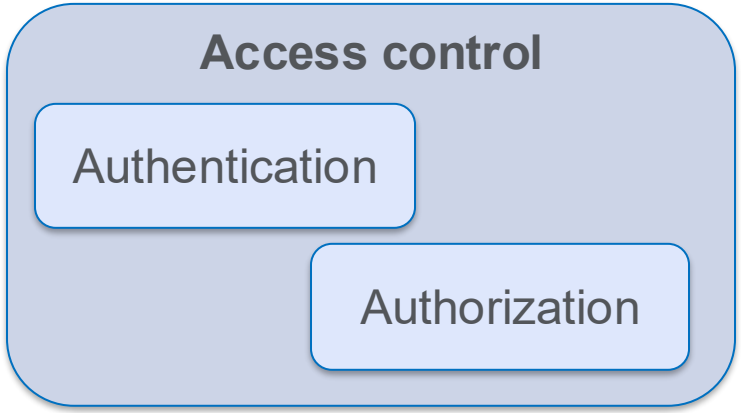
Scope of access control

- How much information will be let out of the system?
  - Consider a cloud-based CDE
  - Even if the user has the right to view an IFC model in CDE, it makes a significant difference from the security perspective whether she has the right
    - to just view an IFC model in browser or
    - to download the IFC model for viewing with a local tool (e.g., with Blender)
  - Even if the user ultimately has the access to all the information in the BIM model through the browser, she can't without significant remodeling effort
    - store the model to act as a basis of some later modeling work (thus infringing IPRs)
    - send the model to others (or even sell it)
    - make complex analyzes of the model using external software
  - If the model can be downloaded, all these are possible
- When download must be allowed, other measures may be used
  - e.g., partitioning the model based on sections/floors/systems and grant the access only to necessary parts
- Security advantage of CDEs
  - CDEs that allow the viewing of the model in place can be safer than traditional file-based project banks that require the user to download models

Metropolia

Concepts related to access control

- Access control
  - selective restriction of access to a resource
  - resource can range from the whole CDE or the project data to a specific BIM model or its version or even parts of the model
  - access control includes authentication and authorization
- Authentication
  - a process of verifying a user's identity
- Authorization
  - to define the access policy to resources
  - often formalized as access control rules in the system
- Single sign-on
  - an authentication scheme that allows a user to log in with a single identity to any of several related but independent software systems



Metropolia

Authentication methods

Knowledge-based authentication (What you know)

- The user provides something she knows
  - Passwords and PINs: User provides a secret string of characters or digits
  - Security questions: Users answer predefined personal questions (e.g., “Mother’s maiden name?”). Often used in account recovery.
  - Passphrases: Like passwords, but typically longer and structured as sentences to increase security

Possession-based authentication (What you have)

- The user possesses a specific object to proves the identity
  - One-time passwords (OTP): A temporary password generated by a device or app (e.g., Google Authenticator, Authy), delivered via text message, email, or dedicated applications
  - Smartcards: Physical cards (with embedded chips) that authenticate a user when inserted into a reader
  - Security tokens: Physical devices that generate a unique code or are plugged into a computer (e.g., RSA SecurID, YubiKey)
  - Mobile authentication apps: Applications like Google Authenticator or Microsoft Authenticator that generate temporary codes for two-factor authentication

Biometric authentication (What you are)

- Unique physical or behavioral characteristics of the user
  - Fingerprint scanning: Sensor that read a fingerprint
  - Facial recognition: Identification using facial features
  - Iris or retina scanning: Scans the patterns in the iris or retina
  - Voice recognition: Voice patterns of the user
  - Behavioral biometrics: Patterns of behavior such as typing rhythm, mouse movements, or interaction with a device

Location-based authentication (Where you are)

- The location of the user is used to validate identity
  - GPS-based authentication: Uses a device’s GPS location to confirm the user's location during login attempts
  - IP-based authentication: Restricts or allows access based on the IP address or geographical location of the user
  - Geofencing: Allows access based on whether the user is within a defined physical location (e.g., within a building or region)

Multi-factor authentication (MFA)

- Combines above methods to enhance security
  - Two-factor authentication (2FA): E.g., a password (something you know) and a code sent to the phone (something you have)
  - Three-factor authentication (3FA): Adds a third factor, like a fingerprint scan (something you are)

Access control systems

- Discretionary access control (DAC) / Individual-based access control
  - access rights are assigned directly to individual users
  - the owner has the discretion to decide who can access the resource and what actions they can perform
  - flexible in a sense that it allows for fine-grained access control (user level)
  - does not scale well and can lead to security risks
- Role-based access control (RBAC)
  - simplifies management by grouping users under roles
  - access rights and permissions are associated with roles
  - most typically used in large-scale deployments
- Attribute-based access control (ABAC)
  - offers the most flexibility
  - supporting complex and dynamic environments
  - requires more upfront policy definition
  - can be complex to implement

Metropolia

Example: RBAC in a construction project

Project manager: Oversees the project, manages schedules, budgets, and has broad access to all project-related information

- **Access rights**: All project-related data: schedules, budgets, designs, reports, ...
- **Privileges**: View, create, edit, delete, approve documents. Assign tasks. Set deadlines

Site engineer: Manages on-site operations and needs access to design documents, schedules, and safety protocols

- **Access rights**: Schedules, site drawings, materials, daily logs, safety protocols
- **Privileges**: View and update daily reports, site data, and schedule progress. Cannot alter high-level designs or budgets

Architect: Responsible for the building's design and should have access to design models, drawings, and specifications

- **Access rights**: All design models, architectural drawings, and specifications. Limited access to scheduling and budgeting data.
- **Privileges**: Can create, view, and edit design documents. Only view high-level project status reports. Collaborate with the BIM manager for model updates

BIM manager: Manages BIM models and needs access to design data, clash detection reports, and model updates

- **Access rights**: All BIM models, clash detection reports, and related design information
- **Privileges**: Edit and manage the model data, perform clash detection, update models, and share reports with other roles

Civil engineer: Works on infrastructure (roads, drainage, or bridges), with access to their design and specifications

- **Access rights**: Access to relevant civil design documents (e.g., roads, drainage systems), as well as construction schedules
- **Privileges**: Can view and update civil design data and collaborate with other engineers, but cannot alter architectural drawings

Procurement officer: Handles purchasing materials and services, with access to vendor information, contracts, and order tracking

- **Access rights**: Vendor information, purchase orders, contracts, and materials data
- **Privileges**: Create, view, and edit procurement documents but cannot access design or on-site construction data

Foreman: Supervises workers on-site and needs access to daily task lists, safety documents, and schedules

- **Access rights**: Task lists, worker assignments, safety protocols, site documentation
- **Privileges**: Update daily progress, mark task completion, and file safety reports. Cannot modify designs or schedules

Client representative: Reviews progress, approves work. Needs limited access to high-level project documents and reports

- **Access rights**: Limited access to high-level reports, project status, and milestone approvals
- **Privileges**: View reports. Approve/reject project milestones. Cannot modify any project data

Health and safety officer: Ensures compliance with safety regulations. Needs access to safety protocols, incident reports, and worker certifications

- **Access rights**: Safety protocols, worker certifications, incident reports, and safety audits
- **Privileges**: View and edit safety documents and incident reports. Cannot alter design or procurement documents.

Example: ABAC in a construction project

Attributes in ABAC

- In ABAC, **access decisions** are made based on a combination of attributes about
  - **User attributes:** job title, certifications, clearance level
  - **Resource attributes**: document type, project phase, confidentiality level
  - **Environmental attributes**: time of day, location, network conditions
  - **Action attributes:** view, edit, approve

Example: ABAC in a project

- **User attributes**
  - **Job title**: Architect, BIM manager, Site engineer, Civil engineer, Procurement officer
  - **Clearance level**: Confidential, Restricted, Public
  - **Certifications**: Safety training, Project management certification
  - **Experience level**: Junior, Senior
- **Resource attributes**
  - **Document type**: BIM model, schedule, budget, contract, safety protocol
  - **Project phase**: Design, Construction, Handover
  - **Confidentiality level**: Public, Internal, Restricted, Confidential.
- **Environmental attributes**
  - **Time of day**: Access might be restricted to work hours for certain actions
  - **Location**: On-site or remote access could affect permissions
  - **Network security**: Whether the user is on a secure network or accessing data over a public connection
- **Action attributes**
  - **View**: Read-only access to documents or models
  - **Edit**: Ability to make changes to data.
  - **Approve**: Permission to authorize changes or project milestones

ABAC Access decision example

- Let's say **Sophia** is an **Architect** on a construction project. In an ABAC system, her access to a specific resource (e.g., a BIM model) might depend on:

- **User attributes**
  - Sophia's **job title**: Architect
  - Sophia's **clearance level**: Restricted
  - Sophia's **experience level**: Senior (affects editing rights)
- **Resource attributes**
  - The **document type**: BIM Model
  - The **confidentiality level**: Internal
  - The **project phase**: Design
- **Environmental attributes**
  - **Time of day**: Sophia can only access the model during working hours.
  - **Location**: Sophia must be on a secure company network to access the design model for editing.
- **Action attributes**
  - **Edit policy**: User has editing rights if she
    1. has the job title an Architect,
    2. has Senior level experience,
    3. has the clearance level the resource requires,
    4. the access is during working hours, and
    5. the access is from a secure network.

Access decision

– Sophia has the right to edit the BIM model

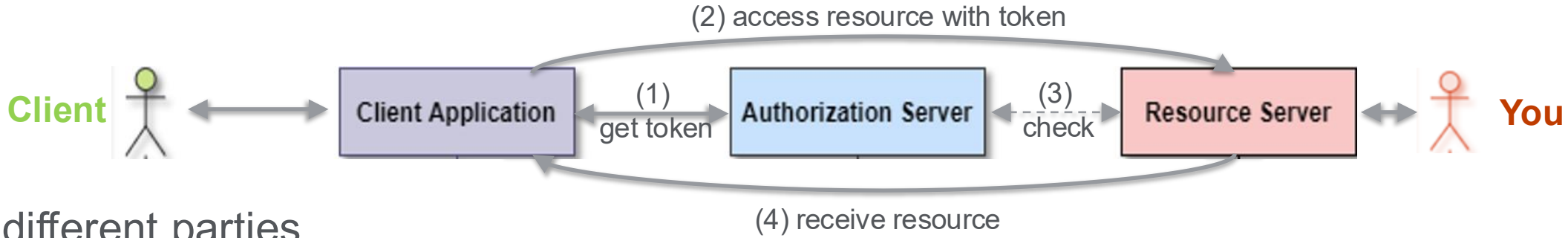
	Individual-Based Access Control (DAC)	Role-Based Access Control (RBAC)	Attribute-Based Access Control (ABAC)
Control focus	Individual users	Predefined roles assigned to users	Based on attributes of users, resources, etc.
Decision factor	Specific individual user identity	User's role or job function	Attributes of user, role, resource, environment
Flexibility	Low: Changes require manual permission updates for each user	Moderate: More flexible than DAC, but limited to predefined roles	High: Extremely flexible, based on dynamic attributes and contextual information
Granularity	Fine-grained: Access is granted per user.	Coarse-grained: Access is granted per role	Very fine-grained: Access can be customized
Scalability	Poor: Difficult to manage with many users	Good: Scales well in organizations with clearly defined roles	Excellent: Can handle complex, dynamic environments with many factors
Context awareness	None: Access decisions are not based on context like time or location	Minimal: Access is based on role, not context (though context-dependent roles can be used)	High: Can include dynamic factors such as time, location, device, network, etc.
Ease of management	Difficult: Permissions must be set individually for each user and resource.	Moderate: Roles must be defined and managed, but changes apply to all in the role	Complex: Requires defining and managing multiple attributes and policies
Common use cases	Small systems or individual user-level controls, file permissions	Organizations with well-defined roles (e.g., corporate environments, structured projects)	Dynamic environments requiring fine-grained, contextual control (highly regulated industries)
Access rights assignment	Resource owners manually assign permissions to individual users	Administrators assign roles to users; each role has predefined access rights	Admins define policies based on attributes; access is dynamically assigned based
Reusability of permissions	Low: Permissions are specific to each user	High: Roles can be reused for multiple users.	High: Policies can be reused across a wide range of users and resources
Use scenario in construction	File sharing in a small project, where the Project Manager directly grants access to individual team members.	Large construction projects with distinct roles such as Architect, BIM Manager, and Site Engineer, each with predefined rights	Complex projects with changing phases and conditions, where access needs to depend on project phase, location, user credentials, etc.
Examples of access criteria	User: "John has read access to this file."	Role: "All Engineers can view the project schedule."	Attributes: "Senior Architects working on-site during work hours can edit BIM models"
Security risk	Higher: Individual assignments can lead to excessive access	Lower: Roles limit access, but over-assigned roles can pose risks	Lower: Detailed attribute policies minimize risk but can be complex to manage properly
Auditability	Limited: It can be difficult to track access across individual users	Good: Easier to track because users inherit access rights from roles	Excellent: Provides detailed logs based on user, resource, and environmental attributes

OAuth – Delegation of authorization

OAuth (Open Authorization)

Open standard for access delegation (currently OAuth 2.0)

- resource owner (resource server) delegates the authorization responsibility to authorization server
- users do not need to give their passwords to different websites (resource servers)



Roles of different parties

1. **Resource owner** (You) have control over certain resources or data
   - For instance, BIM models, project plans, documents, or databases
2. **Client** (External application) needs access to your project data
   - For instance, a project management tool, a mobile app, or calendar system
3. **Authorization server** (Gatekeeper) acts as a gatekeeper that manages access to your resources
   - It checks whether the client is allowed to access your data.
4. **Access token** (Key): Instead of sharing your username and password with the client, OAuth uses an access token
   - Token is a digital key that the client presents to request access to your resources
5. **User Consent**: Before granting access, you have the final say whether to allow the client's request
   - OAuth ensures that you give explicit consent

Advantages of OAuth

For Users

- **Enhanced security:** By not sharing passwords directly with third-party apps, users reduce the risk of their credentials being compromised. OAuth utilizes tokens that expire, further minimizing the potential damage from a security breach.
- **Simplified login process:** OAuth allows users to log in to multiple services using a single identity provider, eliminating the need to remember numerous passwords.
- **Greater control:** Users have granular control over which apps can access their data and what level of access is granted. This empowers users to protect their privacy and security.

For Developers

- **Streamlined Integration:** OAuth simplifies the process of integrating with third-party services by providing a standardized way to request and obtain access tokens.
- **Improved User Experience:** By offering seamless and secure authentication, OAuth enhances the overall user experience.
- **Scalability:** OAuth is designed to handle a large number of users and applications, making it suitable for high-traffic websites and services.
- **Flexibility:** OAuth supports various authorization grant types, allowing developers to choose the most appropriate method for their specific use case.
- **Security:** OAuth prioritizes security by using tokens and limiting the scope of access granted to third-party apps.

Combining OAuth with RBAC or ABAC

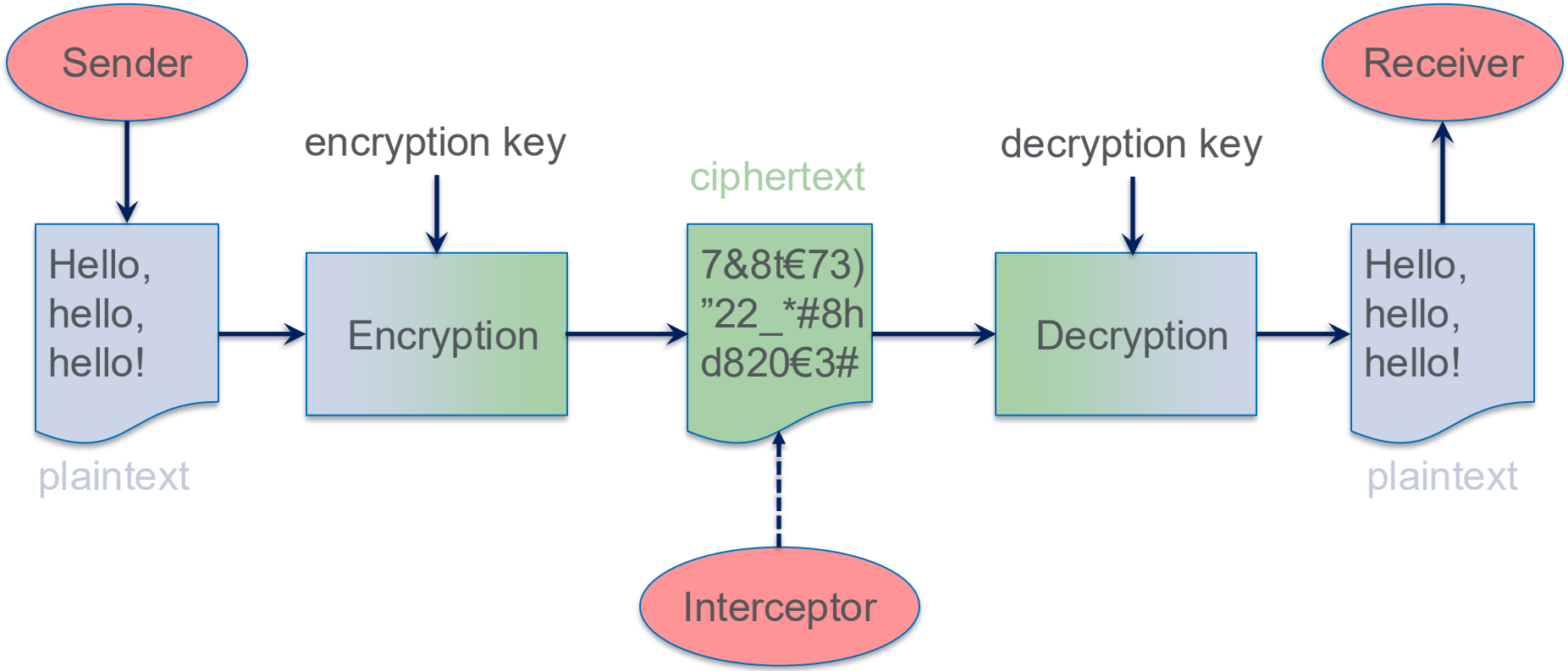
- Consider a CDE where design company A.com has shared a BIM model
- A.com can grant an LCA analysis application access to the model
  - OAuth
    - A.com
      - authenticates with the CDE using OAuth
      - grants the LCA application access to the BIM model
  - RBAC or ABAC
    - Determines what specific actions the LCA application can perform on the BIM model
    - For example, the LCA application might be allowed to
      - read BIM model itself but not access documents linked with it
      - read material and product types and quantities but not geometry
- In this scenario,
  - OAuth provides the initial authentication and authorization
  - RBAC or ABAC defines the specific permissions granted to the third-party app

Metropolia

Cryptography

Cryptography

- Idea: Using strong-enough cryptography, data can safely exist anywhere after it has been encrypted
  - it is only readable for those who have the key to decrypt it
  - cryptography is a method to achieve confidentiality but can also be used to achieve authenticity and non-repudiation



Metropolia

Cryptosystems

1. Symmetric cryptosystem
   - the same key is used to encrypt and decrypt the data
   - the key is the shared secret of the sender and the receiver
     - problem: how to exchange the key in a secure manner
   - works well with multiple receivers
   - generally, less computation intensive and practical with long messages
2. Public-key cryptosystem
   - the receiver has two keys, a public one and a private one
     - the private key must not be shared with anyone, while the public key can be shared with everyone
   - the sender encrypts the data with the public key of the receiver
   - the receiver decrypts the encrypted data with her private key
   - works only with one receiver (data needs to be encrypted separately for each receiver)
   - generally, more computation intensive and clumsy with long messages
3. Hybrid cryptosystem
   - first, exchange a symmetric key using public-key cryptography
   - then, exchange the data using symmetric-key cryptography

Metropolia

Symmetric cryptosystems

- Same key for encryption and decryption
  - shared secret of the sender and the receiver
- Advantage
  - good for bulk encryption
  - widely used
- Drawbacks
  - key exchange needs another system to implement
  - do not provide non-repudiation or authenticity
  - historical vulnerability to various attacks
    - for instance, known-plaintext attack

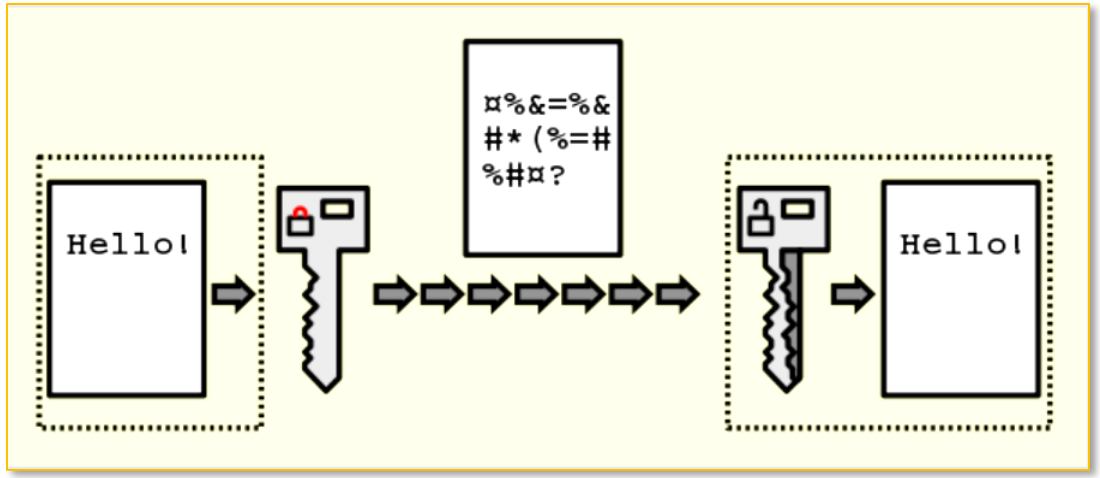
Examples of symmetric-key algorithms:

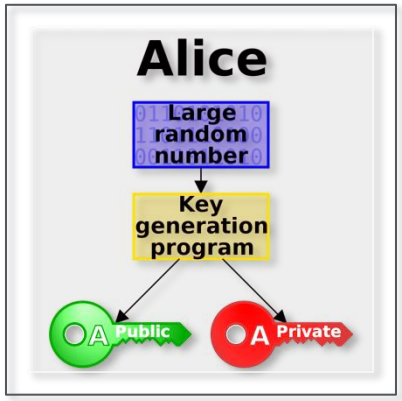
- RC4
- DES
- 3DES
- Twofish
- Serpent
- AES (Rijndael)
- Camellia
- Salsa20
- ChaCha20
- Blowfish
- CAST5
- Kuznyechik
- Skipjack
- Safer
- IDEA

Metropolia

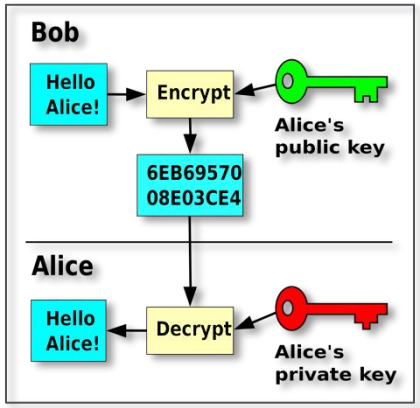
Public-key cryptosystem

- One-to-one
- No shared secret
- Can be used to provide for
  - non-repudiation
  - authenticity
- Open implementations such as PGP

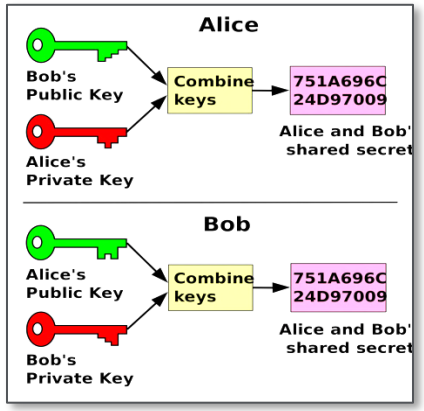




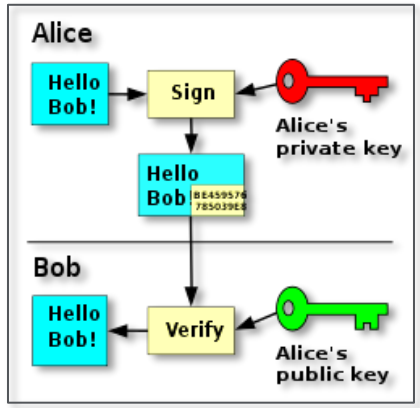
Key generation



Encryption and decryption of data



Creation of a shared secret (e.g., a symmetric key)



Digital signature

Metropolia

Examples of hybrid cryptosystems

- TSL – Transport Layer Security (also used by HTTPS)
  - Used every time you access a web pages using https-protocol
  - Asymmetric encryption is used to establish a secure connection between two parties
    - for securely exchanging a symmetric session key between the client and the server
  - After the symmetric key is exchanged, symmetric encryption is used to encrypt the actual data exchanged between the client and the server, ensuring both confidentiality and efficiency
- SSH – Secure Shell
  - Used in remote accesses to login to another computer
  - During the initial handshake, asymmetric encryption is used to securely negotiate and exchange a session key
  - Once the session key is established, SSH switches to symmetric encryption for encrypting the actual session data, ensuring faster encryption and decryption
- PGP – Pretty Good Privacy / GPG – GNU Privacy Guard
  - PGP/GPG is widely used for secure email communication and file encryption
  - Asymmetric encryption (e.g., RSA or ECC) is used to encrypt a symmetric session key
  - Symmetric encryption (e.g., AES) is used to encrypt the actual message

Other technologies

Metropolia

Firewalls and IDPSs

Firewalls and Network Security Technologies

- Firewalls
  - Act as barriers between trusted and untrusted networks, filtering traffic based on predefined rules
  - Technologies: Stateful firewalls, Next-Generation Firewalls (NGFW), Web Application Firewalls (WAFs)
- Network segmentation
  - Divides a network into isolated segments to contain and minimize the spread of attacks
  - Technologies: VLANs (Virtual LANs), Software-Defined Networking (SDN), Zero Trust Architecture.

Intrusion Detection and Prevention Systems (IDPS)

- Intrusion Detection Systems (IDS)
  - monitor networks or systems for malicious activity or policy violations.
- Intrusion Prevention Systems (IPS)
  - Not only detect but also take action to block or mitigate detected threats.
  - Technologies: Snort (IDS/IPS), Suricata, OSSEC

Metropolia

Data safety

Data loss prevention (DLP)

- DLP systems detect and prevent unauthorized access or transmission of sensitive data outside the network or organization
- Technologies
  - DLP software for email, cloud, and endpoint security
  - Content Filtering
  - Cloud Access Security Brokers (CASBs)

Secure backup and disaster recovery solutions

- Regular, secure backups are essential for ensuring data **availability** in case of an attack or disaster
- Technologies
  - Snapshot backups
  - Incremental backups
  - Encrypted cloud storage
  - Backup management systems

Metropolia

Data Masking and Obfuscation

- Hide sensitive data
  - ensuring that unauthorized users cannot interpret or access the information
- Techniques
  - **Tokenization**: Replaces sensitive data with non-sensitive equivalents (tokens), commonly used in payment processing
  - **Data masking**: Redacts or alters data so that only authorized users can view the original information. Often used in test environments
  - **Homomorphic encryption**: Allows computations to be performed on encrypted data without decrypting it, preserving privacy
    - Keeps data encrypted also in the “data in use” phase
    - Fully homomorphic encryption would open tremendous new business possibilities, for instance, in the context of cloud computing
      - However, performance remains an issue

Metropolia

Integrity mechanisms

- Integrity
  - ensure that data remains unchanged during transmission or storage (unless authorized changes are made)
- Hash functions to compute a checksum of the content
  - verify that the content has not been tampered with
- Hashing algorithms
  - Secure Hash Algorithm (SHA-256, SHA-3)
  - Message Digest (previously MD5 was widely used, but is now deprecated)
- Digital signatures
  - Used for verifying the integrity and authenticity of messages and documents
  - Combine hashing and asymmetric encryption to ensure that data hasn't been tampered with
  - Technologies
    - RSA Digital Signatures
    - ECDSA (Elliptic Curve Digital Signature Algorithm)

User safety

Virtual Private Networks (VPNs)

- Create a secure, encrypted tunnel for data to transit between a user and a remote server
- Protects the confidentiality of data in transit
- Technologies
  - OpenVPN
  - IPsec VPN
  - SSL VPN

Endpoint Security Solutions

- Securing individual devices (laptops, phones, servers) that connect to the network
- Protection from malware and other attacks
- Technologies
  - Antivirus software
  - Endpoint Detection and Response (EDR)
  - Host Intrusion Detection Systems (HIDS).

Metropolia

Thank you!

Metropolia