* 1. **Security controls notes**

**Categories**

Technical – The control is implemented as a **system** (hardware, software, or firmware).

EX: Firewalls, antivirus software, OS access control models.

Managerial – This control gives **oversight** of the information system.

EX: Risk identification, or a tool allowing the evaluation and selection of other sec controls.

Operational – This control is implemented primarily by **people**.

EX: **Security guard**s and training programs.

Physical – Controls such as alarms, gateways, locks, lighting, and security cameras that **deter** and **detect** access to a premises and hardware.

**Control types**

Preventive – Acts to eliminate or reduce the **likelihood** that an attack can succeed. Operates **BEFORE** an attack can take place.

 EX: ACLs on a firewall, Antimalware software.

Detective – Does not prevent or deter access but will **identify** and **record** an attempted or successful intrusion.

 EX: Logs

Corrective – Eliminates or reduces the **impact** of a security policy violation. This is used **AFTER** an attack.

 EX: A backup system to restore data AFTER an attack. A patch management system that eliminates the vulnerability exploited during the attack.

Directive – Enforces a **rule of behavior**, such as a policy, best practice standard or SOP.

 EX: an employee contract that outlines what will happen if they do not comply to security policies. Training and awareness programs

Deterrent – May not physically or logically prevent access but it **PSYCHOLOGICALLY** discourages an attacker from attempting an intrusion.

 EX: Signs and warnings of legal penalties against trespass or intrusion.

Compensating – A **substitute** for a principal control that affords the **same** **or** **better** level of protection but uses different methodology or technology.

EX:If something is broken, you can substitute it with something that will afford the same or better even if it uses different technologies.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.2 Security concepts notes**

**CIA** – Also referred to as AIC. **C**onfidentiality, **I**ntegrity, **A**vailability. The systems used to store, transmit and process data must demonstrate the properties of security. Secure information has three properties:

Confidentiality – Information can only be read by people who have been **explicitly authorized** to access it.

 Integrity – Data is **stored** and **transferred** **as intended** and that any modification is authorized.

 Availability – Information is **readily** **accessible** to those **authorized** to view or modify it.

**Non-repudiation** – A person cannot deny doing something, such as: **Creating, Modifying** or **Sending** a resource.

(EX: A legal document, such as a will, must usually be witnessed when it is signed.)

**Cybersecurity Framework** – Within the goal of ensuring Info security, Cybersecurity refers specifically to provisioning secure processing hardware and software. Info security and Cybersecurity tasks can be classified as 5 functions (Framework developed by National Institute of Standards and Technology – NIST).

Identify – **Develop** **security** **policies** and **capabilities**. Evaluate risk, threats and vulnerabilities and recommend security controls to **mitigate** them.

Protect – Procure/develop, install, operate, and decommission IT hardware and software assets with **security** as an **embedded** **requirement** of every stage of this operation’s lifecycle.

Detect – Perform ongoing, proactive **monitoring** to ensure that controls are effective and capable of protecting against new types of threats.

 Respond – **Identify**, **analyze**, **contain**, and **eradicate** **threats** to systems and data security.

 Recover – Implement cybersecurity resilience to **restore** **systems** and **data** if other controls are unable to prevent attacks.

**Why use a framework? –** The use of a framework allows and organization to make an objective statement of its **current** **cybersecurity** **capabilities**, identify a target level of capability, and prioritize investments to achieve that target. This gives a **structure** to internal risk management procedures and provides an externally verifiable statement of **regulatory** **compliance**.

**Gap analysis** – A process that identifies how an organization’s security systems **deviate** from those **required** or **recommended** by a **framework**. A Gap analysis is performed when **first** **adopting** a framework or when **meeting** a **new** **industry** or **legal** **compliance** **requirement**. The analysis might be repeated every few years to meet compliance requirements or to validate any changes that have been made to the framework.

For each section of the framework, a Gap analysis report will provide:

-An **overall score**

-A detailed **list** of **missing** or **poorly** **configured** controls associated with that section

-**Recommendations** for **remediation**.

While a Gap analysis COULD be performed by the internal security team, it is more likely to involve a **third-party consultant**. Frameworks and compliance requirements from regulations and legislation can be **complex** enough to require a **specialist**. Advice and feedback from an external party can alert the internal security team to oversights and to **new trends** and **changes** **in best practice**.

**Access control** – An access control system ensures that an information system meets the goals of the CIA triad. Access control governs how **subjects**/principals may **interact** with **objects**.

Subjects – People, devices, software processes, or any other system that can request and be **granted** **access** **to** **resources**.

Objects – Objects are resources. Network, server, database, app or file.

(Subjects are **assigned** **rights** or **permissions** on resources.)

**IAM** – Modern **access** **control** is typically implemented as an **I**dentity and **A**ccess **M**anagement (IAM) system. Also referred to as **AAA**. The use of IAM to describe enterprise security workflows is becoming more prevalent as the importance of the **identification** process is better acknowledged. IAM comprises four main processes:

Identification – Creating an **account** or **ID** that **uniquely** represents the user, device, or process on the network.

Authentication – Proving that a subject is **who or what it claims to be when it attempts to access the resource**. An authentication factor determines what sort of credential the subject can use.

(EX: People can be authenticated by providing a password; a computer system could be authenticated by using a token such as a digital certificate.)

Authorization – **Determines** what **rights** subjects should have on each resource and **enforces** those rights.

(EX: You may have authorization to READ a file, but the owner of the file has decided you are not authorized the WRITE/EDIT the file.)

Accounting – **Tracking** authorized usage of a resource or use of rights by a subject and **alerting** when **unauthorized** **use** is **detected** or **attempted**.

 (EX: A server log that makes note of every user that logs in successfully / unsuccessfully.)

**Zero Trust** – A security model that assumes that all devices, users, and services **are not inherently trusted**, regardless of whether inside or outside a network’s perimeter. Instead, the Zero trust model requires ALL users and devices to be authenticated and authorized before accessing network resources. The goal of zero trust design is to make the **implicit trust zone** as **small** / **transient** as possible.

**Control plane** – Manges policies that **dictate** **how** **users** and **devices** are authorized to access network resources. It is implemented through a centralized **policy** **decision** **point**. The policy decision point is responsible for defining policies that limit access to resources on a **need-to-know** basis, **monitoring** **network** **activity** for suspicious behavior and **updating** **policies** to reflect changing network conditions and security threats.

The policy decision point is comprised of two subsystems:

Policy Engine – **Configured** with subject and host identities and credentials, access control policies, up-to-date threat intelligence, behavioral analytics, and other results of host and network security scanning and monitoring. This comprehensive state data allows it to **define an algorithm** and **metrics** for making **dynamic** **authentication** and **authorization** **decisions** on a per-request basis.

Policy Administrator – Responsible for **managing** the process of issuing access tokens and establishing or tearing down sessions, based on the decisions made by the policy engine. The policy admin **implements** **an** **interface** between the control plane and data plane.

--------------------------------------------------------------------

Adaptive identity – Recognizes that user identities are **not** **static,** and that identity verification must be **continuous** and based on a **user’s current context** and the resources they are attempting to access.

Threat scope reduction – Access to network resources is granted on a **need-to-know** basis, and access is **limited** to only those **resources required to complete a specific task**. This concept reduces the network’s attack surface and limits the damage that a successful attack can cause.

Policy-driven access control – Describes how access control policies are used to **enforce** **restrictions** based on user **identity**, **device** **posture**, and **network** **context**.

(Device posture refers to the security status of a device: security configs, software versions, patch levels.)

**Data plane** – While systems in the Control plane define policies and make decisions, systems in the Data plane **establish** **sessions** for **secure** **information** **transfers**.

Subject / System – A **subject** (user or service) uses a **system** (Client host PC, laptop, or smartphone) to make **requests** for a given **resource**. Eachrequestis **mediated** bya **Policy Enforcement Point.**

Policy Enforcement Point – The Policy Enforcement Point **interfaces** with the P**olicy** A**dmin** to set up a secure data pathway if access is approved, or tear down a session if access is denied or revoked. The Policy Enforcement Point system is the **only one trusted** to **communicate** **requests** and **receive** **decisions** from the Policy Admin.

Implicit trust zones – The **data** **pathway** established **between** the **Policy** E**nforcement** P**oint** and the **resource** is referred to as an implicit trust zone.

**Extra Info** - Separating the **Control plane** and **Data** **plane** is significant because it allows for a more **flexible** and **scalable** network architecture.

**Zero** **Trust** **Architecture** examples to google if you’re curious:

-Google BeyondCorp

-Joint Enterprise Defense Infrastructure

-Cisco Zero Trust Architecture

-Palo Alto Networks Prisma Access

**Physical security** – Helps protect the organizations physical assets, including servers, datacenters, and other critical infrastructure from unauthorized access, theft, or damage. Reduces the risk of insider threats.

Bollards – Short vertical **posts** made of durable materials. Installed at intervals around a perimeter or entrance. Protects pedestrians from vehicular traffic and **prevents** **unauthorized** **vehicle** **access**.

Access control vestibule – Also known as a **mantrap**. Regulates **entry** to a **secure** **area**. Involves two doors or gates that interlock and permit only **one individual** to pass through at a time.

Fencing – Protects the **exterior** of a building. Usually transparent, robust, and secure against climbing. A drawback is that it gives the building an intimidating appearance that some companies might not want.

Video surveillance – Cheaper means of providing surveillance than maintaining separate guards at each gateway or zone. Can also serve as a deterrent. A drawback is that the response time may be longer than that of a security guard if no one is monitoring the surveillance system.

Security guard – The visible presence of guards is a very effective intrusion detection and deterrence mechanism.

Access Badge – Plastic cards embedded with magnetic strips, **radio** **frequency** **identification** **chips**, or **near**-**field** **communication** technology. Issued to authorized employees. Also serve as a form of identification.

Lighting – Helps secure the building at **night**. Works as a deterrent by making intrusion more difficult and surveillance easier.

Sensors – Provide proactive detection and alerting capabilities against potential security breaches.

Infrared Sensors – Detect changes in **heat** **patterns** caused by moving objects such as human intruders.

 Pressure Sensors – Typically installed inside **floors** or **mats** and are activated by weight.

Microwave Sensors – Emits microwave pulses and measures the **reflection** **off** a **moving** **object**. Often combined with infrared detectors in dual-technology motion sensors. **Less** **likely** to **trigger** **false** **alarms** as the infrared and microwave sensors must be tripped simultaneously to trigger an alarm.

Ultrasonic Sensors – Emits sound waves at frequencies above the **range** of **human** **hearing** and measure the **time** it takes for the waves to return after hitting an object. Often used in automated lighting systems to switch lights on when someone enters a room.

**Deception and Disruption technology** – Cybersecurity resilience tools and techniques to increase the cost of attack planning for the threat actor. The following are all cybersecurity tools used to detect and defend against attacks.

Honeypot – **Decoy** **systems** that **mimic** real systems and applications. Designed to allow security teams to **monitor** attackers’ activity and **gather** information about their tactics and tools.

Honeynet – A **network** of **interconnected** **honeypots** that simulate an entire network, providing a more extensive and **realistic** **environment** for attackers to engage with.

Honeyfile – **Fake** **files** that appear to contain sensitive information, used to detect attempts to access and steal data.

Honeytoken – **False** **credentials**, login credentials, or other data types used to distract attackers, trigger alerts, and provide insight into attacker activity.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.3 Change management**

**Business processes impacting security operation** – Change management refers to a systematic approach that **manages all changes made to a product or system**, ensuring that methods and procedures are used to handle these changes efficiently and effectively. This helps minimize risks associated with the changes, ensuring they do not negatively impact the organization’s security posture, service availability or performance.

Changes typically managed in a change management program include:

Software deployments

System updates

Software patching

Hardware replacements or upgrades

Network modifications

Changes to system configurations

New product implementations

New software integrations

Changes and refreshes to support environments

If not properly manages, these changes can introduce **new vulnerabilities** into the system, **disrupt** **services** or negatively impact the organizations **compliance** **status**.

Approval process

-Typically involves several stages designed to ensure **proper** **assessment** and **approval** of change proposals. Usually begin with submitting a **request** **for** **change** (**RFC**) that outlines the **detail** **of the** **proposed** **change**, including its **purpose**, **scope**, and **potential** **impact**.

-The change request is reviewed by a designated change manager or committee that assesses its **feasibility**, **risks**, alignment with **organizational** **objectives**, and **policy** **compliance**.

-Following the initial review, the change request undergoes a **formal approval process** involving relevant stakeholders, such as management, IT teams, and any impacted departments, to ensure consensus and authorization before the change is implemented. Throughout the process, documentation and communication are crucial in tracking the status and outcome of approved changes.

Ownership – Refers to individuals or groups that are primarily responsible for implementing a specific change. Can be Project managers, team leaders, or anyone responsible for the change. Owners are accountable for ensuring that the change is implemented as planned, risks are managed effectively, and there’s a clear plan for communication and training associated with the change.

Stakeholders – Includes anyone with a **vested interes**t in the change or project being implemented or developed. The involvement of stakeholders facilitates a **comprehensive** **review** of proposed changes **helping to identify non-obvious risk** and identify effective implementation plans to minimize risk and business disruptions. Promotes **acceptance** and **adoption** of the changes because they were involved in the planning and decision-making process. Fosters **ownership** and **responsibility**, which are crucial for successful change implementation.

(EX: Employees, managers, the Change Advisory Board (CAB), customers, vendors, partners.)

**Change Management Concepts:**

Impact analysis – The process of **identifying** and **assessing** the potential implications of a proposed change, including **how the change will** **impact** individual **users**, business **processes**, or interconnected **systems**.

Test results – Before implementation, changes must first be evaluated in a **test** **environment** to ensure they work as intended and do not cause issues. Test results provide valuable insight into the likelihood of success and **help** **identify** potential **issues** **without** **impacting** business **operations**.

Backout plan – A **contingency** **plan** for reversing changes and returning systems and software to their original state **if the implementation plan fails**. A well-defined backout plan helps minimize downtime and reduces the risk of data loss or other severe impacts.

Maintenance window – A predefined, recurring **time frame** **for implementing changes**. They are typically scheduled during periods of **low** **activity** to minimize business disruptions.

Standard operating procedure (SOP) – Detailed, written **instructions** that describe how to carry out routine operations or changes. In change management, SOPs ensure that changes are implemented consistently and effectively. They are generally developed during testing phases and provide **detailed** **steps** for employees tasked with implementing a change to help reduce errors.

**Technical implications** –

Allow lists/deny list – Allow list describes a list of approved software, hardware, and specific change types that are not required to go through the entire change management process. An allow list may also include specific individuals with change management approval authority.

Deny list includes explicitly blocked software, hardware, and specific change types. Might include software / hardware with **known** **security** or **compatibility** **issues**, **high-risk** or **high**-**impact** changes that must always go through the full change management process, or individuals who are not authorized to implement or approve changes.

Restricted activities – Refer to actions or changes that require additional scrutiny, Strict controls, or higher levels or approval/authorization due to their potential impact on critical systems, sensitive data, or regulatory compliance.

Downtime – Critical considerations because they typically have a direct impact on business operations. Reconfigurations and patching changes often require restarting services or applications, leading to downtime. One of the main goals of Change Management is to minimize disruptions by scheduling restarts or downtime events during maintenance windows or off-peak times to reduce the impact on users / business processes.

Service restart – Often directly impacts business operations. Schedule for maintenance windows or off-peak times.

Application restart – Often directly impacts business operations. Schedule for maintenance windows or off-peak times.

Legacy applications – Pose a unique challenge regarding Change Management as these systems are often critical to business operations and are difficult to manage. Typically built using outdated technology, which introduces compatibility issues when implementing changes. May require “fit-gap” software or modifications to the newer components to ensure compatibility. Lack vendor support.

Dependencies – Services and applications often depend on other software, interfaces, and services to function correctly. These **dependencies** complicate changes because a service **restart in one area** may significantly **impact** **another**.

**Version control** – Refers to **tracking** and **controlling** **changes** to **documents**, **code**, or other important **data**. Organizations can use version control to **maintain** **a historical** **record** **of** **changes**, ensure only approved changes are implemented, and quickly revert changes to a previous version as warranted.

It is essential to assess how a change can impact existing policies, procedures, and diagrams. Update diagrams and documents as you make changes, and clearly label them to avoid confusion when referencing them. Archive the old documents so you may still reference them if you need.

**Documentation**

Upgrading diagrams – Typically updated whenever significant changes or modifications to a process, system, or application occur.

Upgrading policies/procedures – Changes may impact existing policies and procedures. As a result, these documents need to be reviewed and updated to ensure they align with the new processes, guidelines, or controls introduces through the change. Once document updates have been completed, the new versions should be clearly labeled, and the older versions should be archived but still available for reference.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.4 Cryptographic solutions**

**Public key infrastructure** – A framework that helps to establish trust in the use of public key cryptography to sign and encrypt messages via digital certificates. A digital certificate is a public assertion of identity, **validated by a certificate authority (CA).**

Public key – When you want others to send you confidential messages, you give them your public key to use to encrypt the message. The message can then only be decrypted by your private key, which you keep known only to yourself.

Private key – When you want to authenticate yourself to others, you sign a has of your message with your private key. You give others your public key to use to verify the signature. As only you know the private key, everyone can be assured that only you could have created the signature.

The basic problem with **public key cryptography** is that while the owner of a private key can authenticate messages, there is no mechanism for establishing the owner’s identity. How can you be sure that a shopping site or banking service is really maintained by whom it claims?

The **PKI** aims to prove that the owners of public keys are who they say they are. Under PKI, anyone issuing a public key should publish it in a **digital certificate**. The certificate’s validity is guaranteed by a certificate authority (CA).

A **digital certificate** is essentially a wrapper for a subject’s public key. As well as the public key, it contains information about the subject and the certificate’s issuer. Ther certificate is digitally signed to prove that it was issued to the subject by a particular CA.

Key escrow – If a private or secret key is lost or damaged, ciphertexts cannot be recovered unless a backup of the key has been made. Making copies of the key is problematic as it becomes more likely that a copy will be compromised and more difficult to detect that a compromise has occurred.

These issues can be mitigated by using escrow and M of N controls. Escrow means that something is held independently. In terms of key management, this refers to archiving a key with a third party. M of N means that an operation cannot be performed by a single individual. Instead, a quorum (M) of available persons (N) must agree to authorize the operation.

A key can be **split into parts**. Each part can be held by separate escrow providers, reducing the risk of compromise. An account with permission to access a key held in escrow is referred to as a key recovery agent (KRA). A recovery policy can require two or more KRAs to authorize the operation. This mitigates the risk of a KRA attempting to impersonate the key owner.

**Encryption** –

Levels – **Data at rest** encompasses a great many storage mechanisms. These can be thought of in terms of Encryption Levels. Lower levels, such as encrypting a whole disk, have the advantages of simplicity, but can be complex to manage when multiple users need to access the data. Higher levels, such as applying encryption via the file system or via a database management system can be combined with granular access controls.

Full disk encryption (FDE) – Refers to a product that encrypts the **whole contents** of a storage device, including **metadata** areas not normally accessible using ordinary OS file explorer tools. FDE also encrypts **free** **space** **areas**. FDE **primarily** **protects** **against** **physical** **theft** of the disk.

Partition – An HDD or SSD can be divided into separate logical areas called **partitions**. Each partition can be formatted with a different file system and mounted as a drive or volume in the OS. Some disk encryption products might be able to **encrypt these partitions** **selectively**, **rather than the whole disk**. The partitions could be encrypted using different keys.

File encryption – A file encryption product is software that applies to encryption to induvial files (or perhaps to folders/directories). This might depend on file system support. For example, Microsoft’s Encrypting File System (EFS) requires that the volume be formatted with NTFS.

Volume encryption – A volume is any storage resource with a single file system. A volume is the way the OS “sees” a storage resource. The technology underlying the volume might be a removable disk or a partition on an HDD or SSD. It could also be a RAID array. Consequently, a volume encryption product is likely to refer to one that is implemented as a software application rather than by disk firmware. A volume encryption product may or may not encrypt free space and/or file metadata.

Database-level encryption – Database or page-level encryption and decryption occurs when any data is **transferred** **between** **disk** and **memory**. This is referred to as Transparent Data Encryption (TDE) in SQL Server. A page is the means by which the database engine returns the data requested by a query from the underlying storage files. This type of encryption means that **all the records are encrypted while they are stored on disk**, protecting against theft of the underlying media. It also encrypts **logs** **generated** by the **database**.

Record-level encryption – Many databases contain secrets that should NOT be known by the database administrator. Public key encryption can solve this problem by storing the private key used to unlock the value of a **cell outside of the database**. Cell/column encryption is applied to one or more fields within a table. With **SQL Server’s “Always Encrypted” feature**, the data remains encrypted when loaded into memory. It is only decrypted when the client application supplies the key. This allows for **separation of duties** between the database administrator and data owner.

Asymmetric – Encryption and Decryption are performed by two **different** but related public and private keys in a key pair. When a public key is used to encrypt a message, only the paired private key can decrypt the ciphertext. The public key cannot be used to decrypt the ciphertext. The drawback of asymmetric encryption is that it involves substantial computing overhead compared to symmetric encryption.

Symmetric – Encryption and Decryption are both performed by the **same** secret key. The secret key must be kept known to authorized persons only. If the key is lost or stolen, the security is breached. Symmetric encryption is used for confidentiality. Symmetric encryption is very fast. It is used for bulk encryption of large amounts of data. **Symmetric** **encryption** **CANNOT be used for authentication or integrity.**

Transport / communication – Transport / communication encryption protects **data-in-motion**. Various transport encryption products have been developed for different networking solutions. For example:

 Wi-fi Protected Access (WPA) – Securing traffic sent over a wireless network.

Internet Protocol Security (IPsec) – Securing traffic sent between two endpoints over a public or untrusted transport network. Referred to as a VPN.

Transport Layer Security (TLS) – Securing application data, such as web or email data.

Transport encryption also uses cryptography to **ensure that integrity and authenticity** of messages, so that the recipient can verify that they were **not modified by someone other than the sender**. Integrity and authenticity checking can use a **Hash-based Message Authentication code (HMAC)**. A HMAC combines the secret key derived during key exchange with a hash of the message.

Key exchange – With **data-at-rest**, an asymmetric cipher is not typically used to encrypt the network data directly, because it is too inefficient. Transport encryption products use a system of Key exchange. This allows the **sender** and **recipient** to **exchange** a **symmetric** **encryption** key securely by using public key cryptography.

 Algorithms –

Key length – Keys for modern symmetric ciphers use a **pseudorandomly** generated number of bits. **The number of bits is the key length**.

For example: The most commonly used cipher is the Advanced Encryption Standard (**AES**). This can be used with two key lengths.

-AES-128 uses a 128-bit key length.

-A bit can have one of two values (0 or 1), so the number of possible key values is two multiplied by itself a number of times equivalent to the key length. This is written as 2128, where 2 is the base and 128 is the exponent.

-AES-256 has a keyspace of 2256. This keyspace is not twice as large as AES-128.. it is **MANY TRILLIONS of times** **bigger** and consequently significantly more resistance to brute force attacks. The drawback of using larger keys is that the computer must use more memory and processor cycles to perform encryption and decryption.

**Tools** – Can use a **Cryptoprocessor** for key generation and storage. Because it is dedicated to a single function, the cryptoprocessor hardware has a smaller attack surface than a general-purpose computer. A cryptoprocessor can also perform operations such as decryption and **signing on behalf of apps**. This means that the **key material never leaves the cryptoprocessor**. There are two main ways of implementing cryptoprocessor hardware:

Trusted platform module (TPM) – A cryptoprocessor implemented as a module within the CPU on a computer or mobile device. TPMs are produced to different version specifications, with versions **1.2** and **2.0** in current use. Version **2.0** is **NOT** backward compatible with version **1.2**.

Hardware security module (HSM) – Cryptoprocessor hardware implemented in a removable or dedicated form factor, including rack-mounted appliances, plug-in PCIe adapter cards, and USB-connected security keys. It is also possible to provision an HSM as a virtual appliance.

Secure enclave – One vulnerability in this system is that decrypted data needs to be loaded into the computer’s system memory (RAM) for applications to access it. This raises the potential for a malicious process to gain access to the data via some type of exploit. This vulnerability can be mitigated by implementing a secure enclave. A trusted execution environment (TEE) secure enclave, such as Intel Software Guard Extensions, is able to protect data stored in system memory so that an untrusted process cannot read it. A secure enclave is designed so that even processes with root or system privileges cannot access it without authorization. The enclave is locked to a list of one or more digitally signed processes.

Key management system – Some organizations prefer to centralize key generation and storage using a tool such as a key management system. For example, a dedicated server or appliance is used to generate and store keys. When a device or app needs to perform a cryptographic operation, it uses the key management interoperability protocol (KMIP) to communicate with the server.

**Obfuscation** – The art of making a message or data **difficult to find**. It is security by **obscurity**, which is normally deprecated.

Steganography – (literally means “hidden writing”) embeds information within an unexpected source. The container document or file is called the *covertext*.

Tokenization – Means that all or part of the value of a database field is replaced with a randomly generated token. The token is stored with the original value on a token server or token vault, separate from the production database. An authorized query or app can retrieve the original value from the vault, if necessary, so tokenization is reversible. Tokenization is used as a substitute for encryption because, from a regulatory perspective, an encrypted field is the same value as the original data.

(EX: A hospital might use tokenization to separate Personal Identifiable Information from their digital forms.)

Data masking – Can mean that all or part of the contents of a database field are redacted by substituting all character strings with “x”, for example. A field might be partially redacted to preserve metadata for analysis purposes. Data masking can also use techniques to preserve the original format of the field.

 **Tokenization** and **Data** **masking** are used for **de-identification**.

**Hashing** – A cryptographic hashing algorithm produces a fixed-length string of bits from an input plaintext that can be of any length. The output can be referred to as a has or as a message digest. The function is designed so that it is impossible to recover the plaintext data from the digest (one-way) and so that different inputs are unlikely to produce the same output (a collision).

A hashing algorithm is used to prove integrity. There are two popular implementations of hash algorithms:

Secure Hash Algorithm **(SHA)** – Considered the strongest algorithm. There are variants that produce different-sized outputs, with longer digest considered more secure. The most popular variant is SHA256, which produces a **256-bit** digest.

Message Digest Algorithm #5 **(MD5)** – Produces a **128-bit** digest. MD5 is not considered to be quite as safe for use as SHA256, but it might be required for compatibility between security products.

**Salting** – Cryptographic hash functions are often used for password storage and transmission. A hash cannot be decrypted back to the plaintext password that generated it. Hash functions are one way. HOWEVER, passwords stored as hashes are vulnerable to **Brute force / Dictionary attacks.**

**Brute force attack** – Simply runs through every possible combination of letters, numbers, and symbols.

**Dictionary attack** – Creates hashes of common words and phrases.

Both these attacks can be slowed down by adding a **salt** **value** when creating the hash.

(salt + password) \* SHA = hash

A unique, **random** **salt** **value** should be generated for each user account. This mitigates the risk that if users choose identical plaintext passwords, there would be identical hash values in the password file. The salt is not kept secret, because any system verifying the has must know the value of the salt. It simply means that an **attacker cannot use precomputed tables of hashes**.

(EX: Attacker couldn’t use Rainbow table, etc. to get known hashes. Salt is just adding a little extra security to your hashes.)

**Digital signatures** – **Public key** cryptography can **authenticate a sender** because they control a private key that produces messages in a way that no one else can. **Hashing** proves **integrity** by computing a unique fixed-size message digest from any variable length input. These two cryptographic ciphers can be **combined to make a digital signature**.

**Key stretching** – Takes a key that’s generated from a user password plus a random salt value and repeatedly converts it to a longer more disordered key. The initial key may be put through thousands of rounds of hashing. This might not be difficult for the attacker to replicate, so it **doesn’t actually make the key stronger**. It does **slow** **the attack** **down**, because the attacker has to do all this extra processing for each possible key value.

**Blockchain** – A concept in which an expanding list of transactional records is secured using cryptography. Each record is referred to as a **block** and is run through a hash function. The hash value of the previous block in the chain is added to the hash calculation of the next block in the chain. This ensures that each successive block is cryptographically linked. Each block validates the hash of the previous block all the way through to the beginning of the chain, ensuring that each historical transaction has not been tampered with.

**Open public ledger** – The blockchain is recorded in an Open public ledger. This ledger does not exist as an individual file on a single computer; rather, one of the most important characteristics of a blockchain is that it is decentralized. The ledger is distributed across a peer-to-peer (P2P) network in order to mitigate the risks associated with having a single point of failure or compromise. Blockchain users can therefore trust each other equally. Likewise, another defining quality of a blockchain is its openness – everyone has the same ability to view every transaction on a blockchain.

Blockchain technology has a variety of potential applications. It can ensure the integrity and transparency of financial transactions, legal contracts, copyright, and intellectual property (IP) protection, online voting systems, identity management systems, and data storage.

**Certificates** –

Certificate authorities – The **PKI** aims to prove that the owners of public keys are who they say they are. Under PKI, anyone issuing a public key should **publish it in a digital certificate**. The certificate’s validity is **guaranteed by a Certificate Authority (CA).**

Certificate revocation lists (CRL) – A certificate may be revoked or suspended by the owner or by the CA for many reasons. (EX: The private key has been compromised, the business closed, a user leaving the company, a domain name being changed, ETC). There must be a mechanism to inform users whether a certificate is valid, revoked, or suspended. A CA must maintain a CRL of all revoked and suspended certificates. The CRL must be accessible to anyone relying on the validity of the CA’s certificates. Each certificate should contain information for the browser on how to check the CRL.

 Revoked – Cert is no longer valid and cannot be un-revoked or reinstated.

 Suspended – Cert can be re-enabled.

Online certificate status protocol (OCSP) – An OCSP server provides up-to-date information to check certificate’s status. Most OCSP servers can query the certificate database directly and obtain the real-time status of a certificate.

Self-signed – In some circumstances, using **PKI can be too difficult or expensive to manage**. Any machine, web server, or program code can be deployed with a self-signed certificate. Can be **useful** in **development** and **test** **environments**. The nature of self-signed certificates makes them very **difficult** **to** **validate**. They should **NOT be used to protect critical hosts and applications**.

Third-party – PKI can use private or Third-party Cas. For **public** or **business-to-business** communications, a Third-party CA can be used to establish **a trust relationship** between servers and clients.

Third-party CAs often operate in a hierarchical model. The root CA issues certificates to one or more intermediate CAs. The intermediate CAs issue certificates to subjects (**leaf** or end entities). Each leaf certificate can be traced to the root CA along the certification path. Also referred to as **“Certificate chaining”** or **“Chain of trust”**

(EX: Comodo, DigiCert, GeoTrust, IdenTrust, Let’s Encrypt.)

Root of trust – Defines how users and different CAs can **trust one another**. Each CA issues itself a certificate. This is referred to as the **root** **certificate**. The **root certificate is self-signed**, meaning the CA server signs a certificate issued to itself.

A root certificate uses an **RSA key size** of **2048** or **4096** bits or the ECC equivalent. The subject of the root certificate is set to the organization/CA name.

The root certificate can be used to sign other certificates issued by the CA. Installing the CA’s root certificate means that host will automatically trust any certificates signed by that CA.

Certificate signing request generation (CSR) – When a subject wants to obtain a certificate, it first generates a key pair comprising private and public asymmetric keys for the chosen cipher, such as RSA or ECC, and key length. The private key must be kept well protected and known only to the subject. The subject then completes a CSR and submits it to the CA. The CSR is a file containing the information that the subject wants to use in the certificate, including its public key. The CA reviews the certificate and checks that the information is valid. If the request is accepted, the CA signs the certificate and sends it to the subject.

Wildcard – In PKI, a digital certificate that will match multiple subdomains of a parent domain. A wildcard domain, such as About\_us/Examlite.com (instead of just Examlite.com) means that the certificate issued to the parent domain will be accepted as valid for all subdomains.