Cryptographic Solutions

Objectives:

- 1.4 - Explain the importance of using appropriate cryptographic solutions
- 2.3 - Explain various types of vulnerabilities
- 2.4 - Given a scenario, you must be able to analyze indicators of malicious activity

- **Cryptographic Solutions**
  - *Cryptography*
    - Practice and study of writing and solving codes
    - Encryption to hide information's true meaning
  - *Encryption*
    - Converts plaintext to ciphertext
    - Provides data protection at rest, in transit, and in use
  - Data States
    - *Data at Rest*
      - Inactive data on storage devices
    - *Data in Transit*
      - Moving across networks
    - *Data in Use*
      - Currently undergoing change
  - Algorithm and Key
    - *Algorithm (Cipher)*
      - Performs encryption or decryption

        - *Key*
            - Essential for determining cipher output
    - Key Strength and Rotation
        - *Key Length*
            - Proportional to security
        - *Key Rotation*
            - Best practice for security longevity
    - Symmetric and Asymmetric Encryption
        - *Symmetric*
            - Uses same key for encryption and decryption
        - *Asymmetric*
            - Uses a pair of keys for encryption and decryption
    - Symmetric Algorithms
        - DES
        - Triple DES
        - IDEA
        - AES
        - Blowfish
        - Twofish
        - Rivest Cipher
    - Asymmetric Algorithms
        - Diffie-Hellman
        - RSA
        - Elliptic Curve Cryptography
    - *Hashing*
        - Converts data into fixed-size string (digest) using hash functions

        - Algorithms
            - MD5
            - SHA Family
            - RIPEMD
            - HMAC
    - *Public Key Infrastructure (PKI)*
        - Framework managing digital keys and certificates for secure data transfer
    - *Digital Certificates*
        - Electronic credentials verifying entity identity for secure communications
    - *Blockchain*
        - Decentralized, immutable ledger ensuring data integrity and transparency
    - Encryption Tools
        - TPM
        - HSM
        - Key Management Systems
        - Secure Enclave
    - Obfuscation
        - Steganography
        - Tokenization
        - Data Masking
    - Cryptographic Attacks
        - Downgrade Attacks
        - Collision Attacks
        - Quantum Computing Threats

- **Symmetric vs Asymmetric**
    - *Symmetric Encryption*
        - Uses a single key for both encryption and decryption
        - Often referred to as private key encryption
        - Requires both sender and receiver to share the same secret key
        - Offers confidentiality but lacks non-repudiation
        - Challenges with key distribution in large-scale usage
            - More people means more sharing of the keys
    - *Asymmetric Encryption*
        - Uses two separate keys
            - Public key for encryption
            - Private key for decryption
        - Often called "Public Key Cryptography"
        - No need for shared secret keys
        - Commonly used algorithms include Diffie-Hellman, RSA, and Elliptic Curve Cryptography (ECC)
        - Slower compared to symmetric encryption but solves key distribution challenges
    - *Hybrid Approach*
        - Combines both symmetric and asymmetric encryption for optimal benefits
        - Asymmetric encryption used to encrypt and share a secret key
        - Symmetric encryption used for bulk data transfer, leveraging the shared secret key
        - Offers security and efficiency
    - *Stream Cipher*
        - Encrypts data bit-by-bit or byte-by-byte in a continuous stream
        - Uses a keystream generator and exclusive XOR function for encryption

        - Suitable for real-time communication data streams like audio and video
        - Often used in symmetric algorithms
    - *Block Cipher*
        - Breaks input data into fixed-size blocks before encryption
            - Usually 64, 128, or 256 bits at a time
        - Padding added to smaller data blocks to fit the fixed block size
        - Advantages include ease of implementation and security
        - Can be implemented in software, whereas stream ciphers are often used in hardware solutions

- **Symmetric Algorithms**
    - *DES (Data Encryption Standard)*
        - Uses a 64-bit key (56 effective bits due to parity)
        - Encrypts data in 64-bit blocks through 16 rounds of transposition and substitution
        - Widely used from the 1970s to the early 2000s
    - *Triple DES (3DES)*
        - Utilizes three 56-bit keys
        - Encrypts data with the first key, decrypts with the second key, and encrypts again with the third key
        - Provides 112-bit key strength but is slower than DES
    - *IDEA (International Data Encryption Algorithm)*
        - A symmetric block cipher with a 64-bit block size
        - Uses a 128-bit key, faster and more secure than DES
        - Not as widely used as AES

- *AES (Advanced Encryption Standard)*
  - Replaced DES and 3DES as the US government encryption standard
  - Supports 128-bit, 192-bit, or 256-bit keys and matching block sizes
  - Widely adopted and considered the encryption standard for sensitive unclassified information
- *Blowfish*
  - A block cipher with key sizes ranging from 32 to 448 bits
  - Developed as a DES replacement but not widely adopted
- *Twofish*
  - A block cipher supporting 128-bit block size and key sizes of 128, 192, or 256 bits
  - Open source and available for use
- *RC Cipher Suite (RC4, RC5, RC6)*
  - Created by cryptographer, Ron Rivest
  - RC4 is a stream cipher with variable key sizes from 40 to 2048 bits, used in SSL and WEP
  - RC5 is a block cipher with key sizes up to 2048 bits
  - RC6, based on RC5, was considered as a DES replacement
- Classification
  - All the mentioned algorithms are symmetric
  - Most are block ciphers except for RC4, which is a stream cipher
- Note: When working with encryption, identify if it's symmetric or asymmetric and whether it's a block or stream cipher

- **Asymmetric Algorithms**
  - Public Key Cryptography
    - No shared secret key required

            - Uses a key pair
                - Public key for encryption
                - Private key for decryption
            - Provides confidentiality, integrity, authentication, and non-repudiation
        - Confidentiality with Public Key
            - Encrypt data using the receiver's public key
            - Only the recipient with the corresponding private key can decrypt it
        - Non-Repudiation with Private Key
            - Encrypt data using the sender's private key
            - Anyone with access to the sender's public key can verify the sender's identity
        - Integrity and Authentication with Digital Signature
            - Create a hash digest of the message
            - Encrypt the hash digest with the sender's private key
                - *Digital Signature*
                    - A hash digest of a message encrypted with the sender's private key to let the recipient know the document was created and sent by the person claiming to have sent it
            - Encrypt the message with the receiver's public key
            - Ensures message integrity, non-repudiation, and confidentiality
        - Common Asymmetric Algorithms
            - *Diffie-Hellman*
                - Used for key exchange and secure key distribution
                - Vulnerable to man-in-the-middle attacks, requires authentication
                - Commonly used in VPN tunnel establishment (IPSec)
            - *RSA (Ron Rivest, Adi Shamir, Leonard Adleman)*
                - Used for key exchange, encryption, and digital signatures

            - Relies on the mathematical difficulty of factoring large prime numbers
            - Supports key sizes from 1024 to 4096 bits
            - Widely used in organizations and multi-factor authentication
        - *Elliptic Curve Cryptography (ECC)*
            - Efficient and secure, uses algebraic structure of elliptical curves
            - Commonly used in mobile devices and low-power computing
            - Six times more efficient than RSA for equivalent security
            - Variants include
                - ECDH (Elliptic Curve Diffie-Hellman)
                - ECDHE (Elliptic Curve Diffie-Hellman Ephemeral)
                - ECDSA (Elliptic Curve Digital Signature Algorithm)

- **Hashing**
    - *Hashing*
        - One-way cryptographic function that produces a unique message digest from an input
    - *Hash Digest*
        - Like a digital fingerprint for the original data
        - Always of the same length regardless of the input's length
    - Common Hashing Algorithms
        - *MD5 (Message Digest Algorithm 5)*
            - Creates a 128-bit hash value
            - Limited unique values, leading to collisions
            - Not recommended for security-critical applications due to vulnerabilities

- SHA (Secure Hash Algorithm) Family
  - *SHA-1*
    - Produces a 160-bit hash digest, less prone to collisions than MD5
  - *SHA-2*
    - Offers longer hash digests (SHA-224, SHA-256, SHA-348, SHA-512)
  - *SHA-3*
    - Uses 224-bit to 512-bit hash digests, more secure, 120 rounds of computations
- *RIPEMD (RACE Integrity Primitive Evaluation Message Digest)*
  - Versions available
    - 160-bit (Most common)
    - 256-bit
    - 320-bit
  - Open-source competitor to SHA but less popular
- *HMAC (Hash-based Message Authentication Code)*
  - Checks message integrity and authenticity
  - Utilizes other hashing algorithms (e.g., HMAC-MD5, HMAC-SHA1, HMAC-SHA256)

- *Digital Signatures*
  - Uses a hash digest encrypted with a private key
  - Sender hashes the message and encrypts the hash with their private key
  - Recipient decrypts the digital signature using the sender's public key
  - Verifies integrity of the message and ensures non-repudiation
- Common Digital Signature Algorithms
  - *DSA (Digital Security Algorithm)*
    - Utilized for digital signatures

            - Uses a 160-bit message digest created by DSS (Digital Security Standard)
        - *RSA (Rivest-Shamir-Adleman)*
            - Supports digital signatures, encryption, and key distribution
            - Widely used in various applications, including code signing
    - Hashes change drastically even with minor changes in input
    - Hashing is used to verify data integrity and detect any changes

- **Increasing Hash Security**
    - Common Hashing Attacks
        - *Pass the Hash Attack*
            - A hacking technique that allows the attacker to authenticate to a remote server or service by using the underlying hash of a user's password instead of requiring the associated plaintext password
            - Hashes can be obtained by attackers to impersonate users without cracking the password
            - Difficult to defend against due to various Windows vulnerabilities and applications
            - Penetration tools like Mimikatz automate hash harvesting
            - Prevention
                - Ensure trusted OS
                - Proper Windows domain trusts
                - Patching
                - Multi-factor authentication
                - Least privilege

- *Birthday Attack*
  - Occurs when two different messages result in the same hash digest (collision)
  - Named after the Birthday Paradox, where shared birthdays become likely in a group
  - Collisions in hashes can be exploited by attackers to bypass authentication systems
  - Use longer hash output (e.g., SHA-256) to reduce collisions and mitigate the attack
- Increasing Hash Security
  - *Key Stretching*
    - Technique that is used to mitigate a weaker key by creating longer, more secure keys (at least 128 bits)
      - increases the time needed to crack the key
    - Used in systems like Wi-Fi Protected Access, Wi-Fi Protected Access version 2, and Pretty Good Privacy
  - *Salting*
    - Adds random data (salt) to passwords before hashing
    - Ensures distinct hash outputs for the same password due to different salts
    - Thwarts dictionary attacks, brute-force attacks, and rainbow tables
  - *Nonces (Number Used Once)*
    - Adds unique, often random numbers to password-based authentication processes
    - Prevents attackers from reusing stolen authentication data
    - Adds an extra layer of security against replay attacks

- Limiting Failed Login Attempts
  - Restricts the number of incorrect login attempts a user can make
  - Increases security by deterring attackers attempting to guess passwords
  - Typically, lock the account after three incorrect attempts

- **Public Key Infrastructure (PKI)**
  - PKI Components
    - An entire system involving hardware, software, policies, procedures, and people
    - Based on asymmetric encryption
    - Facilitates secure data transfer, authentication, and encrypted communications
    - Used in HTTPS connections on websites
  - Establishing a Secure Connection
    - User connects to a website via HTTPS
    - Web browser contacts a trusted certificate authority for the web server's public key
    - A random shared secret key is generated for symmetric encryption
    - The shared secret is securely transmitted using public key encryption
    - The web server decrypts the shared secret with its private key
    - Both parties use the shared secret for symmetric encryption (e.g., AES) to create a secure tunnel
  - Security Benefits
    - Confidentiality
      - Data is encrypted using a shared secret
    - Authentication
      - The web server's identity is verified using its private key

        - Visual indicators like a padlock show secure communication
    - Public Key Infrastructure vs. Public Key Cryptography
        - *Public Key Infrastructure (PKI)*
            - Encompasses the entire system for managing key pairs, policies, and trust
            - Involves generating, validating, and managing public and private key pairs that are used in the encryption and decryption process
            - Ensures the security and trustworthiness of keys
        - *Public Key Cryptography*
            - Refers to the encryption and decryption process using public and private keys
            - Only a part of the overall PKI architecture
    - *Key Escrow*
        - Storage of cryptographic keys in a secure, third-party location (escrow)
        - Enables key retrieval in cases of key loss or for legal investigations
        - Relevance in PKI
            - In PKI, key escrow ensures that encrypted data is not permanently inaccessible
            - Useful when individuals or organizations lose access to their encryption keys
        - Security Concerns
            - Malicious access to escrowed keys could lead to data decryption
            - Requires stringent security measures and access controls

- **Digital Certificates**
    - *Digital Certificates*
        - Digitally signed electronic documents

        - Bind a public key with a user's identity
        - Used for individuals, servers, workstations, or devices
        - Use the *X.509 Standard*
            - Commonly used standard for digital certificates within PKI
            - Contains owner's/user's information and certificate authority details
    - Types of Digital Certificates
        - *Wildcard Certificate*
            - Allows multiple subdomains to use the same certificate
            - Easier management, cost-effective for subdomains
            - Compromise affects all subdomains
        - *SAN (Subject Alternate Name) field*
            - Certificate that specifies what additional domains and IP addresses are going to be supported
            - Used when domain names don't have the same root domain
        - Single-Sided and Dual-Sided Certificates
            - *Single-sided*
                - Only requires the server to be validated
            - *Dual-sided*
                - Both server and user validate each other
                - Dual-sided for higher security, requires more processing power
        - *Self-Signed Certificates*
            - Digital certificate that is signed by the same entity whose identity it certifies
            - Provides encryption but lacks third-party trust
            - Used in testing or closed systems
        - *Third-Party Certificates*

- Digital certificate issued and signed by trusted certificate authorities (CAs)
- Trusted by browsers and systems
- Preferred for public-facing websites

- Key Concepts
  - *Root of Trust*
    - Highest level of trust in certificate validation
    - Trusted third-party providers like Verisign, Google, etc.
    - Forms a certification path for trust
  - *Certificate Authority (CA)*
    - Trusted third party that issues digital certificates
    - Certificates contain CA's information and digital signature
    - Validates and manages certificates
  - *Registration Authority (RA)*
    - Requests identifying information from the user and forwards certificate request up to the CA to create a digital certificate
    - Collects user information for certificates
    - Assists in the certificate issuance process
  - *Certificate Signing Request (CSR)*
    - A block of encoded text with information about the entity requesting the certificate
    - Includes the public key
    - Submitted to CA for certificate issuance
    - Private key remains secure with the requester
  - *Certificate Revocation List (CRL)*
    - Maintained by CAs
    - List of all digital certificates that the certificate authority has already

revoked

- Checked before validating a certificate

▪ *Online Certificate Status Protocol (OCSP)*

- Determines certificate revocation status or any digital certificate using the certificate's serial number
- Faster but less secure than CRL

▪ *OCSP Stapling*

- Alternative to OCSP
- Allows the certificate holder to get the OCSP record from the server at regular intervals
- Includes OCSP record in the SSL/TLS handshake
- Speeds up the secure tunnel creation

▪ *Public Key Pinning*

- Allows an HTTPS website to resist impersonation attacks from users who are trying to present fraudulent certificates
- Presents trusted public keys to browsers
- Alerts users if a fraudulent certificate is detected

▪ *Key Escrow Agents*

- Securely store copies of private keys
- Ensures key recovery in case of loss
- Requires strong access controls

▪ *Key Recovery Agents*

- Specialized type of software that allows the restoration of a lost or corrupted key to be performed
- Acts as a backup for certificate authority keys

○ Trust in Digital Certificates

        - Trust is essential in digital certificates
        - Compromised root CAs can impact all issued certificates
        - Commercially trusted CAs are more secure
        - Self-managed CAs must be vigilant against compromises

- **Blockchain**
    - *Blockchain*
        - Shared immutable ledger for transactions and asset tracking
        - Builds trust and transparency
        - Widely associated with cryptocurrencies like Bitcoin
        - Is essentially a really long series of information with each block containing information in it
            - Each block has the hash for the block before it
        - Block Structure
            - Chain of blocks, each containing
                - Previous block's hash
                - Timestamp
                - Root transactions (hashes of individual transactions)
            - Blocks are linked together in a chronological order
        - *Public Ledger*
            - Secure and anonymous record-keeping system
            - Maintains participants' identities
            - Tracks cryptocurrency balances
            - Records all genuine transactions in a network
    - Blockchain Applications
        - *Smart Contracts*

      - Self-executing contracts with code-defined terms
      - Execute actions automatically when conditions are met
      - Transparent, tamper-proof, and trust-enhancing
    - Commercial Uses
      - Companies like IBM promote blockchain for commercial purposes
      - Permissioned blockchain used for business transactions
      - Enhances trust and transparency with immutable public ledger
    - *Supply Chain Management*
      - Transparency and traceability in the supply chain
      - Immutable records of product origin, handling, and distribution
      - Ensures compliance and quality control
  - Broad Implications of Blockchain
    - Versatility
      - Beyond finance and cryptocurrencies
      - Applications across various industries
      - Promises transparency, efficiency, and trust
    - Decentralization
      - Key feature of blockchain
      - Eliminates need for central authorities
      - Empowers peer-to-peer networks
    - Immutable Ledger
      - Ensures data integrity
      - Records cannot be altered or deleted
      - Reinforces trust in transactions and information
    - Digital Evolution
      - Blockchain's impact on technology and industries

- Potential to reshape traditional systems
- Offers transparency, efficiency, and trust in the digital era

- **Encryption Tools**
  - Encryption Tools for Data Security
    - *TPM (Trusted Platform Module)*
      - Dedicated microcontroller for hardware-level security
      - Protects digital secrets through integrated cryptographic keys
      - Used in BitLocker drive encryption for Windows devices
      - Adds an extra layer of security against software attacks
    - *HSM (Hardware Security Module)*
      - Physical device for safeguarding and managing digital keys
      - Ideal for mission-critical scenarios like financial transactions
      - Performs encryption operations in a tamper-proof environment
      - Ensures key security and regulatory compliance
    - *Key Management System*
      - Manages, stores, distributes, and retires cryptographic keys
      - Centralized mechanism for key lifecycle management
      - Crucial for securing data and preventing unauthorized access
      - Automates key management tasks in complex environments
    - *Secure Enclaves*
      - Coprocessor integrated into the main processor of some devices
      - Isolated from the main processor for secure data processing and storage
      - Safeguards sensitive data like biometric information
      - Enhances device security by preventing unauthorized access

- **Obfuscation**
  - Obfuscation Techniques in Data Security
    - *Steganography*
      - Conceals a message within another to hide its very existence
      - Involves altering image or data elements to embed hidden information
      - Primary goal is to prevent the suspicion that there's any hidden data at all
      - Used alongside encryption for added security
      - Detection is challenging due to hiding data in plain sight
    - *Tokenization*
      - Substitutes sensitive data with non-sensitive tokens
      - Original data securely stored elsewhere
      - Tokens have no intrinsic value
      - Reduces exposure of sensitive data during transactions
      - Commonly used for payment systems to comply with security standards
    - *Data Masking (Data Obfuscation)*
      - Disguises original data to protect sensitive information
      - Maintains data authenticity and usability
      - Used in testing environments, especially for software development
      - Reduces the risk of data breaches in non-production settings
      - Common in industries handling personal data
      - Masks portions of sensitive data for privacy, e.g., credit card digits, social security numbers

- **Cryptographic Attacks**
  - *Cryptographic Attacks*

- Techniques and strategies that adversaries employ to exploit vulnerabilities in cryptographic systems with the intent to compromise the confidentiality, integrity, or authenticity of data

- *Downgrade Attacks*
  - Force systems to use weaker or older cryptographic standards or protocols
  - Exploit known vulnerabilities or weaknesses in outdated versions
  - Example: POODLE attack on SSL 3.0
  - Countermeasures include phasing out support for insecure protocols and version-intolerant checks
- *Collision Attacks*
  - Find two different inputs producing the same hash output
  - Undermine data integrity verification relying on hash functions
  - Vulnerabilities in hashing algorithms, e.g., MD5, can lead to collisions
  - Birthday Paradox or Birthday Attack
    - The probability that two distinct inputs, when processed through a hashing function, will produce the same output, or a collision
- *Quantum Computing Threat*
  - *Quantum computing*
    - A computer that uses quantum mechanics to generate and manipulate quantum bits in order to access enormous processing powers.
    - Uses quantum bits (qubits) instead of using ones and zeros
  - *Quantum Communication*
    - A communications network that relies on qubits made of photons (light) to send multiple combinations of ones and zeros simultaneously which results in tamper resistant and extremely fast communications
  - *Qubit*

        - A quantum bit composed of electrons or photons that can represent numerous combinations of ones and zeros at the same time through superposition
        - Enable simultaneous processing of multiple combinations
    - Quantum computing is designed for very specific use cases
        - Complex math problems
        - Trying to do something like the modeling of an atom or atomic structure
    - Threat to traditional encryption algorithms (RSA, ECC) by rapid factorization of large prime numbers
    - *Post-quantum cryptography*
        - A new kind of cryptographic algorithm that can be implemented using today's classic computers but is also impervious to attacks from future quantum computers
        - Aims to create algorithms resistant to quantum attacks
        - First method is to create post-quantum cryptography is to increase the key size
            - Increases the number of permutations that are needed to be brute-forced
        - Second method is to create something like lattice-based cryptography and super singular isogeny key exchange
    - NIST selected four post-quantum cryptography standards
        - CRYSTALS-Kyber - general encryption needs
        - Digital signatures
            - CRYSTALS-Dilithium
            - FLACON
            - SPHINCS+