Digital Signature Standard

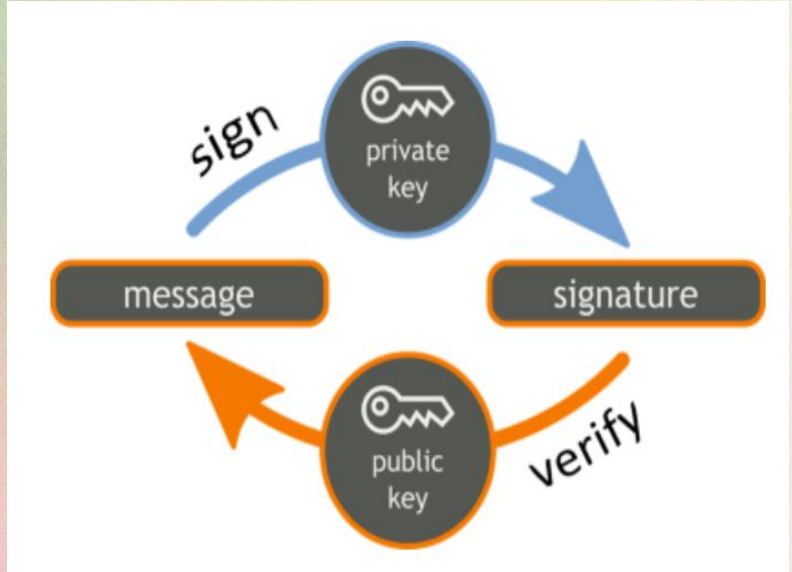
- Geetha P K - 22z219
Mithra K M - 22z238
Moumitha K - 22z241
Sruthi S - 22z264
Tharigalakshmi S - 22z268

Introduction

- Mithra K M 22z238

Digital Signature

- A **digital signature** is an electronic code attached to a message or document.
- Used for message authentication.
- It binds a **person or entity** to digital data, just like a handwritten signature binds a person to a document.
- It ensures that the **message truly comes from the sender** and **cannot be denied or altered**.
- Widely used in **business and legal applications** to prevent disputes.



A digital signature scheme works using **three main algorithms**:

1. **Key Generation** – Creates a pair of keys:
    a. **Private key** (kept secret)
    b. **Public key** (shared for verification)
2. **Signing Algorithm** – Uses the private key to generate a **unique signature** for a message.
3. **Verification Algorithm** – Uses the public key to **check the authenticity** of the message and signature.

Need for Digital Signature:

- Increasing shift from paper-based to electronic communication.
- Need to ensure **trust** in digital transactions.
- Prevent **forgery** and **tampering** of electronic data.
- Provide a way to **verify sender identity** and **document authenticity**.
- Enable **legal acceptance** of e-documents.

Digital Signature Standard:

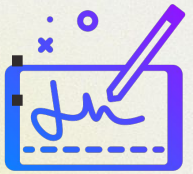
- A **Digital Signature Standard (DSS)** is a **federal standard** for creating and verifying digital signatures.
- Developed by **NIST (National Institute of Standards and Technology)**.
- Specifies algorithms like:

  DSA (Digital Signature Algorithm)

  RSA

  ECDSA (Elliptic Curve DSA)

Purpose for Digital Signature Standard:



To establish a **standardized method** for digital signing and verification.

To ensure:

- **Integrity** → data unchanged
- **Authenticity** → trusted sender
- **Non-repudiation** → legal proof

To enable **trusted electronic communication** across systems.

Requirements, Security Objectives

- Tharigalakshmi S 22z268

Requirements for the Digital Signature Standard

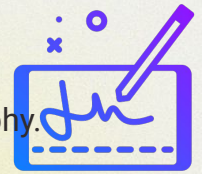
The Digital Signature Standard (DSS) is a standard that defines the approved cryptographic methods for generating and verifying digital signatures. The core requirements center on **cryptographic algorithms, Public Key Infrastructure (PKI), and key management** practices that ensure the security and validity of digital signatures.

Cryptographic requirements – specific algorithms (RSA, ECDSA, EdDSA), hash functions (SHA), random number requirements.

Public Key Infrastructure (PKI) – certificate authorities, validation, and identity binding.

Key and Certificate Management – storage, key rotation, auditing, and split knowledge.

Cryptographic requirements



The foundation of the DSS is a suite of approved cryptographic algorithms that use asymmetric (public-key) cryptography.

- **Asymmetric key pair:**
  Each user has two keys — a **private key** for creating signatures and a **public key** for verifying them. The private key must always be kept secret and secure.

- **Approved algorithms:**
  DSS uses **FIPS-approved algorithms** as listed in the latest standard (FIPS 186-5):

  RSA (Rivest–Shamir–Adleman): Uses the difficulty of factoring large prime numbers.

  ECDSA (Elliptic Curve Digital Signature Algorithm): Offers strong security with smaller key sizes, suitable for devices with limited power.

  EdDSA (Edwards Curve Digital Signature Algorithm): A fast and efficient elliptic curve algorithm.

- **Secure hash function:**
  A **Secure Hash Algorithm (SHA)** must be used to generate a fixed-length message digest of the data. The signature is applied to this digest, ensuring that any change to the data invalidates the signature.

- **Unique random number for signing:**
  Some algorithms, like **DSA**, require a new, random number for each signature to make sure every signature is unique, even if the same message is signed multiple times

Public Key Infrastructure (PKI) requirements

The standard requires a supporting infrastructure to manage and distribute the cryptographic components securely.

Digital certificates:

A digital certificate securely links a person's public key to their identity. It is issued by a trusted organization called a Certificate Authority (CA).

Certificate validation:

Before use, the system checks if the certificate is valid — not expired or cancelled by the CA.

Secure binding:

The Public Key Infrastructure (PKI) makes sure there is a safe and trusted connection between a user's identity and their public key.

Key and certificate management requirements

For a digital signature to remain trustworthy, the entire lifecycle of its cryptographic keys and certificates must be managed securely.

- **Secure generation and storage:** Private keys should be created using strong, approved cryptographic methods and stored safely to avoid any misuse or leakage.

- **Dual control and split knowledge:** In sensitive systems, no single person should have full control of a key. Dual control means two or more people handle key-related tasks together. Split knowledge ensures no one person knows the entire key value.

- **Key rotation:** Keys should be changed regularly and retired after a specific time period to maintain security.

- **Auditable processes:** All key management actions — generation, distribution, storage, and deletion — should be recorded and follow formal security policies.

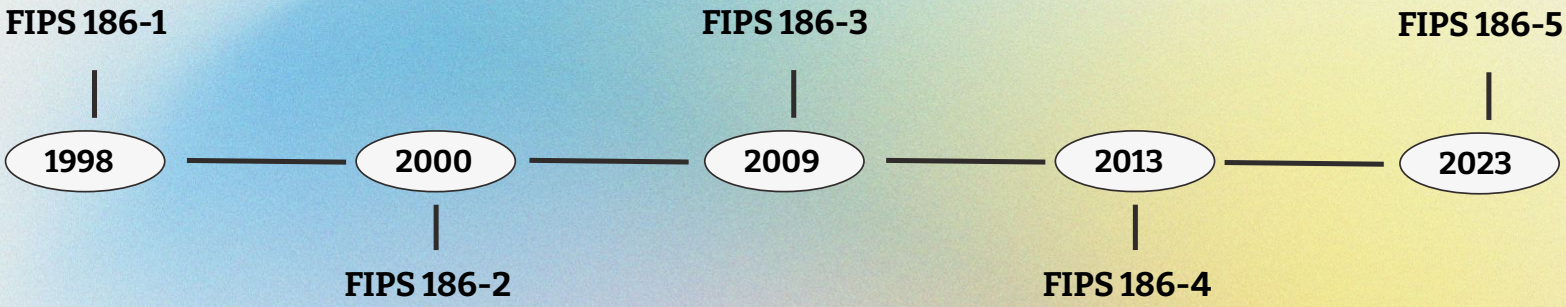
Objectives of Digital Signature Standard

- **Authentication:** Verifies the signer's identity using their private key and builds trust that the message is from a genuine source.
- **Integrity:** Ensures the data has not been altered after signing by using a secure hash function.
- **Non-repudiation:** Prevents the signer from denying their digital signature or the act of signing the document.
- **Timestamping:** Digital signatures can include a trusted timestamp to prove when the signature was created, even if the certificate expires later.
- **Auditing:** Signed documents create a record of all signing activities, helping with compliance and providing evidence for audits or legal purposes.
- **Efficiency:** DSS provides a standard, legally recognized way for electronic transactions, reducing errors and speeding up business processes.

DSS Evolution and Revisions

- Sruthi S 22z264

DSS Evolution Timeline



FIPS - Federal Information Processing Standards

- Developed by the U.S. National Institute of Standards and Technology (NIST).
- Standards for secure information handling.
- DSS is one such FIPS that defines how digital signatures are created and verified.

FIPS 186 → Digital Signature Standard (DSS)

FIPS 186-1 (1998)

What it is:

- First revision of DSS
- Refined DSA parameters and generation process

Features:

- Clear guidelines for prime numbers (p, q) and generator (g)
- Deterministic key generation
- Improved interoperability and security

FIPS 186-2 (2000)

What it is:

- Expanded DSS to include RSA and ECDSA

Features:

- Multiple algorithm support for flexibility
- Promoted elliptic curve cryptography (ECC)
- Algorithm agility for performance and security

FIPS 186-3 (2009)

What it is:

- Increased cryptographic strength
- Support for SHA-2 family

Features:

- Larger DSA key sizes (up to 3072 bits)
- SHA-2 hash support (SHA-224, SHA-256, SHA-384, SHA-512)
- Additional elliptic curves and stronger security

FIPS 186-4 (2013)

What it is:

- Enhanced ECC framework and key management

Features:

- Flexible key generation and validation
- Stronger parameter selection guidance
- Improved random number generation

FIPS 186-5 (2023)

What it is:

- Most recent revision
- Focused on modern cryptography and post-quantum considerations

Features:

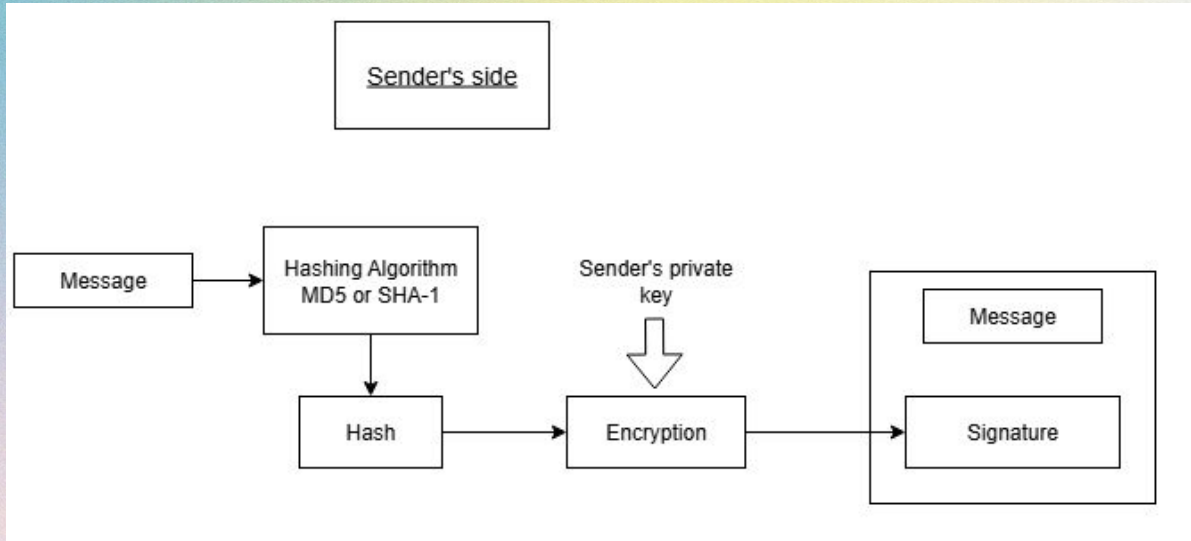
- Deprecated DSA
- Reinforced RSA and ECDSA
- Updated guidelines for secure implementation
- Alignment with post-quantum security research

Digital Signature - Signing and Verification Process

- Moumitha K 22z241

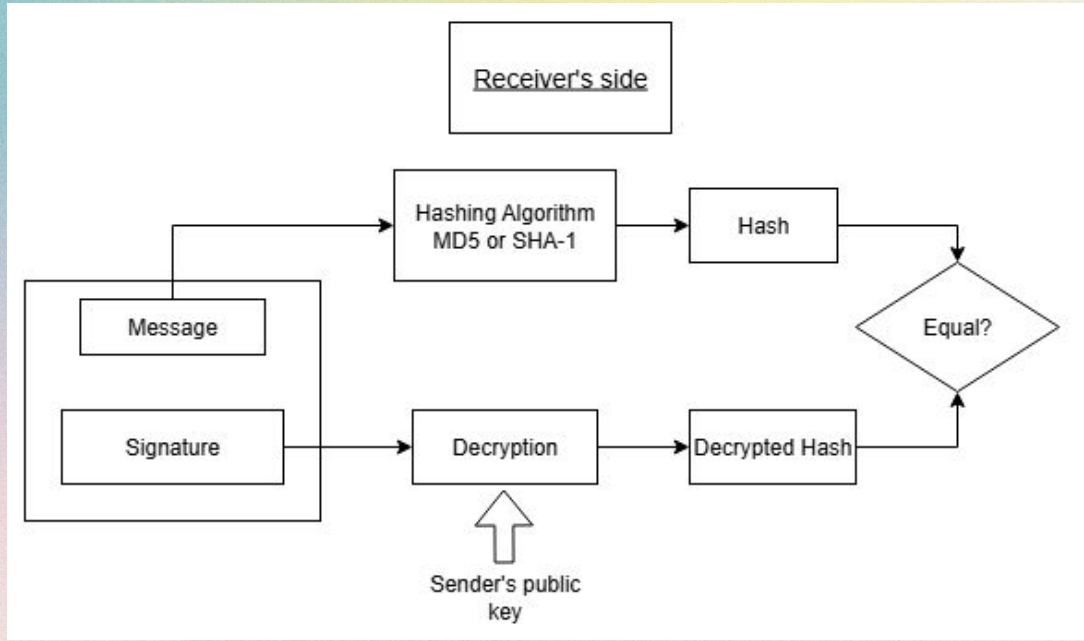
Signature Production (Sender Side)

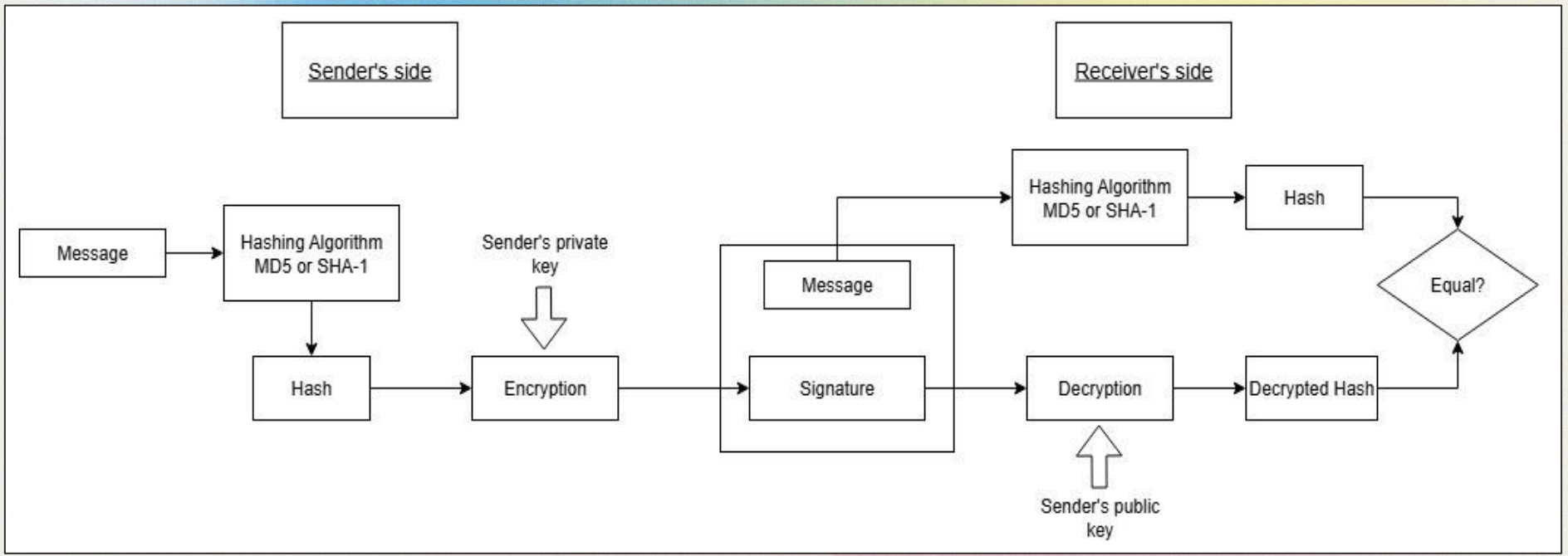
- Original message is taken
- Hash value of the message is generated (using SHA-1 or MD5 algorithm).
- Hash is encrypted using sender's private key
- Encrypted hash = Digital Signature
- Message + Signature are sent to receiver



Signature Verification (Receiver Side)

- Receiver gets message + signature
- Decrypts signature using sender's public key → gets original hash
- Creates a new hash from the received message
- Compares both hash values
- If both are equal → Signature is valid




Sender's side
Receiver's side
Message
Hashing Algorithm
MD5 or SHA-1
Hash
Sender's private
key
Encryption
Message
Signature
Hashing Algorithm
MD5 or SHA-1
Hash
Equal?
Decryption
Decrypted Hash
Sender's public
key

Connection to Security Objectives

Integrity:
- ➔ Verified by comparing both hash values.
- ➔ If hashes match, message content is unchanged.

Authentication:
- ➔ Achieved by using the sender's public key for verification.
- ➔ Confirms the sender's identity.

Non-Repudiation:
- ➔ The sender cannot deny sending the message.
- ➔ Only their private key could have created that signature.

Challenges, Applications

- Geetha PK 22z219

Challenges in Digital Signature Standard

Key Management is tricky :
Users must protect their private keys securely; if a key is lost or stolen, signatures become invalid.

- Even minor carelessness, like storing keys on shared devices, can compromise the entire system.

Compatibility and Integration issues :
Different organizations may use various signature algorithms or formats.

- This makes it difficult for systems to verify each other's signatures without proper standardization.

Algorithm strength and technological evolution :
Cryptographic algorithms can weaken over time as computing power increases.

- For example, older DSS versions using SHA-1 had to be replaced with SHA-2 and SHA-3 for stronger protection.

Legal and regional recognition:

While some countries legally recognize digital signatures, others still prefer traditional methods.

- This inconsistency affects international transactions and document validation across borders.

Implementation and system design issues :

Incorrect or incomplete implementation of DSS can lead to vulnerabilities and verification errors.

- Proper training and quality testing are crucial to avoid technical loopholes.

User awareness and trust :

Many users still do not understand how digital signatures work or why they are safe.

- Lack of awareness can lead to misuse, reduced adoption, and lower trust in the system.

Applications

E-Governance and Public Services :
Governments use DSS for online services like e-tenders, tax filing, and citizen applications.

- It ensures only authorized users submit or approve official documents.

Email and Document Authentication :
Digital signatures verify the sender's identity and confirm that the document hasn't been changed.

- This is crucial in legal, academic, and corporate communications where data integrity matters.

Banking and Financial Transactions:
DSS secures online payments, loan approvals, and fund transfers.

- It guarantees non-repudiation — meaning users can't deny their transactions later.

Software and Code Signing :

Developers sign their software releases to assure users that they are genuine and unaltered.

- This helps prevent the spread of malicious or tampered software updates.

Legal and Business Documentation:

Contracts, agreements, and invoices are signed digitally with legal validity equal to handwritten ones.

- This saves time, reduces paperwork, and supports remote or international business operations.

Healthcare and Medical Systems:

DSS secures patient records, electronic prescriptions, and medical reports.

- It ensures confidentiality, prevents unauthorized modifications, and supports telemedicine safely.

Thank You