**DIGITAL SIGNATURES**

CRYPTOGRAPHY Course Project

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**RIDHI GUNTUR – AP22110011467**

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[11, 2024]**

1. **INTRODUCTION**
2. **WHAT IS DIGITAL SIGNATURE?**

Digital signatures can be viewed as a specific mathematical method which helps validate the authenticity and integrity of the email, message, document or even software, that is being sent from one person to another. It is quite similar to signing a real document or stamping it, but there is a lot more security available with a digital signature.

Digital signatures are incredibly useful as they serve the purpose of providing the evidence of the origin, the identity and the status of the electronic documents, transactions as well as messages sent across in a digital format. Signers are able to use them to show informed consent was given and understood.

1. **BENEFITS OF DIGITAL SIGNATURES**

Digital signatures offer the following advantages:

* **Security:** Digital signatures are incorporated with security functions to protect a legal document from being modified and ensure the authenticity of the signatures. These include asymmetric encryption, PIN numbers, checksum and CRC, CA and TSP validation among other features.
* **Timestamping:** This simply gives the date and time at which the digital signature was created and is especially important in situations where there is a time constraint for instance news press releases, stock exchange trades, lottery ticket numbers and court issues.
* **Saving time:** The process of signing, storing and exchanging hard copies of documents which often takes too long is eliminated by the use of digital signatures allowing easy retrieval and signing of the documents by the organizations.

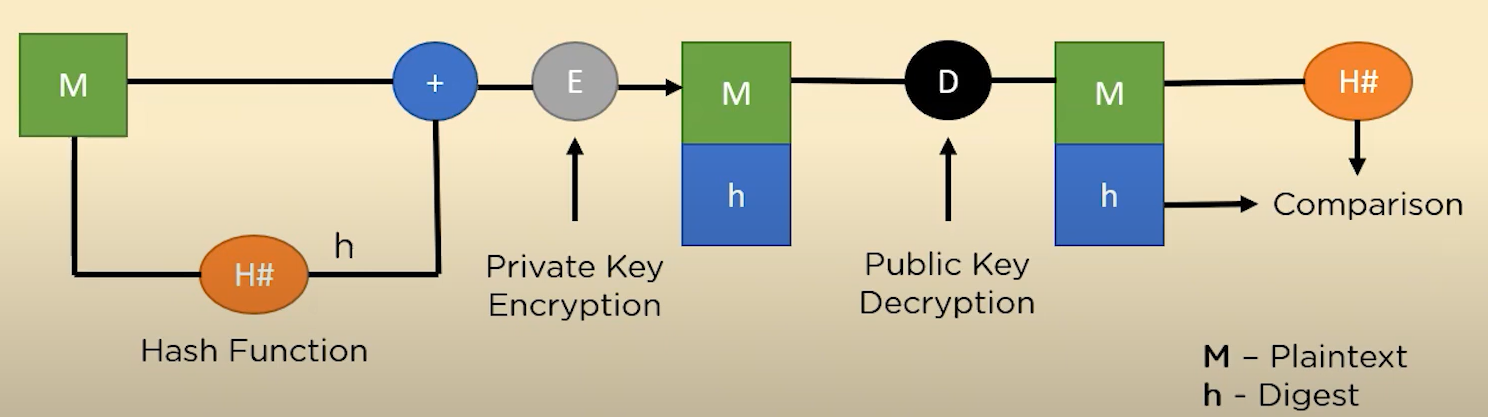
1. **HISTORY**

The concept of a digital signature scheme was first put forth by Whitfield Diffie and Martin Hellman in 1976. However, they merely speculated on the existence of such schemes as based on certain functions that are trap-door one-way permutations. In the subsequent period, Ronald Rivest, Adi Shamir, and Len Adleman created what is known as the RSA algorithm, which proved to be primitive in producing digital signatures (although as a proof of the concept - “plain” RSA signatures are insecure).

1. **CRYPTOGRAPHY IN DIGITAL SIGNATURES**
2. **DIGITAL SIGNATURE**

Digital signatures are the primitives of message authentication that use public keys. Handwritten signatures are frequently used on handwritten or typed texts in the real world.   
  
In a similar vein, a digital signature is a method that links an individual or organization to digital information. The recipient or any other third party may independently confirm this binding.   
  
Data and a secret key that only the signer knows are used to compute the cryptographic value of a digital signature.   
  
In the real world, the recipient of a message need confirmation that it is belonging to the sender and that he cannot deny its provenance. Because there is a very high chance of a dispute over data sharing, this need is particularly important in business applications.

1. **MODEL OF DIGITAL SIGNATURE**



The operation of a digital signature is depicted in this diagram as:

* The plaintext message (M) is the first step in the procedure. A hash function generates a distinct message digest (h) that protects data integrity.
* The digital signature, which is created by encrypting the digest with the sender's private key (E), is subsequently transmitted with the original message.
* The digest (h) is obtained on the recipient's end by decrypting the digital signature with the sender's public key (D).
* Using the same hash function, the recipient re-hashed the plaintext to create a new digest (H#), which it then compared to the decrypted digest.
* The message's integrity and validity are confirmed if the two digests agree.

1. **PURPOSE**

A Digital Signature server’s 3 purposes:

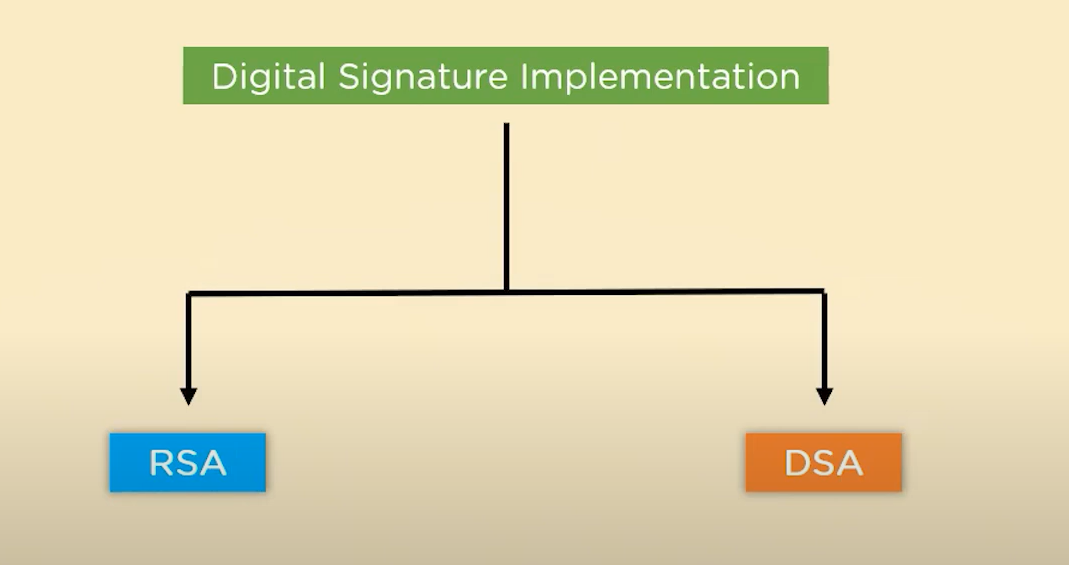
* **Authentication:** A Digital Signature gives the receiver reason to believe the message was created and sent by the claimed sender.
* **Non-Reputation:** With Digital Signature, the sender cannot delay having sent the message later on.
* **Integrity:** A Digital Signature ensures that the message was not altered in transmit.

1. **TYPES OF DIGITAL SIGNATURES**

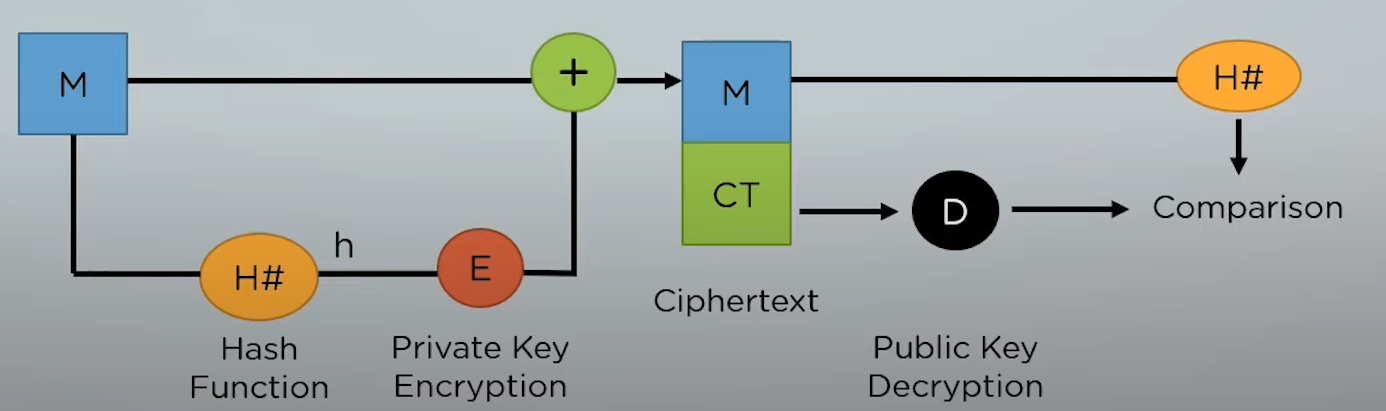
* **Simple Digital Signature:** A simple digital signature is a digital signature that isn't encrypted. Legally, this kind of digital signature is not recommended. Because it is impossible to identify the signer's origin, it is not secure.
* **Basic Digital Signature:** A simple digital signature is essentially the same as a basic digital signature. They lack both legal enforcement and authority. With this kind of digital signature, the sender's identity is not accurately confirmed. Even after a digital document has been digitally signed, its content may still be changed. This kind of digital signature is not secure because it is not constrained by any cryptographic technique or verification procedure.
* **Advanced and Qualified Digital Signature:** The most secure digital signature is Advanced & Qualified, which has the same legal force as a wet signature on paper. Public key infrastructure and asymmetric cryptography technology are used to create sophisticated and qualified digital level signatures. Advanced and qualified digital level signatures can also indicate which devices to use, when to use them, and where to use them, much like a basic category digital signature. Any modifications made after the document is signed can also be readily identified.

1. **DIGITAL SIGNATURE IMPLEMANTATION**

Digital Signature can be implemented in two ways:



* 1. **RSA**One popular cryptographic technique for protecting digital communication is the RSA algorithm. It is founded on the ideas of asymmetric encryption, which uses two different keys: a public key and a private key. To create a digital signature, a message is signed using the private key, and verification is done using the public key.   
       
     RSA combines hash algorithms and encryption to guarantee data authenticity and integrity. Large prime numbers are hard to factor, which makes it computationally impossible for attackers to crack the encryption or signatures.



The diagram outlines the use of the RSA algorithm in digital signatures:

* A message digest (h) of fixed length is produced by processing the message (M) through a hash function.
* To create the digital signature, the digest is encrypted using the sender's private key (E). The recipient then receives both the original message and the encrypted digest, or ciphertext.
* The digest is revealed at the recipient's end when the digital signature is decoded using the sender's public key (D).
* In the meantime, the recipient creates a fresh digest (H#) by applying the same hash function to the message.
* The freshly calculated digest and the decrypted digest are compared. If they coincide, the integrity and validity of the message are validated.
  + 1. **Steps**

1. **Key Generation:**
2. Two large prime numbers are chosen (p and q).
3. Compute n=p×q and z=(p−1) (q−1).
4. Choose a number e where 1 < e < (p−1) (q−1).
5. A number d is selected so that ed mod  z=1 and calculated as d={"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msup><mi>e</mi><mrow><mo>-</mo><mn>1</mn></mrow></msup></mstyle></math>","origin":"MathType for Microsoft Add-in"}mod  (p−1)(q−1).
6. Public key is (n, e) and private key is (q, d).
7. **Encryption and Decryption Function:**
8. If the plaintext is mmm, encrypted ciphertext ccc is calculated as:  
    c={"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msup><mi>m</mi><mi>e</mi></msup></mstyle></math>","origin":"MathType for Microsoft Add-in"}mod  n
9. Under similar assumptions, the plaintext can be calculated as:  
    m={"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msup><mi>c</mi><mi>d</mi></msup></mstyle></math>","origin":"MathType for Microsoft Add-in"} mod  n
   * 1. **Key Features and Applications of RSA**
10. **Advantages of RSA**:
    * High level of security due to large key sizes.
    * Widely used and trusted in various encryption and signature protocols.
11. **Limitations of RSA**:

* Slower performance compared to modern algorithms (e.g., ECC).
* Requires large key sizes for enhanced security, increasing computational cost.

1. **Applications of RSA**:

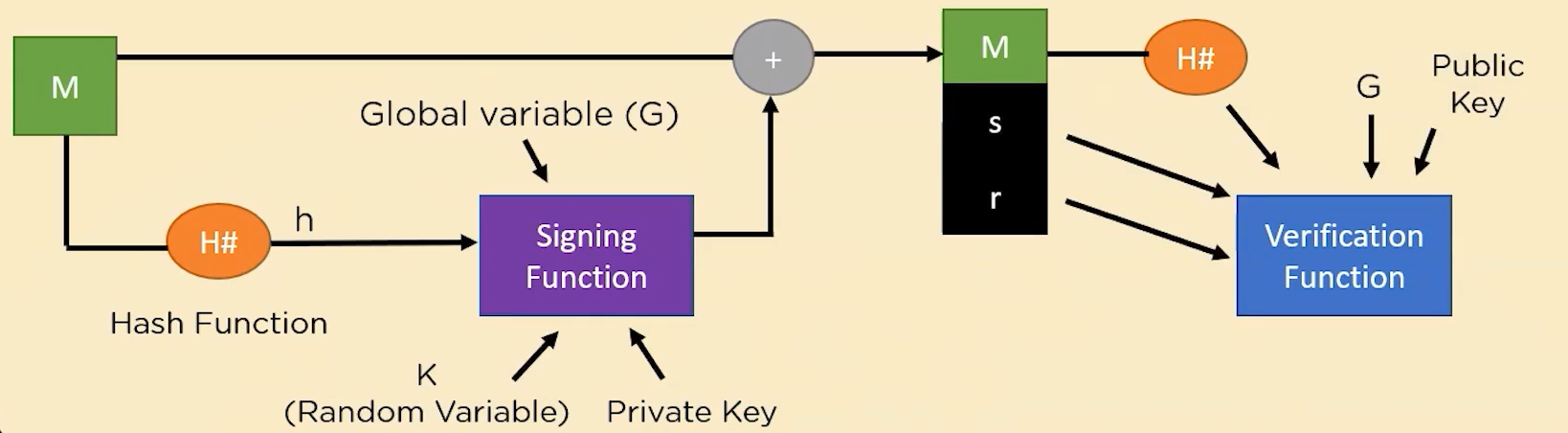
* Secure transmission in email protocols like PGP.
* Used in SSL/TLS for secure web communication.
  1. **DSA**

A cryptographic method called the Digital Signature Algorithm (DSA) is used to guarantee the integrity and validity of digital communications. It works by using the sender's private key to create a distinct digital signature for a message. It is based on public key cryptography. The sender's public key is used by the recipient to validate the signature.

DSA works in two phases:

1. **Signing Phase**: To generate the signature components, the message is hashed to form a digest, which is then concatenated with the private key and a random number.

2. **Verification Phase**: To confirm the signature, the recipient re-hashed the message and used the public key, global parameters, and signature components.   
  
This verifies the sender's identity and makes sure the message hasn't been tampered with. DSA is frequently utilized in software distribution and secure communications, such as email.



The diagram explains the working of the **Digital Signature Algorithm (DSA)**:

* In order to create a message digest (h), the message (M) is first hashed using a hash function.
* The signing function generates a signature with two components: r and s using the digest, a global variable (G), the sender's private key, and a random variable (K) (predefined parameters).
* The recipient receives both the original message and the signature (r, s).
* To confirm the signature, the recipient employs the verification function, which depends on the global variable (G), the sender's public key, and the message hash (H#).
* If the signature is legitimate, it attests to the message's authenticity and the sender's identity.
  + 1. **Steps:**

1. **Key Generation**
2. q -> prime divisor
3. p ->p-1 mod q = 0
4. x -> (private key) -> random integer such that: 0 < x < q
5. y ->(public key) -> can be calculated as: y={"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msup><mi>g</mi><mi>x</mi></msup></mstyle></math>","origin":"MathType for Microsoft Add-in"} mod p
6. G ->any interger that is g^q mod p = 1
7. **Signature Generation**
8. Message is passed through a hash function to generate a digest (h).
9. Choose any random integer k such that: 0 < k < q.
10. To calculate the value of r:

( {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msup><mi>g</mi><mi>k</mi></msup></math>","origin":"MathType for Microsoft Add-in"} mod p) mod q

1. To calculate the value of s:

[{"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msup><mi>K</mi><mrow><mo>-</mo><mn>1</mn></mrow></msup><mspace linebreak=\"newline\"/></math>","origin":"MathType for Microsoft Add-in"}(h+x.R) mod q]

The Signature can be packaged as {r,s}.

1. **Signature Verification**
2. Calculate the message digest using the same hash function.
3. Compute the value of w such that:

s⋅w mod q=1

1. Compute the value of u1 as:

u1=h⋅w mod q

1. Compute the value of u2 as:

u2=r⋅w mod q

1. Finally, the verification component v:

v=[[({"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msup><mi>g</mi><mrow><mi>u</mi><mn>1</mn></mrow></msup><mo>.</mo><msup><mi>y</mi><mrow><mi>u</mi><mn>2</mn></mrow></msup></math>","origin":"MathType for Microsoft Add-in"}) mod p) mod q]

If v==rv , the signature verification is successful.

* + 1. **Key Features and Applications of DSA**

1. **Advantages of DSA**:

* Provides message authenticity and integrity.
* Efficient for verification in high-traffic systems.

1. **Limitations of DSA**:

* Relies on secure key management (compromised keys can undermine security).
* Requires random number generation, which could affect security if weak.

1. **Applications of DSA**:

* Used in secure email protocols (e.g., S/MIME).
* Plays a role in blockchain technology and digital certificates.

1. **CONCLUSION**

Since they offer authenticity, integrity, and non-repudiation, digital signatures are essential to the security of digital communication. This research explored the fundamental ideas and real-world applications of digital signatures, emphasizing how beneficial they are for maintaining security and confidence in online transactions.   
  
The distinct advantages and applications of RSA and DSA are shown in a key comparison. A popular and adaptable cryptographic method, RSA is preferred for its ease of use and dual use in digital signing and encryption. It is frequently seen to be simpler to implement and ideal for general-purpose applications. However, DSA, which was created especially for digital signatures, is computationally efficient and provides speedier signature production. However, because of its extensive testing and longer history, RSA is typically seen as being more secure and adaptable.

1. **REFERNCES**

* <https://www.youtube.com/watch?v=bO4lEQfPn60>
* <https://www.geeksforgeeks.org/digital-signature-algorithm-dsa/>
* <https://www.techtarget.com/searchsecurity/definition/digital-signature>
* <https://en.wikipedia.org/wiki/Digital_signature>