|  |  |
| --- | --- |
| Course: | INFO-6001-23W |
| Project: | Assignment #3 |
| Due Date: | 26/01/2023 |
| Submitting: |  |
| Professor: | Steve Spencer |
| Student Name / No: | Gihan Shamike Liyanage / 1142109 |

1. Message Authentication Code (MAC), Keyed-Hash Message Authentication Code (HMAC), and Secure Hash Algorithm 1 (SHA-1) are all methods for ensuring the integrity and authenticity of data. However, they are used in different ways and have different properties.

**MAC** is a code that produced from a secret key and a message. This key is used to confirm that the input message has not been interfere. The secret key will be shared between the sender and receiver and it use to encrypt and decrypt the MAC. Because the same key is used to generate and verify the MAC, it is considered a symmetric method.

**HMAC** is similar to a MAC, but instead of a single function, it uses a hash function (as SHA-1) in blended with a private key. This key is utilized to generate a unique "digest" of the message, which is then included with the message. Then the recipient can use the same unique key to recreate the digest and compare it with the received one. Because the digest is unique to the specific message and key, it is considered more secure than a regular MAC.

**SHA-1** is a cryptographic hash function. It gets an input and generates a fixed-size output, called a "digest." The digest is a distinctive interpretation of the input message, and any change over to the input message will result in a various digest. SHA-1 is used to make sure the integrity of a message text by comparing the digest of the received message to the digest of the original message. It can not be decrypted by any secret key like the MAC and HMAC. SHA-1 is considered to be less secure and it is highly recommended to use newer algorithms such as SHA-256 or SHA-3.

In summary:

MAC: A code that is created from an input text and a secret key and it use to confirm that the received message has not been interfered with.

HMAC: A hash message authentication code is a method uses a hash function (as SHA-1) with a secret key to generate digest.

SHA-1: A one-way cryptographic hash function that produces a unique digest of an input message and it used to confirm the reliability of a message.

1. Elliptic curve is a method for creating a set of points that can be used for encryption and digital signature. The mathematical foundation of elliptic curve is based on the properties of elliptic curves over finite fields, which is a curve defined by an equation of the form y2 = x3 + ax + b, where a and b are constants. The set of places on the curve, along with a special point called the "point at infinity," form a group.

In cryptography, elliptic curves are used to create a set of points that can be used for encryption and digital signature. The encryption process works by choosing a point on the curve and a large number (called the "private key"), and then performing a series of mathematical operations to generate a second point (called the "public key"). The public key can be freely shared, while the private key must be kept secret. To encrypt a message, the dispatcher uses the recipient's public key to perform a series of mathematical operations on the message, resulting in a ciphertext. Then the receiver can use his private key to perform the reverse operations and decrypt the message.

In digital signature, the process is similar, but it's used for authenticity and integrity. The sender uses their private key to perform a series of mathematical operations on a message, resulting in a signature. The receiver can then use the source's public key to confirm the signature and verify that the message came from the source and has not been hindered.

A current implementation of elliptic curve is Elliptic Curve Digital Signature Algorithm (ECDSA). It's widely used in various cryptographic protocols such as SSL/TLS, PGP, SSH, S/MIME, and many others. The most widely used elliptic curve in ECDSA is the NIST P-256 curve, also known as secp256r1. This curve is standardized and widely used in many cryptographic applications.

Diagram

Description automatically generated with low confidence

* MITM attacks: Also known as Man-In-The-Middle and in this method an attacker captures a message and modifies it prior to forwarding it on to the planned receiver, thereby compromising the digital signature.
* Phishing: attackers may try to scam users into signing and authorizing a document or clicking on a link that appears legitimate, but actually redirects them to a fake website. Other thing is attackers may use various tactics to trick users into revealing their private key or signing a document without realizing its true content.
* Malware: attackers may use malware to steal private keys or intercept signed documents by compromising the digital signatures.
* Weak or stolen credentials: if a user's private key is stolen or their password is guessed, an attacker may use these credentials to sign documents on their behalf.
* Compromised CA: if a CA (certificate authority) that issues and validate digital certificates is compromised, attackers may be able to obtain fraudulent certificates, thus compromising the digital signature.

5.

**Using the Euclidean algorithm**

Step 1: 589432167 ÷ 3498762 = 168 remainder 817095

Step 2: 3498762 ÷ 817095 = 4 remainder 285667

Step 3: 817095 ÷ 285667 = 2 remainder 249528

Step 4: 285667 ÷ 249528 = 1 remainder 36139

Step 5: 249528 ÷ 36139 = 68 remainder 36120

Step 6: 36139 ÷ 36120 = 1 remainder 19

Step 7: 36120 ÷ 19 = 1890 remainder 0

Since the remainder in step 7 is 0, this means the GCD of 589432167 and 3498762 is 19.

Therefore, the GCD (589432167, 3498762) = 19

Reference

*An introduction to elliptic curve cryptography*. (n.d.). Retrieved from An introduction to elliptic curve cryptography: https://www.embedded.com/an-introduction-to-elliptic-curve-cryptography/

*elliptic-curve-cryptography*. (n.d.). Retrieved from elliptic-curve-cryptography: http://hitk.acm.org/elliptic-curve-cryptography-by-riddhiman-dasgupta-and-anirban-ghose/

*Enhancing Data Encryption using Elliptic Curve Cryptography (ECC) Algorithm in 4G Networks*. (n.d.). Retrieved from Enhancing Data Encryption using Elliptic Curve Cryptography (ECC) Algorithm in 4G Networks: https://www.semanticscholar.org/paper/Enhancing-Data-Encryption-using-Elliptic-Curve-in-Osman/92dbddc9c5f8e785ec624dd90f5f7a62487cf0fd

*How To Use GPG to Encrypt and Sign Messages*. (n.d.). Retrieved from How To Use GPG to Encrypt and Sign Messages: https://sysreseau.net/use-gpg-to-encrypt-and-sign-messages/