CYBER SECURITY & CRYPTOGRAPHY (6)

techworldthink • March 11, 2022

15. Compare HMAC and CMAC protocol with suitable diagrams.

Message Authentication Code

Message authentication codes (MACs) are commonly used in electronic funds transfers (EFTs) to maintain information integrity. They confirm that a message is authentic; that it really does come, in other words, from the stated sender, and hasn't undergone any changes en route. A verifier who also possesses the key can use it to identify changes to the content of the message in question.

Three algorithms typically comprise a MAC: a key generation algorithm, a signing algorithm and a verifying algorithm. The key generation algorithm chooses a key at random. The signing algorithm sends a tag when given the key and the message. The verifying algorithm is used to verify the authenticity of the message when given the key and tag; it will return a message of *accepted* if the message and tag are authentic and unaltered, but otherwise, it will return a message of *rejected*.

HMAC

HMAC algorithm stands for Hashed or Hash-based Message Authentication Code. It is a result of work done on developing a MAC derived from cryptographic hash functions. HMAC is a great resistance towards cryptanalysis attacks as it uses the Hashing concept twice. HMAC consists of twin benefits of Hashing and MAC and thus is more secure than any other authentication code. RFC 2104 has issued HMAC, and HMAC has been made compulsory to implement in IP security. The FIPS 198 NIST standard has also issued HMAC.

Objectives –

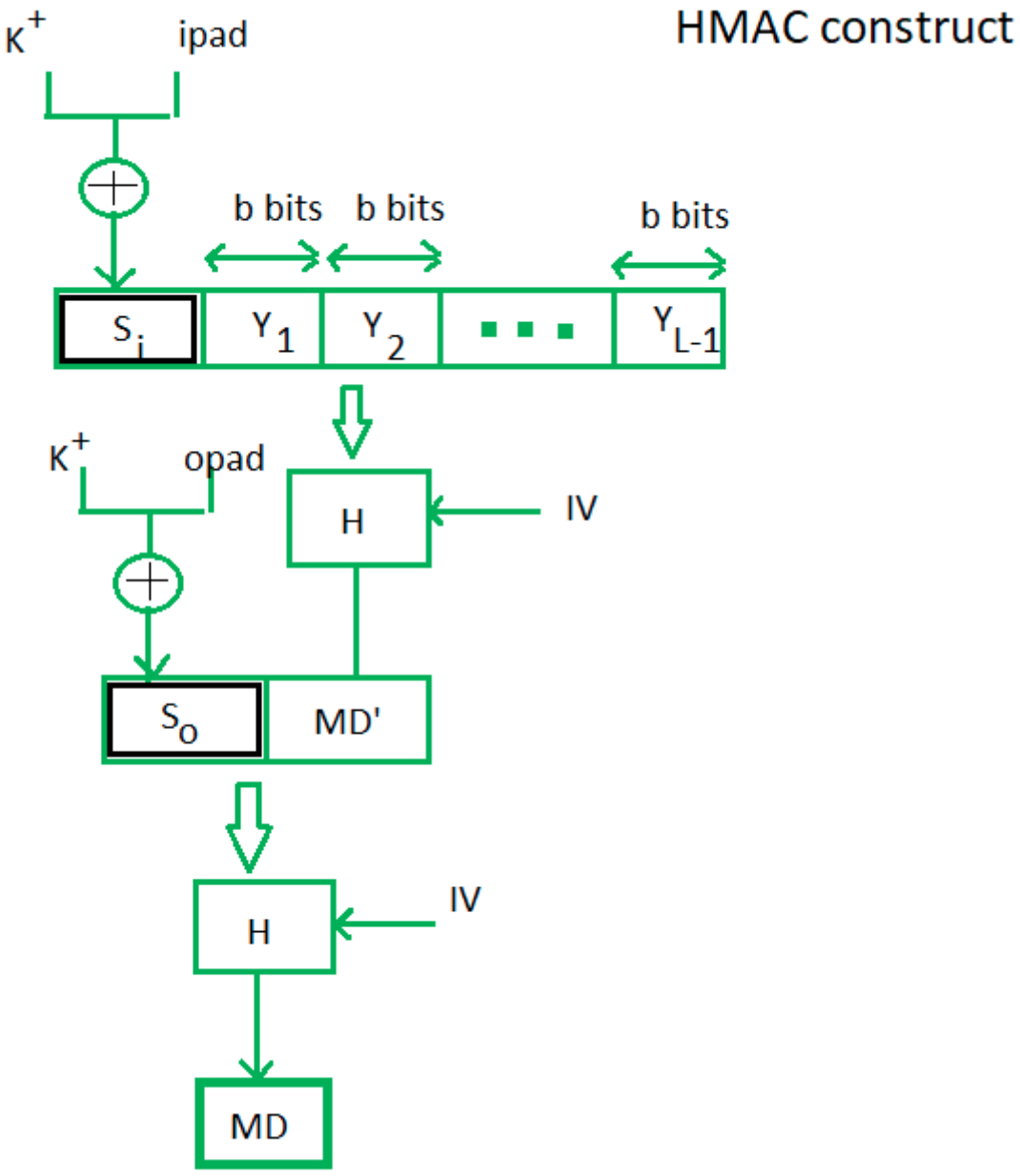
- As the Hash Function, HMAC is also aimed to be one way, i.e, easy to generate output from input but complex the other way round.
- It aims at being less affected by collisions than the hash functions.

- HMAC reuses the algorithms like MD5 and SHA-1 and checks to replace the embedded hash functions with more secure hash functions, in case found.
- HMAC tries to handle the Keys in a more simple manner.

HMAC algorithm –

The working of HMAC starts with taking a message M containing blocks of length *b* bits. An input signature is padded to the left of the message and the whole is given as input to a hash function which gives us a temporary message-digest MD'. MD' again is appended to an output signature and the whole is applied a hash function again, the result is our final message digest MD.

Here is a simple structure of HMAC:



Here, H stands for Hashing function,

M is the original message

Si and So are input and output signatures respectively,

Yi is the ith block in original message M, where I ranges from [1, L)

L = the count of blocks in M

K is the secret key used for hashing

IV is an initial vector (some constant)

The generation of input signature and output signature *Si* and *So* respectively.

$S_i = K^+ \oplus ipad$ where K^+ is nothing but K padded with zeros on the left so that the result is b bits in length

$S_o = K^+ \oplus opad$ where ipad and opad are 00110110 and 01011100 respectively taken b/8 times repeatedly.

$MD' = H(S_i \,||\, M)$

$MD = H(S_o \,||\, MD')$ or $MD = H(S_o \,||\, H(S_i \,||\, M))$

To a normal hash function, HMAC adds a compression instance to the processing. This structural implementation holds efficiency for shorter MAC values.

HMAC Structure

1. The message is divided into L blocks, each of b bits
2. The secret key K+ is left-padded with o's to create a b-bit key. It is recommended that the secret key , before padding be longer than n-bits, where n is the size of the HMAC.

3. The result of step 2 is Exclusive-ORed with a constant called **i-pad** (input pad) to create a b-bit block. The value of i-pad is the b/8 repetition of the sequence 00110110 (36 in hex).
4. The resulting block is prepended to the L-block message.
5. The result of step-4 is hashed to create an n-bit digest. We call this as **intermediate HMAC**.
6. The intermediate HMAC is left-padded with o's to make a b-bit block.
7. Step 3 is repeated with a different constant **o-pad** (output pad). The value of o-pad is the b/8 repetition of the sequence 0101 1100 (5C in hex).
8. The result of step-7 is prepended to the result of step-6.
9. The result of step-8 is hashed with the same hashing algorithm to create the final n-bit HMAC.

CMAC

Cipher-Based Message Authentication Code (CMAC)

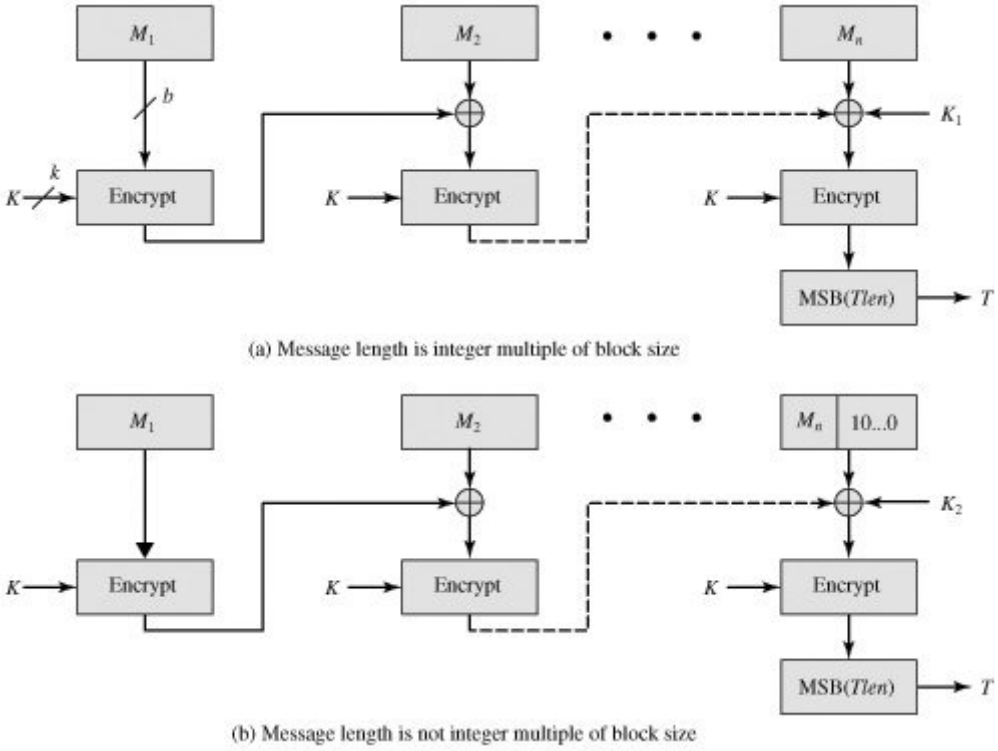
widely used in government and industry

but has message size limitation

can overcome using 2 keys and padding

thus forming the Cipher-based Message Authentication Code (CMAC)

dopted by NIST SP800-38B



In cryptography, a cipher block chaining message authentication code (CBC-MAC) is a technique for constructing a message authentication code from a block cipher. The message is encrypted with some block cipher algorithm in CBC mode to create a chain of blocks such that each block depends on the proper encryption of the previous block. This interdependence ensures that a change to any of the plaintext bits will cause the final encrypted block to change in a way that cannot be predicted or counteracted without knowing the key to the block cipher. To calculate the CBC-MAC of message m, one encrypts m in CBC mode with zero initialization vector and keeps the last block. The following figure sketches the computation of the CBC-MAC of a message comprising blocks m1 || m2||....|| mx using a secret key k and a block cipher E

16. Compare various signature schemes with suitable diagrams.

The conventional handwritten signature on a document is used to certify that the signer is responsible for the content of the document. The signature is physically a part of the document and while forgery is certainly possible, it is difficult to do so convincingly. Trying to mimic a handwritten signature in a digital medium leads to a difficulty since cut and paste operations can be used to create a perfect forgery. Thus, we need to have a way of signing messages digitally which is functionally equivalent

to a physical signature, but which is at least as resistant to forgery as its physical counterpart.

Schemes which provide this functionality are called Digital Signature Schemes. A Digital Signature Scheme will have two components, a private signing algorithm which permits a user to securely sign a message and a public verification algorithm which permits anyone to verify that the signature is authentic. The signing algorithm needs to "bind" a signature to a message in such a way that the signature can not be pulled out and used to sign another document, or have the original message modified and the signature remain valid. For practical reasons it would be necessary for both algorithms to be relatively fast and if small computers such as smart cards are to be used, the algorithms can not be too computationally complex. There are many Digital Signature Schemes which meet these conditions, but we shall only investigate a few of the most popular ones.

RSA signature

Digital Signature Standard

ElGamal schemes

RSA signature

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography :

- A client (for example browser) sends its public key to the server and requests for some data.
- The server encrypts the data using the client's public key and sends the encrypted data.
- Client receives this data and decrypts it.

Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of browser.

Digital signatures are used to verify the authenticity of the message sent electronically. A digital signature algorithm uses a public key system. The intended transmitter signs his/her message with his/her private key and the intended receiver verifies it with the transmitter's public key. A digital signature can provide message authentication, message integrity and non-repudiation services.

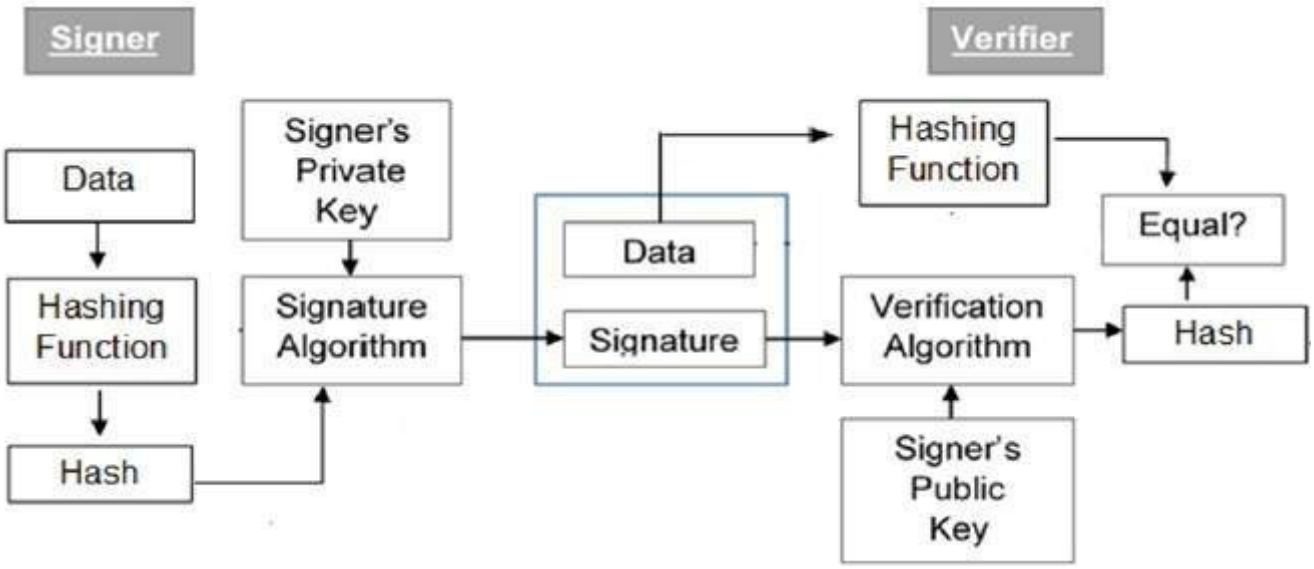
Algorithm

RSA Key Generation:

- Choose two large prime numbers p and q
- Calculate n=p*q
- Select public key e such that it is not a factor of (p-1)*(q-1)
- Select private key d such that the following equation is true (d*e)mod(p-1)(q-1)=1 or d is inverse of E in modulo (p-1)*(q-1)

RSA Digital Signature Scheme: In RSA, d is private; e and n are public.

- Alice creates her digital signature using S=M^d mod n where M is the message
- Alice sends Message M and Signature S to Bob
- Bob computes M1=S^e mod n
- If M1=M then Bob accepts the data sent by Alice.



ElGamal Signature Scheme

ElGamal Signature Scheme The ElGamal digital signature scheme stems from the ElGamal cryptosystem based upon the security of the one-way function of exponentiation in modular rings and the difficulty of solving the discrete logarithm problem.

Signing protocol

The initial setup is the same as that for ElGamal encryption.

1. Alice chooses a large prime p and a primitive root α.

2. She then chooses a secret integer z and calculates $\beta \equiv \alpha \hat{} z \pmod{p}$.

3. The values of p, α and β are made public and z is kept private.

In order to sign the message m Alice follows the steps below:

1. She selects a secret random integer k such that $\gcd(k, p-1) = 1$.

2. She then computes $r \equiv \alpha \hat{} k \pmod{p}$.

3. She then finally computes $s \equiv k \hat{} {-1} (m - zr) \pmod{p-1}$.

4. The signed message is the triplet (m, r,s).

The verification process can then be performed by Bob using public information only.

1. Bob computes $v1 \equiv \beta \hat{} \, r \cdot \, r \hat{} \, s \pmod{p}$ and $v2 \equiv \alpha \hat{} m \pmod{p}$.

2. The signature is declared valid if and only if $v1 \equiv v2 \pmod{p}$.

The Digital Signature Algorithm

(DSA) In 1991 the National Institute of Standards and Technology proposed the Digital Signature Algorithm as a standardized general use secure signature scheme. Like the ElGamal scheme DSA is a digital signature scheme with an appendix

meaning that the message cannot be easily recovered from the signature itself. When using DSA it is usually a hash of the message that is signed. A hash is an encoding that reduces the size of a message to a fixed bit length by using a specific encoding. Let the hash of a message m be $h(m) = x$ and let x be a 160-bit message.

https://www.commonlounge.com/discussion/35a1c2baa00b447f9275e8f71b02ef29#elgamal-signature-scheme