Assignment
of
Network Security
(CEN - 805)

Submitted To

Dr. Mohd Amjad

Submitted By

Name : Mehedi Hasan

Roll No. : 19BCS082

Class : B.Tech in Computer Engineering

Semester : 8th

Date of Submission : 1st May, 2023.

Mehedi Hasan
19BCS082

Assignment

T1: Explain the following with respect to MD5:

i. If the size of the message is 1000 bits, what will be the size of padding bits?

Answer:

We need to add padding bits to make the length a multiple of 512 bits

$$(|M| + |P| + 64) = 0 \bmod 512$$

$$|P| = (-|M| - 64) \bmod 512$$

$$|P| = (-1000 - 64) \bmod 512 = -1064 \bmod 512$$
$$= 472.$$

ii. Function to generate temporary constant

$$K_i = \text{abs}(\sin(1+i)) \times 2^{32}$$

$$i \in [1, 64]$$

1 to 16 are used in round 1 and successive rounds use successive 16 values.

iii. Values of chaining variables

Since MD5 has 128 bit hash value 432 bits variables are initiated

$A = 0X01234567$ $\quad$ $B = 0X89abcdef$

$C = 0Xfedca98$ $\quad$ $D = 0X76543210$

These are called chaining variables.

Mehedi Hasan
19BCS082

iv. Compression Function

A 128 bit buffer (4 registers, 32 bit each) is used to hold the intermediate and final result of hash function.

We define four auxiliary functions that each take as input three-32 bit words and produce as output one 32-bit word.

$$F(X, Y, Z) = XY \vee \text{not}(X)\, Z$$
$$G(X, Y, Z) = XZ \vee Y\, \text{not}(Z)$$
$$H(X, Y, Z) = X \oplus Y \oplus Z$$
$$I(X, Y, Z) = Y \oplus (X \vee \text{not}(Z))$$

In each bit position F acts as a conditional: If X then Y else Z. Function F could use + instead of $\vee$ since XY and not(X)Z will never have 1's in the same bit position.
Function G, H & I are bitwise parallel to the function F.

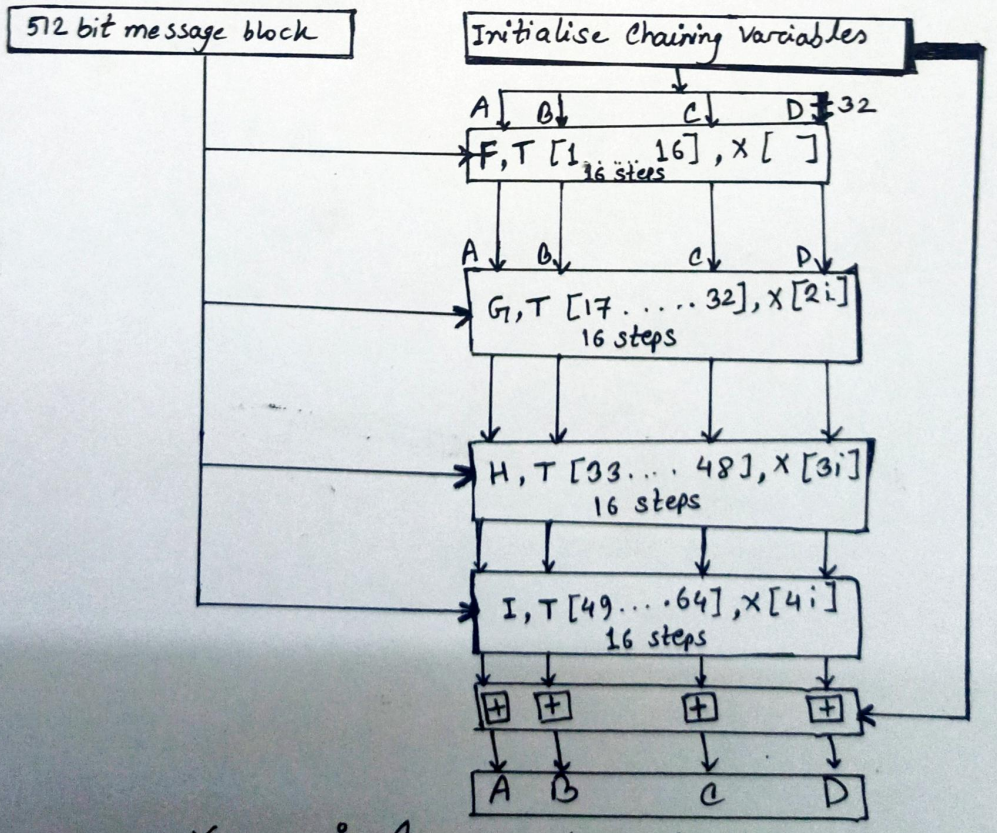

Figure: Compression Function.

Mehedi Hasan
19BCS082

T2: For what purpose the Message Digest is used? Explain the process of padding and chaining variables used in SHA-1.

Answer:

A message digest is a cryptographic hash function containing a string of digits created by a one-way hashing formula.

Message digests are designed to protect the integrity of a piece of data or media to detect changes and alterations to any part of a message. They are a type of cryptography utilizing hash values that can warn the copyright owner of any modifications applied to their work.

Message digest hash numbers represent specific files containing the protected works. One message digest is assigned to particular data content. It can reference a change made deliberately or accidentally, but it prompts the owner to identify the modification as well as the individual(s) making the change. Message digests are algorithmic numbers.

The term is also known as a hash value and sometimes as a checksum.

(P.T.O)

Mehedi Hasan
19BCS082

Padding and chaining are important concepts in the SHA-1 algorithm, which is a widely used cryptographic hash function that generates a fixed-length output (160 bits) from an arbitrary-length input message.

Padding :

The padding step in SHA-1 involves adding additional bits to the end of the input message so that its length is a multiple of 512 bits. The padding is done in such a way that it is unambiguously reversible, so that the original message can be reconstructed.

The padding process involves appending a single "1" bit to the end of the message, followed by as many "0" bits as necessary to bring the length of the message to 64 bits less than a multiple of 512 bits. Finally, the length of the original message is appended as a 64-bit integer in big-endian format, so that the final length of the message is a multiple of 512 bits.

Mehedi Hasan
19BCS082

Chaining:

The chaining step in SHA-1 involves using the output of the previous block as an input to the current block. This ensures that any change in the input message results in a completely different output. The chaining process involves using a set of five variables to maintain the intermediates hash value during the computation.

The five variables used in SHA-1 are:

- A, B, C, D, E: These variables are 32-bit words that hold the intermediate hash value. They are initialized with fixed values (specified by the SHA-1 standard) before processing the message.
- W[t]: This variable is a 32-bit word that holds the expanded message block. The message block is divided into 16 32-bit words, and these words are expanded into 80 32-bit words using a specific algorithm.

During the computation, each 512-bit block of the input message is processed in a series of 80 rounds. In each round, the current values of A, B, C, D, and E are updated based on the current value of W[t] and the intermediate hash value computed in the previous round. After all 80 rounds are completed,

Mehedi Hasan
19BCS082

the final intermediate hash value (H) is obtained by adding the original values of A, B, C, D, and E to the updated intermediate hash values.

The final output of SHA-1 is a 160-bit hash value, which is obtained by concatenating the final values of A, B, C, D, and E in big-endian format. This hash value is considered to be a unique and secure representation of the input message.

Mehedi Hasan
19BCS082

T3: For SHA-512,

i. Show the equation for the values of W_{60} and W_{69}.

ii. Find the value of padding field and the value of length field if the length of the message is 2000 bits long.

Answer:

i. $W_i = W_{i-16} \oplus \text{RotShift}_{1-8-7}(W_{i-15}) \oplus W_{i-7} \oplus \text{RotShift}_{19-61-6}(W_{i-2})$

$W_{60} = W_{44} \oplus \text{RotShift}_{1-8-7}(W_{45}) \oplus W_{53} \oplus \text{RotShift}_{19-61-6}(W_{58})$

$W_{69} = W_{53} \oplus \text{RotShift}_{1-8-7}(W_{54}) \oplus W_{62} \oplus \text{RotShift}_{19-61-6}(W_{67})$.

ii. $|\text{Padding}| = (-2000 - 128) \bmod 1024$

$= (-2128) \bmod 1024 = 944$

The padding consists of one 1 followed by 943 0's.

Value of length field = length of original message

$= 2000$.

Mehedi Hasan
19BCS082

T4: In SHA-512, one block of message consisting of three ASCII characters "abc" which is equivalent to the following 24 bit binary string.

01100001 01100010 01100011

Find the value of words $W_0, W_1, W_2, W_3 \ldots W_{15}$ in HEX.

Answer:

We can recall from step 1 of the SHA algorithm, that the message is padded to a length congruent to 896 module 1024. In this case of a single block, the padding consists of 896 − 24 = 872 bits, consisting of a "1" bit followed by 871 "0" bits. Then a 128-bit length value is appended to the message, which contains the length of the original message (before the padding). The original length is 24 bits, or a hexadecimal value of 18. Putting this all together, the 1024-bit message block, in hexadecimal, is

6162638000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000

This block is assigned to the words W0, An W15 of the message schedule, which appears as follows.

W_0 = 6162638000000000
W_1 = 0000000000000000
W_2 = 0000000000000000
W_3 = 0000000000000000
W_4 = 0000000000000000
W_5 = 0000000000000000
W_6 = 0000000000000000
W_7 = 0000000000000000
W_8 = 0000000000000000
W_9 = 0000000000000000

Mehedi Hasan
19 BCS082

W_{10} = 0000000000000000 W_{13} = 0000000000000000

W_{11} = 0000000000000000 W_{14} = 0000000000000000

W_{12} = 0000000000000000 W_{15} = 0000000000000018

T5: Explain about the functioning of one iteration and compression function used in SHA-512.

Answer:

The SHA-512 algorithm is a secure hash function that is designed to generate a fixed-size, 512-bit message digest from an input message of arbitrary length. It achieves this by applying a series of iterations and compression functions to the input message.

One iteration of SHA-512 consists of several steps, including message padding, message parsing, and processing through the compression function. Here's a brief overview of each step:

1. Message Padding: The input message is first padded with a series of bits to ensure that its length is a multiple of 1024 bits. This is done to ensure that the input message can be broken up into a series of 1024-bit blocks for processing.

2. Message Parsing: The padded message is then parsed into a series of 1024-bit blocks, each of which is processed through the compression function.

(P.T.O)

Mehedi Hasan
19BCS082

3. Compression Function: The compression function takes as input a 1024-bit block of the message and combines it with the current state of the hash function to produce a new state. The compression function consists of several rounds of operations, including message expansion, message mixing, and message substitution, which are designed to produce a highly-secure and unpredictable output.

The compression function used in SHA-512 is based on a set of logical and arithmetic operations, including bitwise operations, modular addition, and conditional assignments. It consists of 80 rounds, each of constants and shifts to ensure that the output is highly unpredictable.

Overall, the combination of these iterations and the compression function ensures that SHA-512 produces a highly-secure and unpredictable message digest that is resistant to attacks.

(P.T.O)

Mehedi Hasan
19BCS082

T6: If the ASCII Charcacter "COMPUTERENGINEERING" is passed as a message to the SHA-512 as input, find the values in HEX assigned to the words W0, W1, W2, ... W15 for the defined message. (Use ASCII code for A: 01000001, B: 01000010, C: 01000011 ..... etc).

Answer:

C = 01000011
O = 01001111
M = 01001101
P = 01010000
U = 01010101
T = 01010100
E = 01000101
R = 01010010

E = 01000101
N = 01001110
G = 01000111
I = 01001001
N = 01001110
E = 01000101
E = 01000101
R = 01010010
I = 01001001
N = 01001110
G = 01000111

Length of message = 19 × 8 = 152 bits

Length of padding) = |P| = (-152 - 128) mod 1024
= 744

One → 1, 743 → 0's

Length field = 00000000 .... (112 '0' bits) 10011000

(P.T.O)

Mehedi Hasan
19 BCS082

⇒ The padded message will be :-

01000011	01001111	01001101	01010000
01010101	01010100	01000101	01010010
01000101	01001110	01000111	01001001
01001110	01000101	01000101	01010010
01001001	01001110	01000111	

1000.... (743 X)

000... (120 X) 10011000

The padded message is divided into sixteen 64-bit words, W0 through W15.

⇒ W_0 = 0X434F4D5055544552

W_1 = 0X454E4749 4E454552

W_2 = 0X494E474 000000000

W_3 = 0X0000000000000000

W_4 = 0X00000 00000000000

W_5 = 0X0000000000000000

⋮

W_{14} = 0X0000000000000000

W_{15} = 0X0000000000000098

Mehedi Hasan
19BCS082

T7: Using the ElGamal Digital signature scheme, let $p = 881$ and $d = 700$. Find values for e1 and e2, Choose $r = 17$. Find the value of signature S1 and S2 if the message $M = 400$.

Answer:

We have $p = 881$ $d = 700$. We choose $e_1 = 3$. Then $e_2 = e_1^d \bmod p = 471$.

$S_1 = e_1^r \bmod p = 3^{17} \bmod 881 = 540$

$S_2 = (M - d \times S_1) r^{-1} \bmod (p-1) = (400 - 700 \times 540) 17^{-1} \bmod 880 = 720$

We can verify the signature because V_1 is congruent to V_2.

$V_1 = e_1^M \bmod p = 3^{400} \bmod 881 = 186$

$V_2 = e_2^{S_1} \times S_1^{S_2} \bmod p = 471^{540} \times 540^{720} \bmod 881 = 186$.

T8: Alice choose $q = 101$ and $p = 8081$. Alice selects $e_0 = 3$ and calculates $e1 = e_0^{(p-1)/q} \bmod p = 6968$. Alice chooses $d = 61$ as the private key and calculates $e_2 = e_1^d \bmod p = 2038$. Now Alice can send a message to Bob. Assume that hash of the message $h(M) = 5000$ and Alice chooses $r = 61$:

i. Generate the signature of the message.

ii. Verify the signature generated by Alice.

Mehedi Hasan
19BCS082

Answer:

Alice chooses $q = 101$ and $p = 8081$. Alice selects $e_0 = 3$ and calculates $e_1 = e_0^{(p-1)/q} \bmod p = 6968$. Alice chooses $d = 61$ as the private key and calculates $e_2 = e_1^d \bmod p = 2038$. Now Alice can send a message to Bob. Assume that $h(M) = 5000$ and Alice choose $r = 61$:

$h(M) = 5000 \quad r = 61$

$S_1 = (e_1^r \bmod p) \bmod q = 54$

$S_2 = ((h(M) + dS_1)^{r-1}) \bmod q = 40$

Alice sends M, S_1, and S_2 to Bob. Bob uses the public keys to calculate V.

$S_2^{-1} = 48 \bmod 101$

$V = [(6968^{5000 \times 48} \times 2038^{54 \times 48}) \bmod 8081] \bmod 101 = 54$

Because S_1 and V are congruent, Bob accepts the message.

TQ: Using the RSA scheme, let $p = 809$, $q = 751$, and $d = 23$. Calculate the public key e. Then

i. Sign and verify a message with $M1 = 100$. call the signature S_1.

ii. Sign and verify a message with $M2 = 50$. Call the signature $S2$.

iii. Show that if $M = M1 \times M2 = 5000$, then $S = S1 \times S2$.

(P.T.O)

Mehedi Hasan
19BCS082

Answer:

We have $n = 809 \times 751 = 607559$ $\phi(n) = (809-1) \times (751-1) = 606000$. Since $d = 23$, we have $e = d^{-1} \bmod \phi(n) = 158087$.

i. We have

$S_1 = M_1^d \bmod n = 100^{23} \bmod 607559 = 223388$

$M_1 = S_1^e \bmod n = 223388^{158087} \bmod 607559 = 100$

ii. We have

$S_2 = M_2^d \bmod n = 50^{23} \bmod 607559 = 5627$

$M_2 = S_2^e \bmod n = 5627^{158087} \bmod 607559 = 50$

iii. If $M = M_1 \times M_2 = 5000$, we have

$S = M^d \bmod n = 5000^{23} \bmod 607559 = 572264$

$S = (S_1 \times S_2) \bmod n = (223388 \times 5627) \bmod 607559 = 572264$

T10: Using the RSA digital signature scheme, let $p = 809$, $q = 751$ and $d = 23$. Calculate the set of public key and Pacivate key and then sign and verify a message with $M = 50$.

Mehedi Hasan
19BCS082

Answer:

$P = 809$

$q = 751$

$M = P \times q = 6,07,559$

$\Phi(M) = (p-1) \times (q-1)$

$= (808) \times (750) = 6,06,000$

$d = 23$

$e = d^{-1} \bmod \Phi(M)$

$= 23^{-1} \bmod (6,06,000) = 158087$

⇒ Private key = $d = 23$

Public key = $(e, M) = (158087, 607559)$

⇒ Signing :-

$M = 50$

$S = M^d \bmod n$

$= (50)^{23} \bmod (607559) = 5627$

⇒ Verifying :-

$M' = S^e \bmod n$

$= (5627)^{158087} \bmod 607559$

$M' = 50$

$M' = M$

Hence, message is accepted. Because signature is verified.

Mehedi Hasan
19BCS082

T11: What is digital signature? Explain the mechanism of generation of digital signature using Schnorr digital signature technique.

Answer:

Digital Signature:

A digital signature is a means by which the sender can electronically sign the data and the receiver can electronically verify the signature. The sender uses a process that involves showing that she owns a private key related to the public key that she has announced publicly. The receiver uses the sender's public key to prove that the message is indeed signed by the sender who claims to have sent the message.

The mechanism of generation of digital signature using Schnorr digital signature technique is explained below:

S_1, S_2 : signatures
M: Message
(e_1, e_2, p, q) : Alice's public key
(d): Alice's Private key
r: Random secret

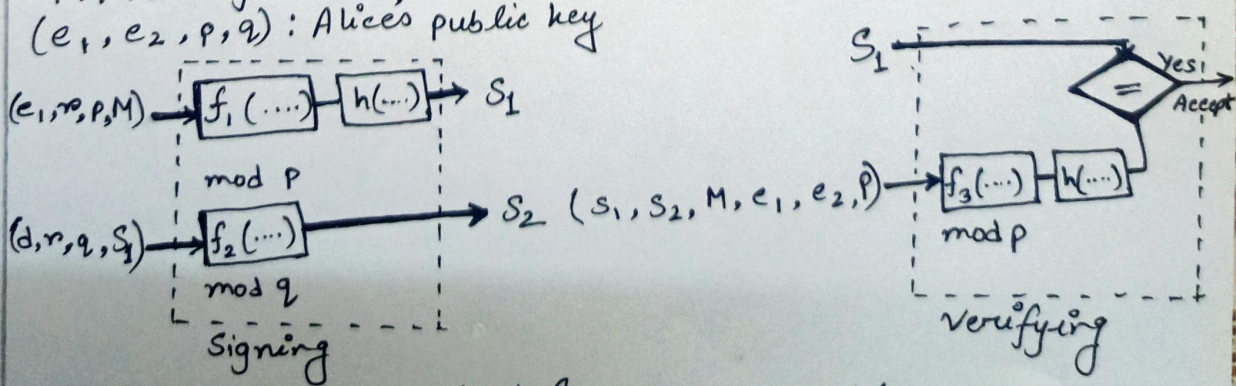

Figure: General idea behind the Schnorr digital signature scheme.

In the signing process, two functions create two signatures, in the verifying process, the output of one function is compared to the first signature for verification. Above figure also shows the inputs to each function. The important point is that the scheme uses two moduli: p and q. Function 1 and 3 use p; function 2 uses q. The details of inputs and the functions will be discussed shortly.

Key Generation:

Before signing a message, Alice needs to generate keys and announce the public ones to the public.

1. Alice selects a prime p, which is usually 1024 bits in length.
2. Alice selects another prime q, which is the same size as the digest created by the cryptographic hash function (currently 160 bits, but it many change in the future). The prime q needs to divide (p-1). In other words, $(p-1) = 0 \mod q$.
3. Alice chooses e_1 to be the q^{th} root of 1 modulo p. To do so, Alice chooses a primitive element in Z_p, e_0 and calculates $e_1 = e_0^{(p-1)/q} \mod p$.
4. Alice chooses an integer, d, as her private key.
5. Alice calculates $e_2 = e_1^{d} \mod p$.
6. Alice's public key is (e_1, e_2, p, q); her private key is (d).

In the Schnorr digital signature scheme, Alice's public key is (e_1, e_2, p, q); her private key (d).

Mehedi Hasan
19BCS082

Signing and verifying:

M: Message
S_1, S_2: Signatures
V: verification
r: Random secret
(d): Alice private key
(e_1, e_2, p, q): Alice's public key
|: Concatenation
h(...): Hash algorithm

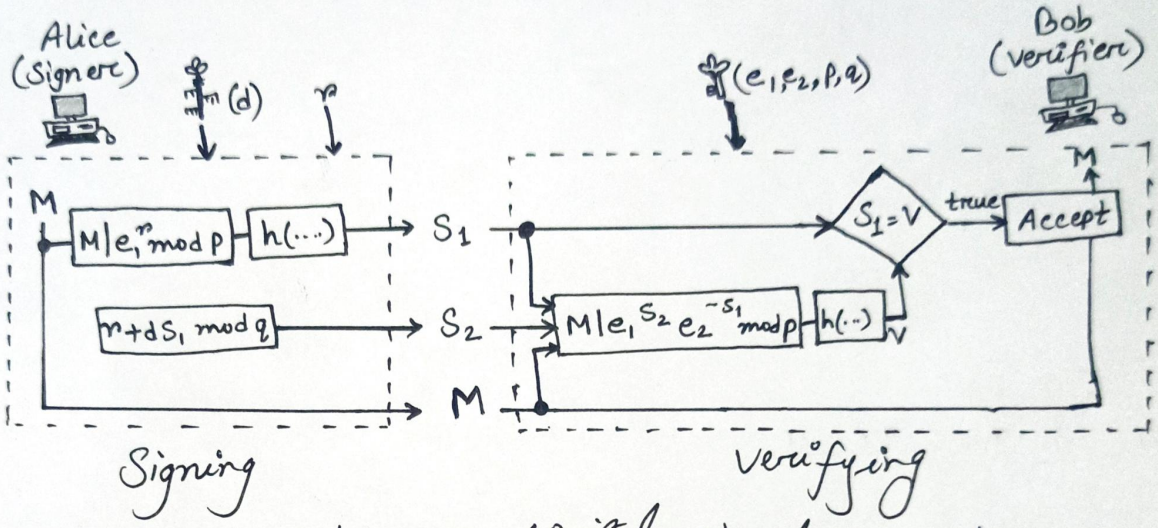

Figure: Schnorr digital signature scheme.

Signing:

1. Choose a random integer r. r needs to be changed each time a new message is being sent.
2. Calculate the first signature $S_1 = h(M|e_1^r \bmod p)$.
3. Calculate the second signature $S_2 = r + d \times S_1 \bmod q$.
4. Send M, S_1 and S_2.

Verifying Message:

The receiver, receives M, S_1 and S_2.

1. Calculate $V = h(M|e_1^{S_2} e_2^{-S1} \bmod p)$
2. If S_1 is congruent to V modulo p, message is accepted, otherwise, rejected.

Mehedi Hasan
19BCS082

T12: What is Blind Digital Signature? Explain the blind digital signature generation and verification process using RSA.

Answer:

Blind Signature:

Sometimes we have a document that we want to get signed without revealing the contents of the document to the signer. For example, a scientist, say Bob, might have discovered a very important theory that needs to be signed by a notary public, say Alice, without allowing Alice to know the contents of the theory. David Chaum has developed some patented blind digital signature schemes for this purpose. The main idea is as follows:

a. Bob creates a message and blinds it. Bob sends the blinded message to Alice.

b. Alice signs the blinded message and returns the signature on the blinded message.

c. Bob unblinds the signature to obtain a signature on the original message.

(P.T.O)

Mehedi Hasan
19BCS082

Blind Signature Based on the RSA Scheme

Let us briefly describe a blind digital signature scheme developed by David chaum. Blinding can be done using a variation of the RSA scheme. Bob selects a random number, b, and calculates the blinded message $B = M \times b^e \bmod n$, in which e is Alice's public key and n is the modulus defined in the RSA digital signature scheme. Note that b is sometimes called the blinding factor. Bob sends B to Alice.

Alice signs the blinded message using the signing algorithm defined in the RSA digital signature $S_b = B^d \bmod n$, in which d is Alice's private key). Note that S_b is the signature on the blind version of the message.

Bob simply uses the multiplicative inverse of his random number b to remove the blind from the signature. The signature is $S = S_b\, b^{-1} \bmod n$. We can prove that S is the signature on the original message as defined in the RSA digital signature scheme:

$$S = S_b\, b^{-1} = B^d\, b^{-1} = (M \times b^e)^d\, b^{-1} = M^d\, b^{ed}\, b^{-1} = M^d\, b\, b^{-1} = M^d$$

S is the signature if Bob has sent the original message to be signed by Alice.

———— o ————