**Final Exam CMSE-512 26.06.2023, 10.30 (150 min, 100 points)**

St. Name, Surname\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ St.Id#\_\_\_\_\_\_\_\_\_\_\_\_\_

**SixA4-sized sheets of paper with your handwritten notes may be used. Calculators are allowed. Other electronic devices are not allowed**

Instructor Alexander Chefranov

**Totally 11 questions, 8 pages**

Good Luck!

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question | T1 | T2 | T3 | ~~T4~~ | T5 | T6 | T7 | T8 | T9 | T10 | T11 | Total |
| Point | 5 | 5 | 5 | ~~6~~ | 6 | 6 | 13 | 13 | 13 | 14 | 14 | ~~100~~ 94 |

**Before MT Exam tasks 1-6 (33 points)**

**Task 1 (5 points)** What is the difference between the security threat and the security attack?

The security threat is a potential to violate security, and the security attack is an action aiming to violate security.

**Task 2. (5 points)** What are the three types of the malicious software?

Logical bombs, viruses, warms

**Task 3. (5 points)** For RSA with N=119, define a pair of the private and public keys, and check their correctness. Show your calculations, give necessary explanations.

**Hints**: Two large prime numbers, *p* and *q*, , are selected, and an integer, *d*, is chosen that is relatively prime to *(p-1)(q-1)*. Finally, an integer e is computed such that

, N=pq, C=MemodN, M=CdmodN

EXTENDED EUCLID(m,b)

1. (A1,A2,A3):=(1,0,m); (B1,B2,B3):=(0,1,b);
2. if B3=0 return A3=gcd(m,b); no inverse
3. if B3=1 return B3 = gcd(m,b); B2= b-1 mod m
4. Q=
5. (T1,T2,T3):=(A1-QB1, A2-QB2, A3-QB3)
6. (A1,A2,A3):= (B1,B2,B3)
7. (B1,B2,B3):= (T1,T2,T3)
8. goto 2

N=119=p\*q=17\*7, p=17, q=7, fi(N)=(p-1)(q-1)=16\*6=96. Let e=5, then d=e^(-1) mod 96. Find it by EEA

A=(1,0,96), B=(0,1,5)

Q=floor(A3/B3)=floor(96/5)=19

T=A-q\*B=(1-19\*0, 0-1\*19, 96-19\*5)=(1,-19, 1)

A=B=(0,1,5)

B=T=(1,-19, 1)

B3=1=>B2=-19 mod 96 = 79 is e^(-1) mod 96. Check it: 5\*79=395 mod 96 = 4\*96+1 mod 96 =1, hence, it is correct.

**~~Task 4. (6 points)~~** ~~What S-boxes and what their outputs in the 1~~~~st~~ ~~round of DES define the 1~~~~st~~ ~~, 2~~~~nd~~~~, and 3~~~~rd~~ ~~bit inputs of the S-box S2 in the 2~~~~nd~~ ~~round? Fill in the Answer table below. Explain your answer~~

**Hints**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |

|  |
| --- |
| Permutation function( P ) |
| 16 7 20 21 29 12 28 171 15 23 26 5 18 31 102 8 24 14 32 27 3 919 13 30 6 22 11 4 25 |

 |

Answer table:

|  |  |  |
| --- | --- | --- |
| Bit of R1 | S-box number | S-box output number |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

**Task 5. (6 points)** Explain why 64-bit registers *a,b,c,d,e,f,g,h* are initialized in SHA-512. Calculate the first three hexadecimal digits (3C6) of the register *c*.

**Hints**:





The 3rd prime number, 5, is used to initialize c:

Sqrt(5)= 2.2360679774997896964091736687313, its fractional part is 0.2360679774997896964091736687313, multiply it by 16 and take the integer part:

3.7770876399966351425467786997004, which is 3 as expected; take the fractional, multiply by 16 and take the integer part 12.433402239946162280748459195207, which 12=c, that is expected; and lastly, take the fractional part 0.433402239946162280748459195207, multiply by 16 and take the fractional part: 6.9344358391385964919753471233075, that is 6 as also expected.

**Task 6. (6 points)** How many certificates are used in the SSL protocol, whom they belong to, who and how validates them, and what for each of them is used?

**Hints:**

“Assume that a browser, C, connects to a server, S, that claims to represent a particular enterprise, E (for example, Macy’s). In this case, the protocol consists of the following steps:

1. S sends C a copy of its certificate signed by the CA – in the clear
2. C validates the certificate’s signature using the CA’s public key (included in its browser) and hence knows that the public key in the certificate belongs to the enterprise named in the certificate.
3. C generates and sends to S a session key encrypted with the public key in the certificate.”

One certificate is used, it belongs to the server, client validates it by checking CA’s signature, validity period, and revocation lists. The certificate is used to certify the public key of the server, and the client next uses it to encrypt the session key.

**After MT Exam tasks 7-11 (67 points)**

**Task 7. (13 points)** **What for** and **how** the client, C, creates the dual signature in SET protocol? How is it validated by the merchant, M? Give detailed explanations.

**Hints:**

“Before SET begins, C and M negotiate the terms of a purchase. The protocol begins with a handshake in which C and M exchange certificates and authenticate each other. C sends its certificate to M, and M sends both its certificate and G’s certificate to C, at which point C and M know each other’s and G’s public key. Then the purchase transaction begins.

1. M sends a signed message to C containing a (unique) transaction Id (which is used to guard against replay attacks). C uses the public key in M’s certificate to check the signature and hence knows that the message came from M and was not altered in transit.
2. C sends a message to M containing two parts plus the dual signature:
3. The transaction Id, C’s credit card information, and the dollar amount of the order (but not a description of the items purchased) – encrypted with G’s public key:



1. The transaction Id, the dollar amount of the order, a description of the items purchased (but not C’s credit card information) – encrypted with M’s public key:



The dual signature has three fields:

1. The message digest, MD1, of the first part of the message:



where f is the message digest function

1. The message digest, MD2, of the second part of the message:



1. C’s signature of the concatenation of MD1 and MD2:



Thus, the complete dual signature is



and the complete message sent from C to M is .

The dual signature binds the two parts of the message. So, for example, an attempt by an intruder or M to associate  with  does not work since its message digest, MD2’, will differ from MD2. Although MD2’ can be substituted for MD2 in the dual signature,  cannot be used as the signature for MD1\*MD2’, and only C can compute the correct dual signature for the reconstructed message.

1. M decrypts the second part of the message with its private key (but it cannot decrypt the first part, which contains the credit card number). The merchant then
2. Uses the dual signature to verify that has not been altered in transit. It first computes the message digest of  and checks that it is the same as the second field of the digital signature (MD2). It then uses the public key in C’s certificate to check that the third field is the correct signature for the concatenation of the first two fields.
3. Verifies the transaction Id, the dollar amount of the order, and the description of the items purchased

Next M sends a message to G containing two parts:

(a)  and the dual signature it received from C:



1. The transaction Id and the dollar amount of the order – signed with M’s private and encrypted with G’s public key:



The complete message sent from M to G is , together with copies of C’s and M’s certificates

1. G decrypts the message using its private key.
2. It uses the dual\_signature and the public key in C’s certificate to verify that  was prepared by C and was not altered (as in step 3a).
3. It uses the message digest of the credit card information in C’s certificate to verify the credit card number supplied in .
4. It uses M’s signature in and the public key in M’s certificate to verify that  was not altered
5. It checks that the transaction Id and the dollar amount are the same in  and  (to verify that M and C agreed on the purchase)
6. It checks that the transaction Id was never submitted before (to prevent a replay attack)
7. It does whatever is necessary to approve the credit card request

Then G returns a signed approved message to M. At this point, the transaction is committed.

1. When M receives the approved message, it knows that the transaction has committed. It sends a signed message to C: transaction complete. C then knows that transaction has committed.”

The client C creates the dual signature for binding the two message parts, m1 and m2 sent to the merchant M by C. It is created by hashing each part, concatenating them and signing (calculating hash of the concatenation and encrypting it by C’s private key). The merchant validates the dual signature by recalculating hashes of the message parts, concatenating them, hashing the result and comparing it versus the result of the decryption of the dual signature.

**Task 8. (13 points)** For the digital cash 16-bit serial number, SN, write a mathematical formula or C-like pseudo-code defining its validity as follows: the square of the number in its 8 senior bits shall be k times greater than the number in its junior 8 bits. For example, let the serial number SN=0x1234=0001 0010 0011 0100, and k=3 Then the number in the senior 8 bits is SB=0001 0010 = 0x12 = 16+2=18, and its square is 324. The number in the 8 junior bits is JB=0011 0100 = 0x34 = 3\*16+4 = 52, and its three times value is 156. Since 324 is greater than 156, SN is valid.

SB=floor(SN/256); JB=mod(SN,256); if SB\*SB>k\*JB then return valid else return invalid;

**Task 9. (13 points)** What tags in the XML code below from the Lecture notes are used to define the cipher, the encryption key, and the result of encryption? Why encryption is used here? Give necessary explanations to your answers.

Figure 26.7. An encrypted element within an XML document

<PaymentInfo xmlns = “<http://...>”>

 <Name> John Doe </Name>

 <EncryptedData Type =

“

[http://www.w3.org/2001/04/xmlenc#Element”](http://www.w3.org/2001/04/xmlenc#Element\”  )

xmlns=”[http://www.w3.org/2001/04/xmlenc#](http://www.w3.org/2001/04/xmlenc)”/>

<EncryptionMethod Algorithm =

“[http://www.w3.org/2001/04/xmlenc#tripledes-cbc”/](http://www.w3.org/2001/04/xmlenc#tripledes-cbc)>

 <ds:KeyInfo xmlns:ds =

“[http://www.w3.org/2000/09/xmldsig#](http://www.w3.org/2000/09/xmldsig)“>

 <ds:KeyName>keyABC</ds:KeyName>

 </ds:KeyInfo>

 <CipherData>

 <CipherValue>Zx23XAbc4..</CipherValue>

 </CipherData>

 </EncryptedData>

</PaymentInfo>

EncryptionMethod defines the cipher, the encryption key is defined by ds:KeyInfo, and the result of encryption is defined by CipherValue. The encryption is used to protect the sensitive payment data.

**Task 10. (14 points)** In the figure below, explain what is the security reason for t1, q1, t2, and q2 using? If they are not used, what attack might be conducted? How the attack can be mitigated?



The timestamps t1, t2, and the nonces q1, q2 are used to counter replay attacks. If they are not used, the hash of the user name and password (authenticator) can be intercepted and used by a hacker to impersonate a valid user (replay attack). The replay attack can be mitigated by the use of the dynamic data such as timestamps and hashes, so that the authenticator each time changes, but these dynamic data shall not be used twice. For that purpose, the server shall maintain the database of the already used timestamps and hashes, and each received one checking on existence in the database.

**Task 11. (14 points)** Consider the multilevel security database sample from the Lecture notes below

(a) Employee – the original tuples

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Salary | JobPerformance | TC |
| Smith U | 40000 C | Fair S | S |
| Brown C | 80000 S | Good C | S |

Explain why the query

UPDATE EMPLOYEE

SET JobPerformance=’Excellent’

WHERE Name=’Smith’

invoked by a user of the level C results in

(d) Employee – poly-instantiation of the Smith tuple

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Salary | JobPerformance | TC |
| Smith U | 40000 C | Fair S | S |
| Smith U | 40000 C | Excellent C | C |
| Brown C | 80000 S | Good C | S |

What is poly-instantiation? Why is it used?

The result of the query is the “Smith” tuples, the old and the new one because the old tupleis higher secret than the user’s level. It is called polyinstantion (several instances of the same tuple. It is used to hide from low-level uses the fact of existence of the more secret tuple instance.