**CSC418 – Cryptography and Security**

**School of Computing, USM**

# Final Exam 2017

**Due on Thursday, May 11th at 2:00am.**

**Name: Yuner Han**

**Student ID: w974294**

**Read the following instructions before you start:**

* This test is open-book, open-notes, open Internet.
* You are allowed to write computer program for calculation. But it is not necessary.
* This exam is an individual test. Any type of collusion, for example soliciting answers on the Internet, is not allowed.
* Type your answer in the DOC file and submit it in Blackboard by the due time.
* To avoid the last minute network problem, better to submit at least one hour earlier.
* I reserve the copy right of this exam. Please do not disseminate the problem set.

Have fun!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **I** | **II** | **III** | **IV** | **Total score** |
|  |  |  |  |  |

### Max. Score: 100

1. **Brief Answers. (5 points x 5) (Be concise. Show your knowledge and viewpoints.)**
2. Describe the discrete logarithm problem. Then give an example with certain numbers.

Find k given a and ak. e.g. a = 2, ak = 1024, then k = log2(1024) = 10

1. What is a hash function? What properties are required for a hash function to become a cryptographic hash function? List as many crypto hash functions as you know.

Hash function is a function that takes a variable-length block of input and gives a fixed-length hash value as output. To become a cryptographic hash function, the function must have the one-way property (given a hash value, it’s computationally infeasible to find the input that maps to it), and the collision-free property (it’s computationally infeasible to find two inputs mapping to the same hash value).

XOR with 1-bit rotation to the right. XOR of every 16-bit block. Rabin hash function. SHA. MD5.

1. What is meet-in-the-middle attack? What is man-in-the-middle attack?

Meet-in-the-middle attack can be used when there are multiple encryptions in the algorithm. It encrypts the cipher and decrypts the plaintext with all possible keys, and compares the results to find the keys.

Man-in-the-middle attack is where two parties try to communicate, and instead of the two parties establishing a connection and sharing a secret key, an attacker pretend to be the other side of communication and share secret keys with the two parties respectively. Then all the communication between those two parties can be known and controlled by the attacker.

1. What does MAC stand for? Describe at least two different approaches to producing MAC.

Message Authentication Code.

HMAC: HMAC(*K*, *M*) = H[(*K+*Ꚛopad) ‖ H[(*K+*Ꚛipad) ‖ *M*]] (K+: padded K with 0s to left to the length b of a block, opad: 01011100 repeated b/8 times, ipad: 00110110 repeated b/8 times, H: the hash function)

DAA: Using the cipher block chaining mode of operation of DES with an initialization vector of 0.

1. What properties should a digital signature scheme have? What attacks and forgeries does it face to?

It must verify the author, the date and the time of the signature. It must authenticate the content at the time of the signature. It must be verifiable to third parties to resolve disputes.

Key only attack, known message attack, generic chosen message attack, directed chosen message attack, adaptive chosen message attack.

Total break, universal forgery, selective forgery, existential forgery.

1. **Long Answers. (10 points x 4) (Be concise and thorough. Answers are in textbook but don’t copy and paste. Digest and summarize. Do not exceed the space limit.)**
2. Explain how NIST DSA (digital signature algorithm) works.

The private key x is randomly or pseudo randomly chosen between 1 and q-1, where q is a N-bit prime number. The public key is calculated as y = gx mod p, where p is a prime number with length between 512 and 1024 bits such that q divides p-1, and g = h(p-1)/q mod p where h is an integer between 1 and p-1 with the restriction that g>1. H is the hash function. M is the message. And k is an integer generated randomly or pseudo randomly in each signing.

The first part of the signature r = (gk mod p) mod q. And the second part s = (k-1(H(M)+xr)) mod q. The received verification message is (M’, r’, s’). To verify the signature, the receiver calculate w = (s’)-1 mod q, u1 = (H(M’)w) mod q, u2 = (r’)w mod q, v = ((gu1yu2) mod p) mod q. If v matches r’ then the signature is validated.

1. Explain how Kerberos works.

At the beginning of a logon session, the client sends request access to the ticket-granting server (TGS) with the user ID, the TGS ID and a time stamp. The authentication server (AS) receives the request and responds with a message encrypted with user’s password Kc. The message contains the session key shared between the client and TGS, the ticket-granting ticket, the TGS ID, a time stamp indicating the time at which the ticket was created and the life time of the ticket. The client then prompts the user with user’s password and tries to decrypt the message.

Now that the client has the ticket-granting ticket, during the logon session, each time the user requests access to a new service, the client uses the ticket-granting ticket to authenticate itself. The client sends to TGS the ID of the desired service, the ticket-granting ticket and an authenticator. The ticket-granting ticket is encrypted with a key shared between AS and TGS and contains the session key, the ID and address of the user, the ID of TGS, and the time stamp and the life time of the ticket. The TGS can decrypt the ticket and obtain the session key. The authenticator is encrypted with the session key and contains the ID and address of the user and a time stamp. The TGS can then decrypt the authenticator and compare the user information in it with the user information in the ticket-granting ticket. If all match, the user is authenticated and the TGS replies with a message encrypted with the session key shared between the client and the TGS. The message contains a session key shared by the client and the server V, the service-granting ticket, the ID of V and a time stamp.

Now that the client has the service granting ticket, each time user requests the service, the client sends the service-granting ticket and an authenticator to V. And V verifies the user’s identity using the same method. For mutual communication, V replies with the time stamp in the authenticator incremented by q, encrypted with the session key.

1. Explain how X.509 certificate works.

A user first obtains its certificate containing its ID information, its public key, and the information about the trusted certification authority (CA). The user then generates the hash code from the unsigned certificate and encrypts the hash code with CA’s private key to get the signed certificate. The user sends both the unsigned and the signed certificates. To verify the public key, recipient should decrypt the signed certificate with CA’s public key and get the hash code from the unsigned certificate. The recipient can then compare the hash codes to verify the signature.

The certificate is created by CA and contains the version of the certificate, serial number, signature algorithm identifier, issuer name, period of validity, subject name, subject public key, issuer unique identifier, subject unique identifier, extensions and signature.

The CA should obtain a certificate revocation list containing the signature algorithm identifier, issuer name, this update time, next update time, list of user certificate serial numbers and revocation dates, and signature.

Information about key and policy information, subject and issuer attributes or certification path constraints may be needed and included in extensions.

1. Explain how TLS (transport layer security) works.

The TLS Record Protocol provides confidentiality and integrity for the connection. The protocol takes an application message to be transmitted, fragments the message, optionally compresses the data, computes the MAC of the data with HMAN algorithm, appends the MAC to the data, encrypts the data and the MAC, and adds header including content type, version numbers, and compressed length.

The Change Cipher Spec Protocol contains a simple byte of value 1 to cause the pending state to be copied into the current state which updates the cipher suite.

The Alert Protocol contains the alert level and alert code. If the alert level is fatal, the connection is immediately terminated.

Before the application message is transmitted, the Handshake Protocol is used for the server and client to authenticate each other and to determine the encryption and MAC algorithm and cryptographic keys to be used. First the server and client establish security capabilities by sending the protocol version number, an initial random number, the session ID, the cipher suite, and the compression method. The cipher suite contains the key exchange method and the cipher specifications including the cipher algorithm, the MAC algorithm, the cipher type, parameter indicating whether the cipher is exportable, the hash size, the key material and IV size. Next the server sends certificate message, sends the key to the clients and requests certificates from the clients. Then the client sends the certificate message and key to the server. Finally the server and client both update the current CipherSpec and send the finish message, and the handshake is completed.

To generate a shared master secret by means of key exchange, first a pre-master secret is exchanged and both the server and client calculate the master secret. The MAC secret, key and IV required in CipherSpec are generated from the master secret by hashing it into a sequence of secure bytes of sufficient length.

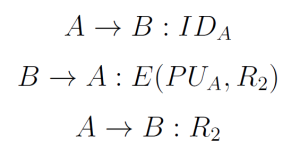
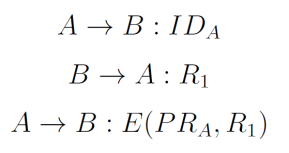
1. **Crypto Computation (15 points)**

(Refer to Chapter 10.) Alice and Bob use Diffie-Hellman key exchange scheme with a common prime q=23 and a primitive root α= 5.

1. If Alice chooses her private key XA = 6, what is Alice's public key YA?
2. If Bob chooses his private key XB = 15, what is Bob's public key YB?
3. What is their shared secret key?
4. YA = 56 mod 23 = 23 mod 23 = 8
5. YB = 515 mod 23 = 27\*5 mod 23 = 13\*5 mod 23 = 19
6. K = 196 mod 23 = 163 mod 23 = 2
7. **Protocol Design and Analysis (20 points)**

(Refer to Chapter 15.) Consider the following two one-way authentication protocols based on asymmetric encryption.

Protocol One: Protocol Tow:

1. Describe the two protocols in your words.
2. What type(s) of attack are the two protocols susceptible to, respectively? Explain.
3. Protocol one: A requests the service by sending its ID to B. B encrypts a message R2 with A’s public key and send the encrypted message to A. A decrypts the message with its private key to get R2 and sends it back to B. Since only A knows its private key and can decrypt the message correctly, if B receives R2 then A is authenticated.

Protocol two: A requests the service by sending its ID to B. B sends a message R1 back to A. A encrypted the message with its private key and send it to B. B tries to decrypt the signed message with A’s public key. If the decrypted message is R1, A is authenticated.

1. Protocol one: Replay attack: an opponent can copy R2 and send it to B later.

Protocol two: Replay attack: an opponent can copy the encrypted message A sends to B and replay it later.