

Name ID Section Seat No

Sirindhorn International Institute of Technology Thammasat University

Final Exam Answers: Semester 2, 2011

Course Title: CSS322 Security and Cryptography

Instructor: Steven Gordon

Date/Time: Wednesday 4 April 2012; 9:00–12:00

Instructions:

- This examination paper has 21 pages (including this page).
- Conditions of Examination: Closed book; No dictionary; Non-programmable calculator is allowed
- Students are not allowed to be out of the exam room during examination. Going to the restroom may result in score deduction.
- Students are not allowed to have communication devices (e.g. mobile phone) in their possession.
- Write your name, student ID, section, and seat number clearly on the front page of the exam, and on any separate sheets (if they exist).

Security and Cryptography, Semester 2, 2011

Prepared by Steven Gordon on 4 April 2012

CSS322Y11S2E02, Steve/Courses/2011/S2/CSS322/Assessment/Final-Exam.tex, r2280

Question 1 [7 marks]

The original standard for encryption in a wireless LAN (WiFi) is called Wired Equivalent Privacy (WEP). Early devices that used WEP allowed the user to select a 10 hexadecimal digit value, which was combined with a 24-bit initialisation vector to produce the encryption key. The IV was sent as plaintext and changed for every packet sent.

- (a) What is the entropy of the user selected value? [1 mark]

Answer. *10 hexadecimal digits is 40 bits. Therefore the entropy is 40*

- (b) An alternative to entering a hexadecimal value would be to allow the user to enter the key using the 94 ASCII printable characters. How many characters are needed for the ASCII-based key? [2 marks]

Answer. *With 94 possible characters, there are 6.55 bits per character. So would need at least 7 characters to create 40 bits.*

When a user can select an ASCII string (from the 94 printable characters) they normally do not chose it randomly. One study has calculated the approximate entropy of ASCII strings if the user can choose: any value; any value, except for those in a dictionary. The entropy values for different length strings is shown in Table 1.

Table 1: Entropy of user chosen ASCII strings

| <i>Length</i> | <i>Any Value</i> | <i>Any Value, except Dict.</i> |
|---------------|------------------|--------------------------------|
| 6 | 14 | 23 |
| 10 | 21 | 32 |
| 14 | 27 | 36 |
| 18 | 33 | 40 |
| 20 | 36 | 42 |
| 22 | 38 | 44 |
| 24 | 40 | 46 |
| 30 | 46 | 52 |
| 40 | 56 | 62 |

An improved security protocol for wireless LAN is called WiFi Protected Access (WPA). It allows a 256-bit key, generated from a password chosen by the user of between 8 to 63 printable ASCII characters. Assume a malicious user can attempt to guess the password at a rate of 1,000 guesses per second.

- (c) If the user chose a 10 character password and was allowed any value, on average approximately how long would it take the malicious user to guess the password? [2 marks]

Answer. From the table, a 10 character password allowing any value gives a password with an entropy of 21. That gives 2^{21} possible passwords. On average half of the passwords need to be guessed giving $2^{20} = 1048576$ guesses. At 1000 per second it takes the attacker 1048.576 seconds or approximately 1000 seconds.

- (d) If the user is allowed to choose a password with any value, except that from a dictionary, then what is the minimum password length that offers the same strength as the 10 hexadecimal digit value in part (a)? [2 marks]

Answer. From part (a) we need a password with entropy of 40. From the table the password must be 18 characters to give an entropy of 40.

Question 2 [8 marks]

The following shows the partial output of the `/etc/shadow` file on a Linux operating system. This file store the usernames and password related information for users of the computer. The values are separated by a `:` character. (Note that the data for each user is normally on a single line; I have wrapped it across two lines to fit within the page for this question).

```
boonsita:$5$8MlKVqhP$sdF897ds12poheds9032.asjfeiojfsdf9REWk32ds/
```

```
chavalit:$6$8jWr21do$0rx85gh9Tz3C5k9sTwoKOVWmFwteaLmR.TkzIdNFCdc1NfqNI
362apshyCIKFE8yBxkBhwMJ1ABCLGQ7N4t6H/
```

```
tossapong:$1$KWas931B$89jASDI3fs.kjlds9FP01/
```

A portion of the `crypt` man page is shown below. It describes the format of the second field from the `/etc/shadow` file above.

If `salt` is a character string starting with the characters `"id"` followed by a string terminated by `"$"`:

```
$id$salt$encrypted
```

then instead of using the DES machine, `id` identifies the encryption method used and this then determines how the rest of the password string is interpreted. The following values of `id` are supported:

| ID | Method |
|----|---|
| 1 | MD5 |
| 2a | Blowfish (not in mainline glibc; added in some Linux distributions) |
| 5 | SHA-256 (since glibc 2.7) |
| 6 | SHA-512 (since glibc 2.7) |

So `5salt$encrypted` is an SHA-256 encoded password and `6salt$encrypted` is an SHA-512 encoded one.

"salt" stands for the up to 16 characters following `"id"` in the salt. The encrypted part of the password string is the actual computed password. The size of this string is fixed:

| | |
|---------|---------------|
| MD5 | 22 characters |
| SHA-256 | 43 characters |
| SHA-512 | 86 characters |

The characters in "salt" and "encrypted" are drawn from the set `[azAZ09./]`.

Answer the questions based on the information above. Assume the users chose their passwords randomly. Assume the algorithms used have no flaws.

- (a) Can you tell which user has the longest password? Explain your answer. [2 marks]

Answer. *No. The file shows the hash of the passwords (combined with a salt). The hash is a particular length, but that doesn't indicate the length of the password. Using the same hash algorithm, the hash of a 5 character password will produce the same length hash value as a password with 20 characters.*

- (b) Assuming Tossapong chose a password p , then write an equation showing how the “encrypted” value, e , is calculated. Use the names of any algorithms and specific values from the file in the equation. [2 marks]

Answer. $e = MD5(p||s)$ where $s = KWAs93lB$.

Concentrating on user Tossapong, assume the Linux system has forced him to choose a 6-character password. A malicious user has paid for database of 10^9 6-character passwords and their corresponding MD5 hash values. The database has no compression or efficient data structures.

- (c) How much data is stored in the database? [2 marks]

Answer. *The MD5 hash value is 128 bits or 16 Bytes. Each password is 6 Bytes. With 10^9 entries then there is 22GB of data. If you didn't know the length of the MD5 hash you could determine from the man page that 22 characters, each from a set of 64, giving 132 bits. Hence an answer of 22.5GB would also be accepted.*

- (d) Can the malicious user use the database to try to find Tossapong's password? If yes, explain how. If no, explain why not. [2 marks]

Answer. *No. The database does not contain the salt value of Tossapong; the hash in the file is of the password and the salt value.*

Question 3 [8 marks]

- (a) Explain the difference between a worm and virus. [2 marks]

Answer. *A virus attaches to another program, while a worm is a standalone program.*

- (b) Explain the difference between a normal (parasitic) virus, a metamorphic virus and a polymorphic virus. [3 marks]

Answer. *A parasitic virus simply copies itself, as is, to other files. When a polymorphic virus copies the original virus to create a new virus, the new virus appears different than the original, but functions the same. For a metamorphic virus, the new virus both appears different and functions differently.*

- (c) Give an example of what a virus could do to be polymorphic. [1 mark]

Answer. *Introduce instructions that do nothing, e.g. NOP in assembly. Rearrange lines of code, assuming the lines are independent of each other.*

- (d) Which of the three types of virus (parasitic, metamorphic, polymorphic) is hardest to detect by anti-virus software? Explain why. [2 marks]

Answer. *Metamorphic, because anti-virus software detects based upon known code or signatures of virus. If the anti-virus knows the code for a virus to be V, then it looks for that code in files. However if the virus changes its code to X, then anti-virus would also need to know X and compare files to both V and X. Metamorphic is harder to detect than polymorphic because with polymorphic there are only a limited number of combinations of instructions that still result in the same behaviour. Metamorphic allows many more combinations.*

Question 4 [5 marks]

Listing 1 shows a set of packets captured when using SSH (further packets were captured beyond frame 38; they are not shown). Listing 2 shows details for selected individual packets from Listing 1. For clarity, some information that is not necessary for answering questions has been removed. Also, some values have been changed to make calculating answers easier.

Listing 1: SSH Packet List

| No. | Time | Source | Dest. | Proto | Info |
|-----|----------|---------|---------|-------|---|
| 16 | 0.133487 | 1.1.1.1 | 2.2.2.2 | SSHv2 | Server Protocol: SSH-2.0-OpenSSH_4.7p1 Debian-8 |
| 18 | 0.133642 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Client Protocol: SSH-2.0-OpenSSH_5.3p1 Debian-3 |
| 20 | 0.158486 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Client: Key Exchange Init |
| 21 | 0.159471 | 1.1.1.1 | 2.2.2.2 | SSHv2 | Server: Key Exchange Init |
| 24 | 0.212451 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Client: Diffie-Hellman GEX Request |
| 26 | 0.235424 | 1.1.1.1 | 2.2.2.2 | SSHv2 | Server: Diffie-Hellman Key Exchange Reply |
| 28 | 0.238691 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Client: Diffie-Hellman GEX Init |
| 29 | 0.283398 | 1.1.1.1 | 2.2.2.2 | SSHv2 | Server: Diffie-Hellman GEX Reply |
| 31 | 0.321912 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Client: New Keys |
| 33 | 0.369375 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Encrypted request packet len=48 |
| 35 | 0.382345 | 1.1.1.1 | 2.2.2.2 | SSHv2 | Encrypted response packet len=48 |
| 37 | 0.819498 | 2.2.2.2 | 1.1.1.1 | SSHv2 | Encrypted request packet len=64 |
| 38 | 0.850053 | 1.1.1.1 | 2.2.2.2 | SSHv2 | Encrypted response packet len=64 |
| ... | | | | | |

Listing 2: SSH Packet Details

Frame 20 (858 bytes on wire, 858 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 788

Padding Length: 8

Key Exchange

Msg code: Key Exchange Init (20)

Algorithms

kex_algorithms string: diffie-hellman-group-exchange-sha256

server_host_key_algorithms string: ssh-rsa

encryption_algorithms_client_to_server string: aes128-ctr,aes192-ctr,aes256-ctr

encryption_algorithms_server_to_client string: aes128-ctr,aes192-ctr,aes256-ctr

mac_algorithms_client_to_server string: hmac-md5,hmac-sha1

mac_algorithms_server_to_client string: hmac-md5,hmac-sha1

compression_algorithms_client_to_server string: none

compression_algorithms_server_to_client string: none

KEX First Packet Follows: 0

Frame 21 (850 bytes on wire, 850 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 780

Padding Length: 10

Key Exchange

Msg code: Key Exchange Init (20)

Algorithms

kex_algorithms string: diffie-hellman-group-exchange-sha256

server_host_key_algorithms string: ssh-rsa

encryption_algorithms_client_to_server string: aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr

encryption_algorithms_server_to_client string: aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr

mac_algorithms_client_to_server string: hmac-md5,hmac-sha1

mac_algorithms_server_to_client string: hmac-md5,hmac-sha1

compression_algorithms_client_to_server string: none

compression_algorithms_server_to_client string: none

KEX First Packet Follows: 0

Frame 24 (90 bytes on wire, 90 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 20

Padding Length: 6

Key Exchange

Msg code: Diffie-Hellman GEX Request (34)

DH GEX Min: 00000008

DH GEX Numbers of Bits: 00000008

DH GEX Max: 0000000F

Frame 26 (218 bytes on wire, 218 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 148

Padding Length: 8

Key Exchange

Msg code: Diffie-Hellman Key Exchange Reply (31)

Multi Precision Integer Length: 129 (decimal)

DH modulus: 239 (decimal)

Multi Precision Integer Length: 1 (decimal)

DH base: 7 (decimal)

Frame 28 (210 bytes on wire, 210 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 140

Padding Length: 6

Key Exchange

Msg code: Diffie-Hellman GEX Init (32)

Multi Precision Integer Length: 128 (decimal)

DH client e: 184 (decimal)

Frame 29 (786 bytes on wire, 786 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 700

Padding Length: 9

Key Exchange

Msg code: Diffie-Hellman GEX Reply (33)

KEX DH host key length: 277 (decimal)

KEX DH host key: 000000077373682D727361000000012300000101009C5052...

Multi Precision Integer Length: 129 (decimal)

DH server f: 122 (decimal)

KEX DH H signature length: 271 (decimal)

KEX DH H signature: 000000077373682D72736100000100172CEA9394795589C5...

MAC: 0000000C0A150000000000000000000000

Frame 31 (82 bytes on wire, 82 bytes captured)

SSH Protocol

SSH Version 2

Packet Length: 12

Padding Length: 10

Key Exchange

Msg code: New Keys (21)

Frame 33 (114 bytes on wire, 114 bytes captured)

SSH Protocol

SSH Version 2

Encrypted Packet: 4F8BD30EB384B2AB713CA1785F17CBF17410E9F4DD82783C...

MAC: 1751A0F06C0C13EB2C4478C4

Frame 35 (114 bytes on wire, 114 bytes captured)

SSH Protocol

SSH Version 2

Encrypted Packet: 912DAFF5E864321439AA2454496AC4D5539E350BE7F3833D...

MAC: BFD7C9EDEB3926E3CB36E29B

(a) What block cipher mode of operation is used in Frame 33? [1 mark]

Answer. *Counter mode of operation (ctr). The first algorithm that the client proposed that the server also supports is selected, i.e. aes-128-ctr.*

(b) What is the key length used in the encryption in Frame 33? [1 mark]

Answer. *128 bit*

(c) What MAC algorithm is used in Frame 33? [1 mark]

Answer. *HMAC-MD5*

In SSH the client authenticates the server based on the public key of the server (which is assumed to be known by the client).

(d) After receiving which frame can the client authenticate the server in the above SSH connection? Explain why you selected the frame. [2 marks]

Answer. *Frame 29, because it contains data signed by the server. When the client receives this, it uses the known public key of the server to decrypt and verify.*

(a) Whose certificate is this? [1 mark]

Answer. *The user CSAIT*

(b) Whose key is used to create the signature? [1 mark]

Answer. *The user SIITTU*

(c) Assume your computer received the above certificate. To verify the certificate your computer (or an application on it) performs the following test:

```
if condition then 'verified'  
else 'error'
```

Write a statement for `condition` (e.g. `A==B`). In your statement you must use the names of algorithms and values from the above certificate (i.e. you cannot use `E()`), as well as clearly identify owners of keys. Use the field names from the certificate in your statement. [3 marks]

Answer. *The certificate is verified if the signature is correct. To determine if it is correct the signature is decrypted using RSA and the Public Key of the issuer and compared to the SHA1 hash of the data. The answer is: $RSA(PU_{SIITTU}, Signature) = SHA1(Data)$*

Question 6 [10 marks]

Consider the mechanism illustrated in Figure 1.

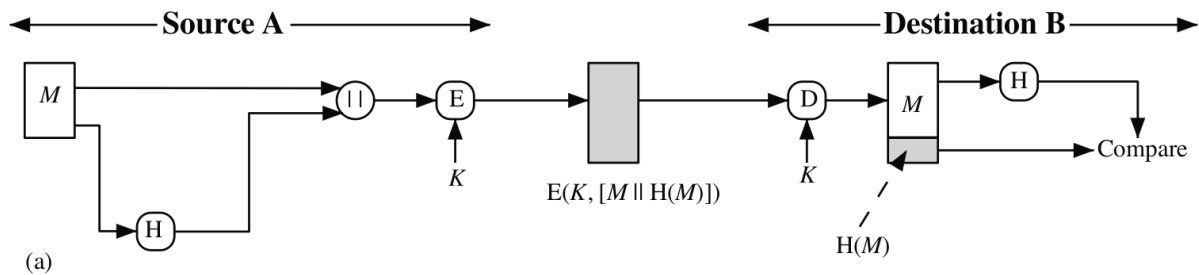


Figure 1: Security mechanism 1

- (a) List all security services provided by this mechanism. [2 marks]

Answer. *Authentication, data integrity, confidentiality.*

- (b) Explain (or define) the *weak collision resistant property* (also called *second pre-image resistant property*) of a hash function. [2 marks]

Answer. *Computationally hard to find a message y such that $H(x) = H(y)$, given the hash function and x*

- (c) If the function $H()$ does not satisfy the weak collision resistant property, then explain what an attacker can attempt to do to defeat the above security service. [3 marks]

Answer. *The attacker cannot do anything: the hash is encrypted and the attacker does not know its value. In this scheme there are two forms of protection against masquerade and modification attacks. One is the hash and the second is the symmetric key encryption. Even if the hash function is insecure, the attacker cannot modify the message to perform an attack. This assumes the encryption cipher is secure and that the receiver can detect structure in the received message.*

- (d) Explain (or define) the *strong collision resistant property* (also called *collision resistant property*) of a hash function. [2 marks]

Answer. *Computationally hard to find any two messages x and y such that $H(x) = H(y)$, given the hash function*

- (e) Which property is easier for an attacker to defeat: weak collision resistant or strong collision resistant? [1 mark]

Answer. *Strong collision resistant*

Question 7 [5 marks]

Your web browser has just accessed `https://it.siiit.tu.ac.th/moodle`. Your computer is using an Ethernet LAN card.

- (a) Draw a protocol stack that shows the specific protocols that the data from your web browser passes via. [2 marks]

Answer.

HTTP

SSL/TLS

TCP

IP

Ethernet DLL

Ethernet PHY

Your web browser has established a secure session and connection to a web server. The browser stores the following information about the session/connection.

- Session ID: id
- Compression method: null
- CipherSuite: `TLS_DH_RSA_WITH_AES_CTR_MD5`
- Master secret: m
- Server MAC secret: x_s
- Client MAC secret: x_c
- Server random: y_s
- Client random: y_c
- Server encrypt key: z_s
- Client encrypt key: z_c

Figure 2 shows the general operation of the SSL record protocol.

- (b) Write an equation that expresses the SSL record operation on a single data fragment, D , from the web server that produces the packet to be sent P . Use the variables above and `||` for the concatenate/append operator. For function names you *must* use the algorithm names (i.e. you cannot use `E()` for encrypt, `H()` for hash; refer to specific algorithms). Denote the SSL header as S . [3 marks]

Answer.

$$P = S || \text{AES_CTR}(z_s, D) || \text{HMACMD5}(x_s, D)$$

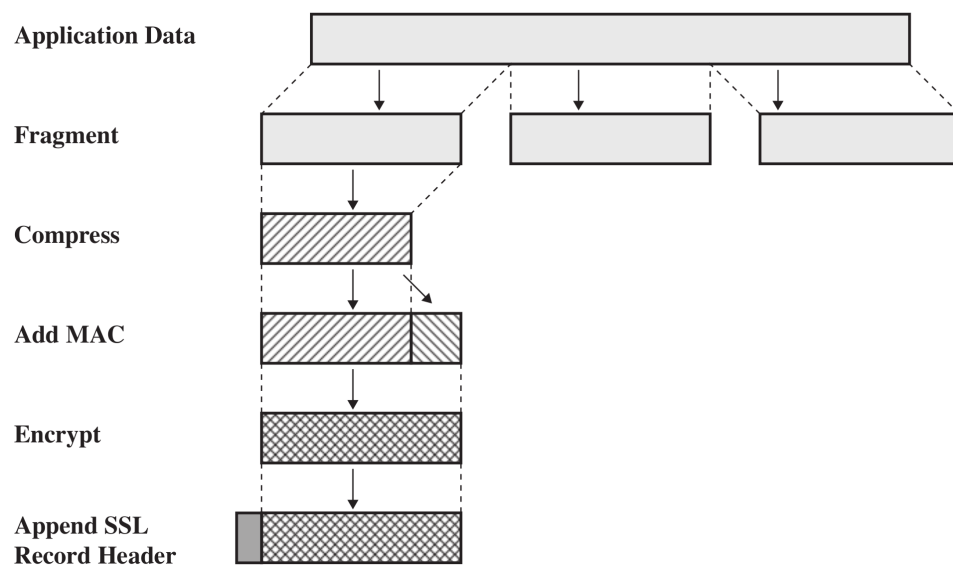
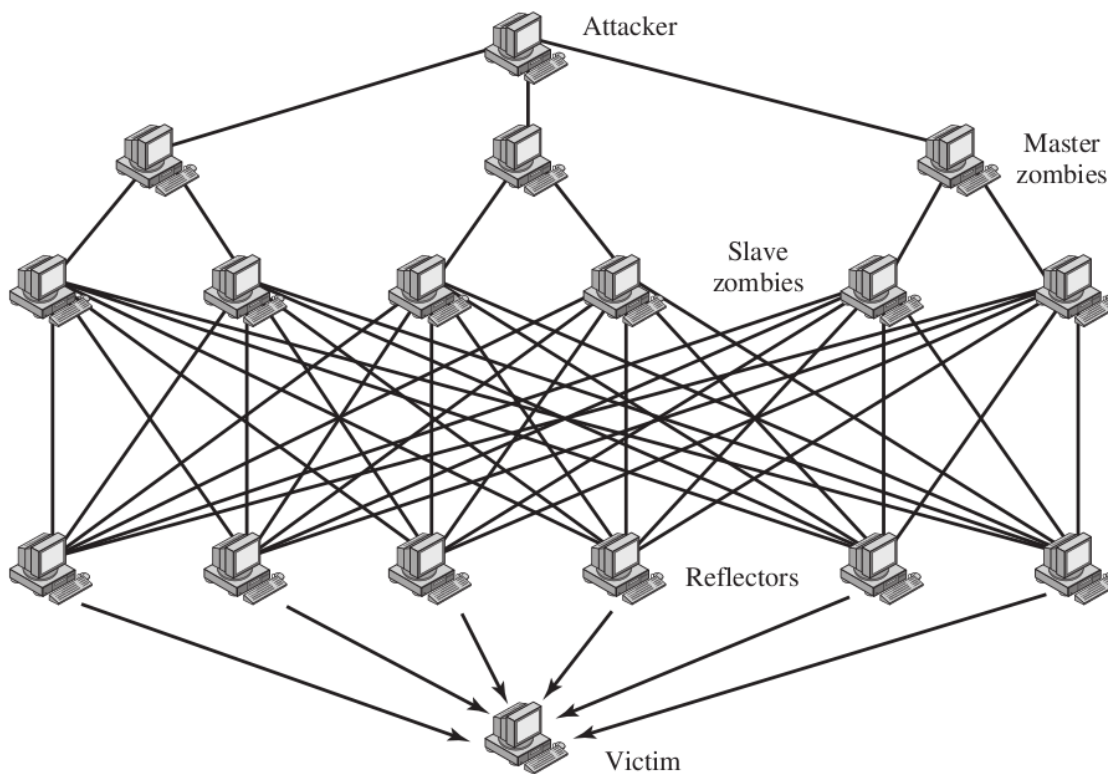


Figure 2: SSL Record Protocol Operation

Question 8 [9 marks]

- (a) Draw a diagram that illustrates a reflector DDoS attack. Show (and label) the nodes involved and the direction of messages. [2 marks]



- (b) Of the nodes involved in the reflector attack above, which nodes are controlled (or infected) by the malicious user? [1 mark]

Answer. *Attack machine and Slave nodes (not reflectors)*

- (c) What are two advantages of a reflector DDoS attack compared to a direct DDoS attack? [2 marks]

Answer. *Easier to get more nodes to attack target, as they (the reflectors) do not need to be under control from the attacker—they can be almost any computer on the Internet. Further separation between the attacker and target, so that trace the attack to the target is harder (must go through nodes not under control of attacker).*

- (d) What is one advantage of a direct DDoS attack compared to a reflector DDoS attack? [1 mark]

Answer. *Wider range of protocols can be used, as the nodes performing the attack are under control of the attacker and can potentially implement any protocol. With a reflector attack the protocol used must be supported by the reflectors.*

- (e) DDoS attacks can be classified by the type of resource consumed. What type of resource does an ICMP Ping attack consume? [1 mark]

Answer. *Network bandwidth*

- (f) What type of resource does a TCP SYN flooding attack consume? [1 mark]

Answer. *Memory of target computer*

- (g) What service do DDoS attacks try to defeat? [1 mark]

Answer. *Availability*

Question 9 [6 marks]

- (a) User A wants to send a MAC authenticated message M to B . Give an equation that describes what is sent to B , i.e. $Sent = \dots$. You must also describe all variables used. [2 marks]

Answer. $Sent = M || MAC(S, M)$ where S is a shared secret key with B .

- (b) Assume a malicious user C intercepts the message sent by A . C modifies the message M . Can B detect this modification? Explain your answer (i.e. what B does to detect or why B cannot detect). [2 marks]

Answer. Yes, B can detect the modification. If M is modified to M' then upon reception B uses the shared secret key S to calculate the MAC of the receive message, M' , i.e. $MAC(S, M')$. B finds $MAC(S, M') \neq MAC(S, M)$ and hence has detected a modification.

- (c) Explain why MAC-based authentication cannot be used as a digital signature. [2 marks]

Answer. A MAC function uses a shared secret key. A message authenticated with a MAC function confirms that the message was generated by either the of the parties that has the secret. It does not confirm which of the two parties generated the message (which is the purpose of a digital signature).

Question 10 [12 marks]

Consider the key distribution protocol illustrated in Figure 3.

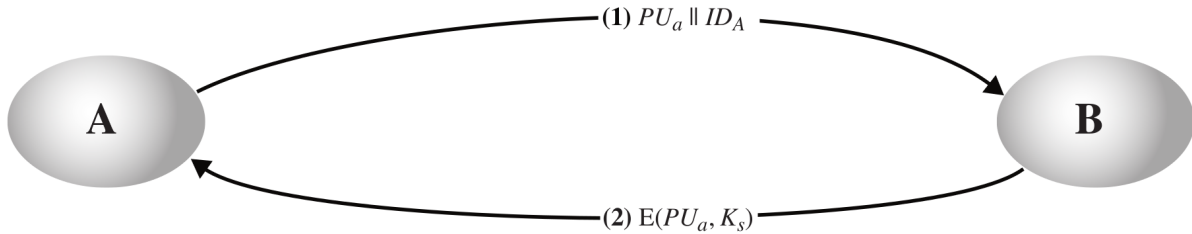


Figure 3: Key distribution 1

- (a) What key is this protocol trying to distribute? [1 mark]

Answer. For A and B to share a secret key K_s

- (b) Draw a diagram that illustrates how a malicious user C can perform a man-in-the-middle attack. Clearly label all messages sent. [3 marks]

Answer. $A \rightarrow C : PU_a || ID_A$

$C \rightarrow B : PU_c || ID_A$

$B \rightarrow C : E(PU_c, K_s)$

$C \rightarrow A : E(PU_a, K_s)$

Consider the key distribution protocol illustrated in Figure 4. *SG Update: figure should refer to (c), (d), (e) and (f)*

There are four missing values in the figure. What are they?

- (c) [1 mark]

Answer. K_a

- (d) [1 mark]

Answer. K_s

- (e) [1 mark]

Answer. $E(K_b, [K_s || ID_A])$

- (f) [1 mark]

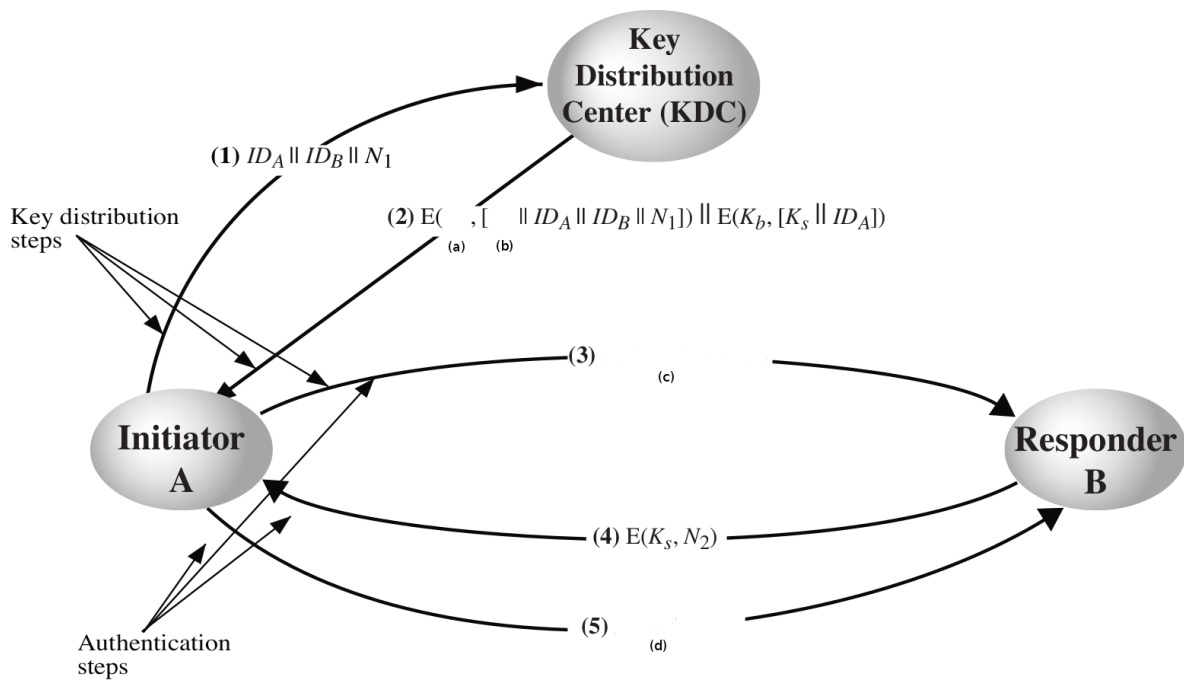


Figure 4: Key distribution 2

Answer. $E(K_s, f(N_2))$

Assume there are 1000 users in a network using the key distribution protocol of Figure 4.

(g) What is the maximum number of master keys in the network at any one time? [1 mark]

Answer. 1000. Each user exchanges a master key with the KDC

(h) What is the maximum number of session keys in the network at any one time? [1 mark]

Answer. 499,500. Each user exchanges a session key with every other user

(i) What is the advantage of using both master and session keys (compared to just using master keys)? [2 marks]

Answer. As the session keys are automatically distributed (while master keys are manually exchanged with the KDC), it is easy to regularly change the session keys. The fewer times a key is used to encrypt data in packets, the less change has of finding that key (or finding data if the key has been compromised). With just master keys, they are manually exchanged and therefore there is too much effort needed to continually change them if the master key is used to encrypt data.