Name .... ID .... Section ...... Seat No ......

Sirindhorn International Institute of Technology Thammasat University

Final Exam Answers: Semester 2, 2010

Course Title: CSS322 Security and Cryptography

Instructor: Steven Gordon

Date/Time: Wednesday 2 March 2011; 9:00–12:00

Instructions:

- This examination paper has 22 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, 2010

Prepared by Steven Gordon on 24 February 2011
CSS322Y10S2E02, Steve/Courses/2010/S2/CSS322/Assessment/Final-Exam.tex, r1704

Question 1 [9 marks]

Consider a system with 26 users (e.g. user A, user B, . . . user Z). Confidentiality of communications between users must be provided using symmetric key cryptography. Figures 1 and 2 show two alternative protocols for key distribution in the system for an example when user A wants to communicate with user B. First consider the protocol in Figure 1.

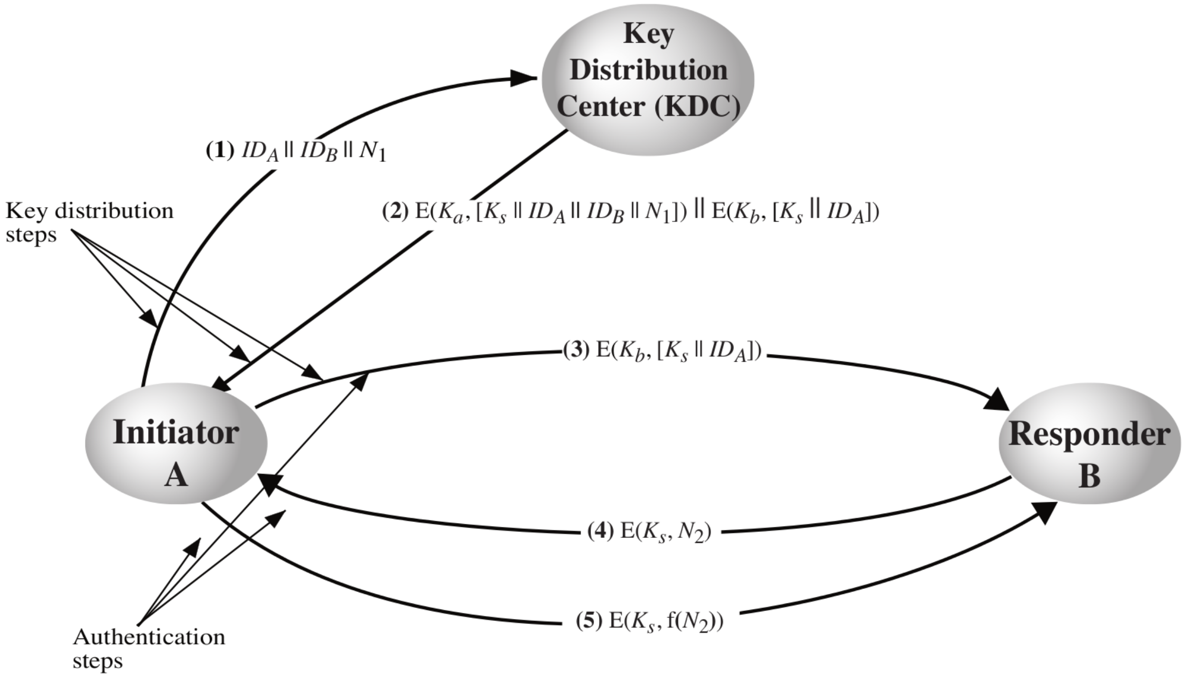


Figure 1: Key distribution protocol 1

(a) What is the set of keys that is assumed to be known by each entity *before* the protocol is applied? [2 marks]

Answer. *User A knows* K_a*; user B knows* K_b*; . . . ; user Z knows* K_z*; and KDC knows* K_a, K_b, . . . , K_z

(b) What is the set of additional keys that are known by each entity *after* the protocol is applied? (that is, in addition to the keys known in part (a)) [1 mark]

Answer. *User A also* K_s*; user B also knows* K_s*; and KDC also knows* K_s

(c) If an attacker intercepts all five messages during the protocol operation, list all the items that the attacker will know. [1 mark]

Answer. ID_A, ID_B, N_1

(d) If after the protocol operation (i.e. all five messages are sent) an attacker later replays message (3), explain how the replay attack would be detected. [2 marks]

Answer. *User B responds with message (4), containing a random nonce encrypted with K_s. B is then expecting message (5) in return (if it does not receive it or receives it with the wrong nonce, then the attack is detected). If the malicious user intercepts message (4) it cannot determine N_2 because it doesn't know K_s, therefore B will not receive the expected response (attack detected). If user A receives message (4) then the attack is detected because A wasn't expecting this message since A did not send message (3).*

Now compare the protocol in Figure 1 with the protocol in Figure 2.

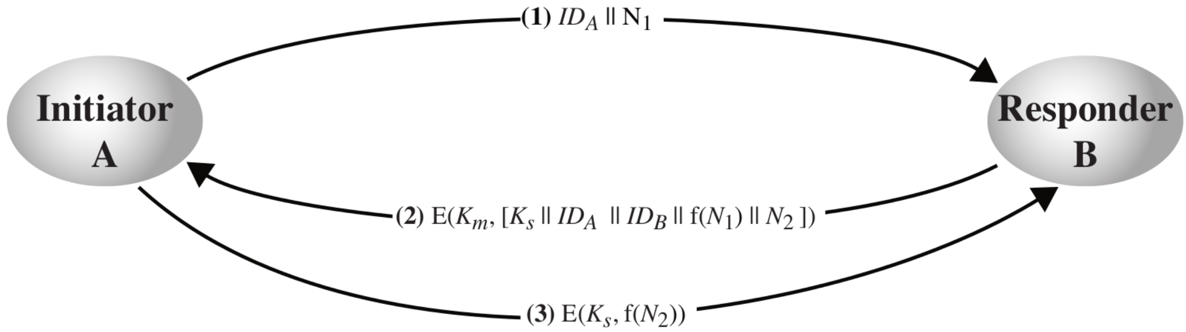


Figure 2: Key distribution protocol 2

(e) What is the total number of keys that user A is assumed to know *before* the protocol is applied in Figure 2? [1 mark]

Answer. *User A must share master keys with all other users, i.e. 25*

(f) Explain an advantage of the protocol in Figure 1 compared to that in Figure 2? [1 mark]

Answer. *Fewer keys to be manually distributed before the protocol operation.*

(g) One advantage of using the protocol in Figure 2 (compared to that in Figure 1) is that it avoids performance bottlenecks at KDC. Explain another advantage of Figure 2. [1 mark]

Answer. *No need to trust KDC*

Question 2 [3 marks]

Consider two sets of rules for generating random passwords:

Option 1 Select 10 characters from the set of: English uppercase and lowercase characters and five punctuation characters _ , . ? !

Option 2 Select 16 characters from the set of: digits and four operators + - / *

(a) Which option produces the strongest passwords? Explain your answer. [3 marks]

Answer. *Option 2. With option 1 there are 57 possible characters giving an entropy of* $10 \times log_2(57) = 58.3$*. With option 2 there are 14 possible characters giving an entropy of* $16 \times log_2(14) = 60.9$*. Effectively, option 2 is equivalent to a 60-bit random string, which will take long to apply brute force against compared to a 58-bit random string (the strength of option 1).*

Question 3 [10 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: 0000000C0A15000000000000000000000

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*

The key exchange algorithm used in the above SSH connection is Diffie-Hellman. The general Diffie-Hellman key exchange algorithm is shown in Figure 3. The global public values that must first be exchanged are the base α and the modulus q.

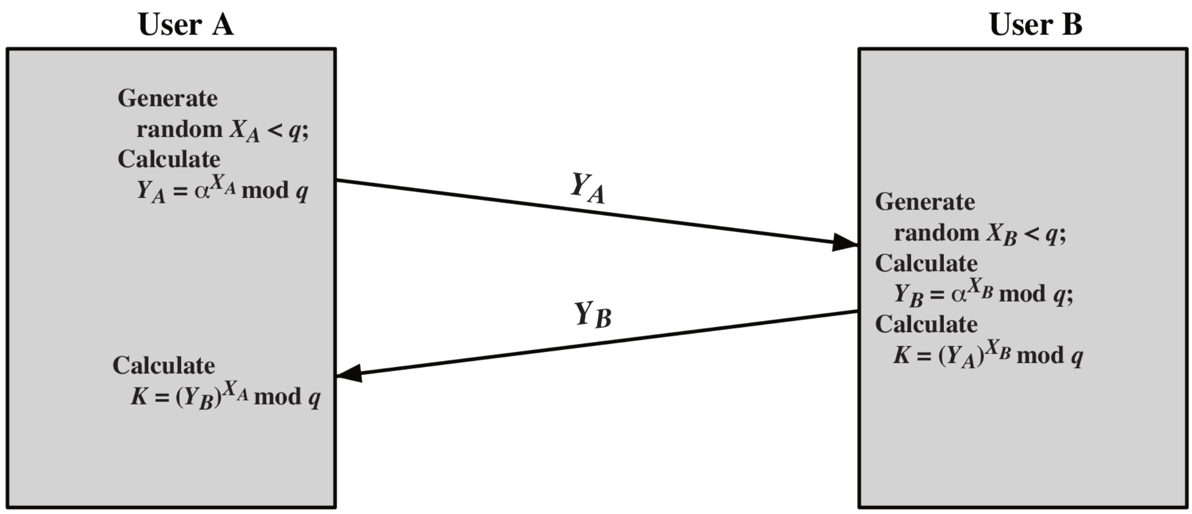


Figure 3: Diffie-Hellman Key Exchange algorithm

(d) If in the Diffie-Hellman exchange the SSH client selected a private $X = 23$, then what is the value of the secret, K, agreed upon from the SSH key exchange? Show your calculations. [3 marks]

Answer. *The packet capture shows the values of* $\alpha = 7$ *and* $q = 239$ *in frame 26. In frame 29 the server sends its DH value (f or* Y_B*) to the client, 122. Therefore* $K = 122^{23} \bmod 239 = 213$.

(e) If an attacker intercepted all packets, explain what equation the attacker would need to solve to discover the secret, K. You must refer to the specific values from the packet capture, not just the variables in Figure 3. [2 marks]

Answer. *The attacker knows* $Y_A = 184$, $Y_B = 122$, $\alpha = 7$ *and* $q = 239$. *To find* K *they must find* X_A *or* X_B, *by solving for example* $184 = 7^{X_A} \bmod 239$.

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

(f) 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.*

Question 4 [6 marks]

Consider the X.509 certificate in Listing 3.

Listing 3: X.509 Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=South Australia, L=Adelaide, O=ABC, OU=Security,
                CN=ABC Security/emailAddress=security@abc.com.au
        Validity
            Not Before: Jan 25 02:25:10 2011 GMT
            Not After : Jan 25 02:25:10 2012 GMT
        Subject: C=TH, ST=Pathumthani, O=TrustUs, OU=Crypto,
                 CN=TrustUsCrypto/emailAddress=crypto@trustus.co.th
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:aa:1f:cf:01:2f:d3:2e:80:63:98:1b:0f:16:5d:
                    dd:af:e2:38:de:78:88:56:b6:14:2b:61:79:92:0b:
                    f3:7f:b6:89:7b:d0:fc:59:5a:1a:be:24:61:39:d5:
                    4d:80:3a:40:2b:7c:89:ef:5e:50:a5:3b:44:68:a9:
                    7f:97:d9:c4:9a:bf:b6:97:eb:4c:87:0d:00:96:b4:
                    f9:ea:8c:6a:cb:e0:bd:f8:a8:1f:82:d3:2b:23:3c:
                    b6:54:85:37:5b:13:1a:2e:be:0d:20:52:c5:98:b6:
                    4c:97:67:6e:b2:43:04:3f:01:41:8e:e0:2f:38:1f:
                    e1:cc:cf:0d:c2:5f:0a:04:a3
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                EA:1C:DC:C5:16:F2:9D:BC:61:5E:A8:D2:67:2A:06:13:C5:64:8A:AE
            X509v3 Authority Key Identifier:
                keyid:61:52:40:EA:7F:E0:EC:77:41:F6:4F:6F:7C:49:EB:05:C1:56:6D:49

    Signature Algorithm: sha1WithRSAEncryption
        a5:7a:36:91:ef:11:46:58:74:37:87:81:7a:99:ff:b6:40:4a:
        80:6a:07:69:e3:3c:33:9a:fd:31:50:e9:9f:bf:ff:36:a4:34:
        21:50:49:70:e0:88:b3:01:c9:51:26:8b:1e:8b:34:ca:4c:3c:
        a2:ab:0a:a3:b3:39:c0:fb:88:72:98:69:c9:af:42:b2:48:1b:
        4e:4a:76:e8:b4:c7:d4:f8:15:d2:5e:f8:69:fc:53:0c:ca:85:
        84:ea:e5:36:17:20:65:fc:d0:3e:d1:33:17:f7:d1:40:f8:3d:
        2a:87:f8:3c:66:8e:43:62:ea:02:ef:7a:d4:a7:55:e9:d9:2d:
        38:1a
-----BEGIN CERTIFICATE-----
MIIC5zCCAlCgAwIBAgIBAzANBgkqhkiG9w0BAQUFADCBnzELMAkGA1UEBhMCVEgx
GDASBgNVBAgTC1BhdGh1bXRoYW5pMREwDwYDVQQHEwhCYW5na2FkaTENMAsGA1UE
ChMEU0lJVDEMMAoGA1UECxMDSUNUMR4wHAYDVQQDExVDZXJ0aWZpY2F0ZSBBdXRo
b3JpdHkxKjAoBgkqhkiG9w0BCQEWG2NzczMyMi1jYUBpY3Quc2lpdC50dS5hYy50
aDAeFw0xMTAxMjUwMjI1MTBaFw0xMjAxMjUwMjI1MTBaMFYxCzAJBgNVBAYTAlRI
MRQwEgYDVQQIEwtQYXRodW10aGFuaTENMAsGA1UEChMEU0lJVDEMMAoGA1UECxMD
SUNUMRQwEgYDVQQDEwtEZW1vIFVzZXIgMjCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEAqh/PAS/TLoBjmBsPFl3dr+I43niIVrYUK2F5kgvzf7aJe9D8WVoaviRh
OdVNgDpAK3yJ715QpTtEaKl/l9nEmr+2l+tMhw0AlrT56oxqy+C9+KgfgtMrIzy2
VIU3WxMaLr4NIFLFmLZMl2duskMEPwFBjuAvOB/hzM8Nwl8KBKMCAwEAAaN7MHkw
CQYDVR0TBAIwADAsBglghkgBhvhCAQ0EHxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2Vy
dGlmaWNhdGUwHQYDVR0OBBYEFOoc3MUW8p28YV6o0mcqBhPFZIquMB8GA1UdIwQY
MBaAFGFSQOp/4Ox3QfZPb3xJ6wXBVm1JMA0GCSqGSIb3DQEBBQUAA4GBAKV6NpHv
EUZYdDeHgXqZ/7ZASoBqB2njPDOa/TFQ6Z+//zakNCFQSXDgiLMByVEmix6LNMpM
PKKrCqOzOcD7iHKYacmvQrJIG05Kdui0x9T4FdJe+Gn8UwzKhYTq5TYXIGX80D7R
Mxf30UD4PSqH+DxmjkNi6gLvetSnVenZLTga
-----END CERTIFICATE-----
```

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

Answer. *The user* TrustUsCrypto

(b) Whose RSA key is included in the certificate? [1 mark]

Answer. *The user* TrustUsCrypto

(c) The RSA algorithm is: $C = M^e \bmod n$. What are the last two hexadecimal digits of e in the users RSA key? [1 mark]

Answer. *The exponent, e, is 65537 in decimal, or 10001 in hex. The answer is 01.*

(d) What are the last two hexadecimal digits of n in the users RSA key? [1 mark]

Answer. *The modulus, n, is given in hex and ends with a3.*

In general, an X.509 certificate for user A can be expressed as:

$$C_A = Data||S$$

where $Data$ is the concatenation of the fields: Version, SerialNumber, SignatureAlgorithm, Issuer, Validity, Subject, SubjectPublicKeyInfo and X509v3extensions.

(e) Write an equation for how S is calculated in the certificate in Listing 3? You must use the names of algorithms used in the above certificate (i.e. you cannot use E()), as well as clearly identify which user each key belongs to. You may use the variable $Data$ in your equation to represent the concatenation of various fields. [2 marks]

Answer.

$$S = RSA(PR_{ABCSecurity}, SHA1(Data))$$

Question 5 [5 marks]

Listing 4 shows the pseudocode of a simple virus.

Listing 4: Pseudocode of a simple virus

```
1. program V {
2.     goto main;
3.     abcd1234;
4.     function infect-executable() {
5.             new-files-infected = 0;
6.             while (new-files-infected < 2) {
7.                     file = select-random-executable-file();
8.                     if (!file-includes-special-string(abcd1234)) {
9.                             prepend V to file;
10.                            new-files-infected++;
11.                    }
12.            }
13.    }
14.    function do-damage() {
15.            file-list = select-files-to-delete();
16.            delete-files(file-list);
17.    }
18.    function condition-true() {
19.            if (condition)
20.                    return true;
21.            else
22.                    return false;
23.    }
24. main: main-program() {
25.            infect-executable();
26.            if condition-true()
27.                    do-damage();
28.            goto next;
29.    }
30. next:
31.            <original program>
32. }
```

(a) A logic bomb may use conditions based on date and time. If this virus was also a logic bomb, give an example of another condition (not based on date or time). [1 mark]

Answer. *Absence or presence of a file.*

(b) What line(s) of code would you modify to implement the logic bomb? [1 mark]

Answer. *Line 19, which checks the condition is true.*

(c) How many other files does this virus infect? Explain your answer. [1 mark]

Answer. *2. Lines 5 to 10 loop until two files have been infected.*

(d) Will this virus infect a file that is already infected with the virus? Explain your answer. [1 mark]

Answer. *No. Line 8 checks whether the file found is already infected.*

(e) Which type of virus is Listing 4: parasitic, metamorphic or polymorphic? Explain your answer (i.e. why it is one type, not the others). [1 mark]

Answer. *This is a parasitic virus. It attaches itself to other programs. It is not polymorphic or metamorphic because it does not change its appearance or behaviour.*

Question 6 [9 marks]

You are developing a shopping website for a company. The website allows users to register (they are given a random, 6-digit user ID and can select any password between 8 and 12 characters in length), login to obtained personalised content and services, as well as to purchase products and services using supplied credit card information. The company runs the web server, as well as a database server for storing user and product information.

(a) What protocol(s) should be used so that information transferred between users and the web server is confidential? [1 mark]

Answer. *HTTPS (or HTTP and SSL), as it provides encryption of data between web browser and server*

(b) The company has obtained a digital certificate issued by the authority VeriSign. Explain how this certificate can be used for web server authentication. (Include any assumptions about the web server or browser). [2 marks]

Answer. *The certificate is sent to the web browser. The web browser must have the certificate of the authority VeriSign. The browser uses VeriSigns certificate to verify the servers certificate, proving that the client is communicating with the intended server.*

(c) Certificates are generally not used for client (user) authentication. Explain then how client authentication is performed (including any assumptions). [1.5 marks]

Answer. *Once a secure connection is established between client and server, the user provides a username and password. The client is authenticated if the supplied username/password match the one selected during registration.*

(d) When a new user registers with the website, explain what identifying information must be stored in the database. [1.5 marks]

Answer. *At least the username/ID and and a hash of the password*

You decide to use a 10-digit *salt* value when implementing the registration/login system.

(e) Explain why using a salt decreases the chance of successful online password guessing. [1 marks]

Answer. *BAD QUESTION: A salt doesn't increase security against online password guessing as the attacker does not need to know the salt*

(f) In addition to the salt, describe two methods you would implement that could prevent or deter online password guessing. [2 marks]

Answer. *Limit the number of incorrect attempts, e.g. to 10. Introduce a delay between incorrect attempts, e.g. 5 seconds before another attempt can be made. Log all incorrect attempts, reporting them to the user once they log in.*

Question 7 [12 marks]

(a) Assuming two primes, $p = 7$ and $q = 19$, generate and fill in the spaces below for a RSA key pair. [4 marks]

PU = {e=5, _____}; PR = {_____, _____}

Answer. *Calculate as follows:*

$$n = pq = 133$$

$$\phi(n) = (p - 1)(q - 1) = 108$$

$$ed \bmod \phi(n) \equiv 1; d = 65$$

Hence PU = {5, 133}; PR = {65, 133}

(b) Assume you have another users RSA key, PU = {3,55}, and they sent a message $M = 17$ as well as a signature of that message $S = 8$ to you. Is the message authentic? (Show any calculations; assume no hash function is used when signing) [3 marks]

Answer. *The signature can be decrypted using the users public key:*

$$S^e \bmod n = 8^3 \bmod 55 = 17$$

Since the decrypted value matches the original message when we assume it is authentic (has not been modified)

(c) Describe the steps that an attacker could take to find the corresponding private key from part (b). [2 marks]

Answer. *Factor n into prime factors, p and q. Then calculate* $\phi(n) = (p - 1)(q - 1)$. *Then find multiplicative inverse of e in* $\bmod \phi(n)$, *i.e. d. Alternatively, manually calculate* $\phi(n)$.

(d) What is the value of the corresponding private key from part (b)? [2 marks]

Answer. *Using approach one from answer of part (c),* $p = 5$ *and* $q = 11$. *Therefore* $\phi(55) = 40$. $d = 27$ *is the multiplicative inverse of* $e = 3$ *in* $\bmod 40$. *Therefore PR = {27,55}.*

(e) Explain why in practical applications of RSA (e.g. using larger numbers than in this question), RSA is considered secure. [1 mark]

Answer. *Factoring a large number into its prime factors is computationally hard. As is manually determining Eulers totient of a large number.*

Question 8 [10 marks]

Consider the mechanism illustrated in Figure 4.

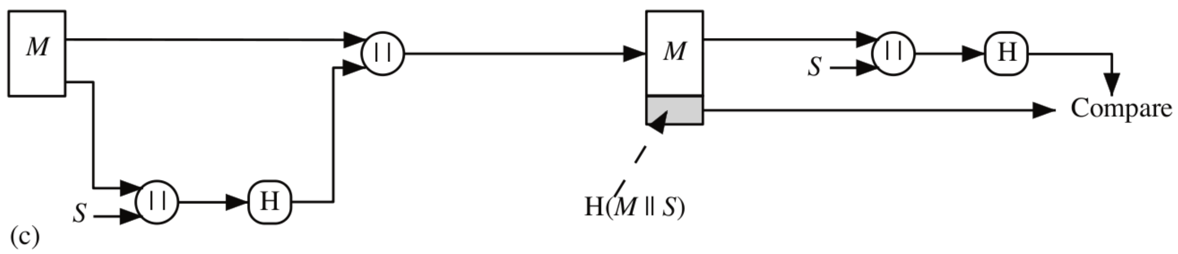


Figure 4: Security mechanism 1

(a) What is a security service that this mechanism provides? [1 mark]

Answer. *Authentication, data integrity*

(b) Explain (or define) the *one-way property* (also called *pre-image resistant property*) of a hash function. [1 mark]

Answer. *Computationally hard to determine the input of a hash function, given only the hash function and the output hash value*

(c) Explain how an attacker can defeat the above security service if the function H() did not have the one-way property. [2 marks]

Answer. *If the one-way property does not hold, then from the hash value,* $\mathrm{H}(M||S)$ *the attacker can find* $M||S$*. Since the attacker also knows* M *they can find* S*, the shared secret. Once they know the secret they could send a message to B, pretending to be A.*

Consider the mechanism illustrated in Figure 5

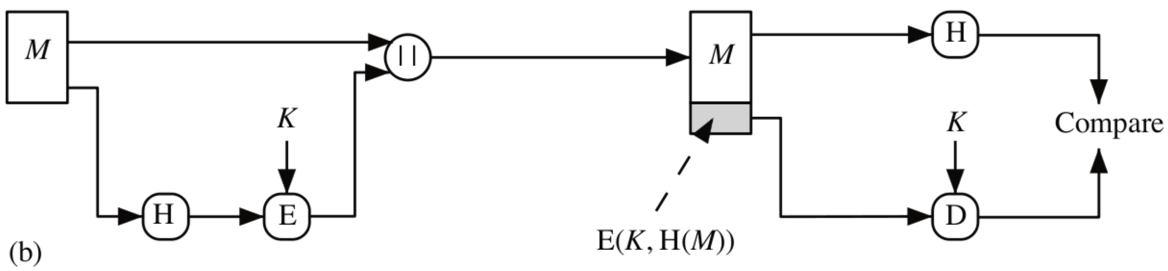


Figure 5: Security mechanism 2

(d) What is a security service that this mechanism provides? [1 mark]

Answer. *Authentication, data integrity*

(e) Explain (or define) the *weak collision resistant property* (also called *second pre-image resistant property*) of a hash function. [1 mark]

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

(f) Explain how an attacker can defeat the above security service if the function H() did not have the weak collision resistant property. [2 marks]

Answer. *If the weak collision resistant property does not hold, then the attacker can replace the message* M *sent by A with another message* y*, where* $H(M) = H(y)$*, and forward the message to B. B will not detect the change because after decrypting, the received hash value* $H(M)$ *will match the calculated hash value* $H(y)$*.*

(g) What is the difference between a hash function and a MAC function? [1 mark]

Answer. *A hash function takes data as input, while a MAC function takes data and a key as input*

(h) Explain what HMAC does when used with MD5. [1 mark]

Answer. *HMAC turns a hash function, MD5, into a MAC function*

Question 9 [5 marks]

A company has developed a new protocol, called *BAHTP*, that is used by a client application on computers in shops around Bangkok to send sales information to a central server in the company main office in Rangsit. The protocol uses TCP/IP. Based on your expert knowledge of OpenSSL libraries, you have been hired by the company to modify the client/server applications so that all communications between them are secure.

(a) Draw a protocol stack of a computer using Ethernet physical and data link layers, that illustrates the protocols in use by the secure client application. [2 marks]

Answer.

BAHTP

SSL/TLS

TCP

IP

Ethernet DLL

Ethernet PHY

When using the secure application, a secure session and connection has been established. The following information is stored by the client computer for this session/connection.

- Session ID: id
- Compression method: null
- CipherSuite: TLS_DH_RSA_WITH_DES_CBC_SHA
- Master secret: s
- Server random: r_s
- Client random: r_c
- Server MAC secret: m_s
- Client MAC secret: m_c
- Server encrypt key: e_s
- Client encrypt key: e_c

Figure 6 shows the general operation of SSL record protocol.

(b) Write an equation that expresses the SSL record operation on a single fragment, F from the client application 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 SSL. [2 marks]

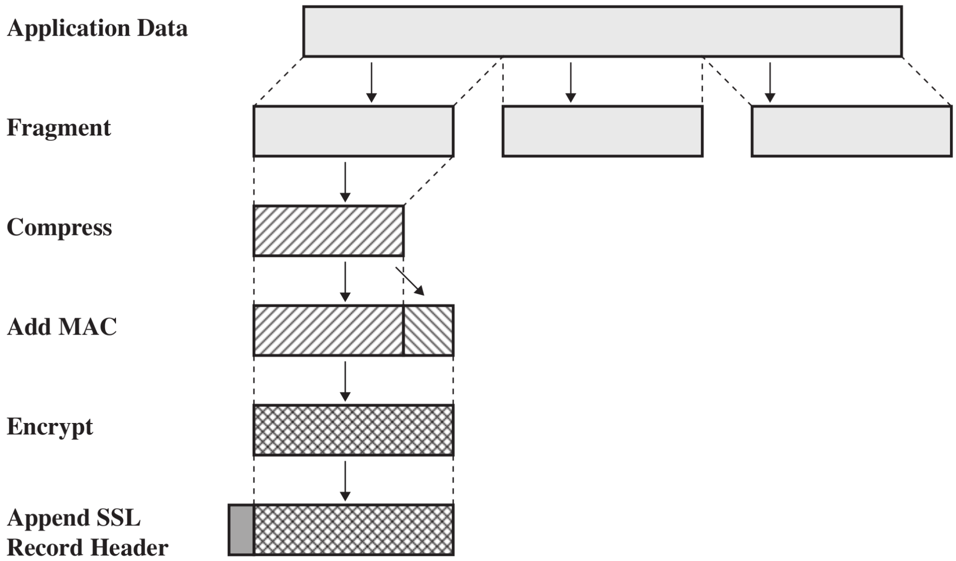


Figure 6: SSL Record Protocol Operation

Answer.

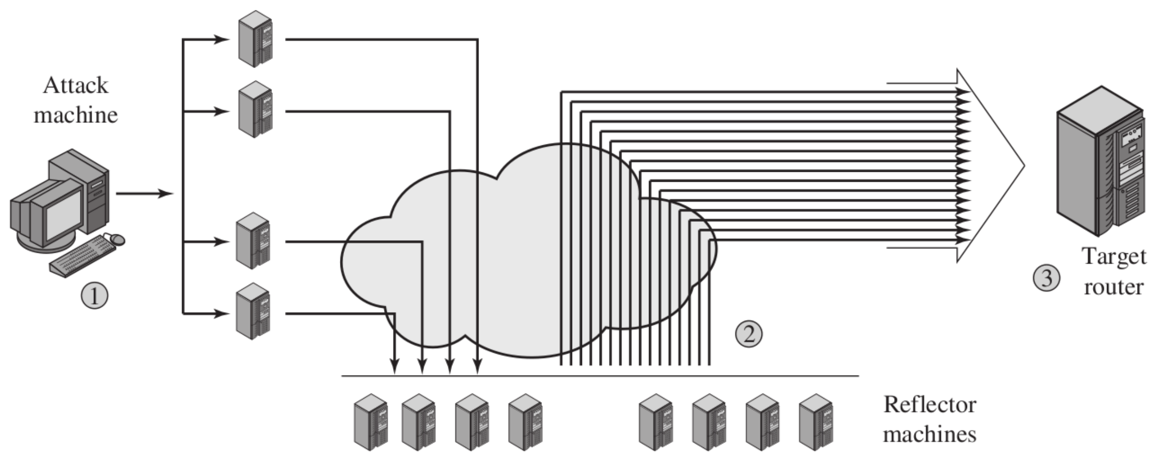
$$P = SSL||\text{DES\_CBC}(e_c, F||\text{HMACSHA}(m_c, F))$$

(c) Explain a security advantage of having multiple secrets/keys. [1 mark]

Answer. *In the above there is a master secret, as well as separate encrypt and MAC keys. An advantage is that the master secret is used very few times to encrypt data sent; instead the encrypt keys are, which may be changed regularly. Therefore an attacker has limited time to try to discover a key. Also, if one encrypt key is compromised, then other data may still be protected.*

Question 10 [6 marks]

(a) Draw a diagram that illustrates an ICMP Ping distributed denial of service attack. Show (and label) the nodes involved (including Attacker, Slaves, Reflectors and Target), the direction of messages and the types of messages. [3 marks]



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

Answer. *Attack machine and Slave nodes*

(c) A DoS makes a system (network and/or computers) unavailable for normal users to use. Explain how the ICMP attack achieves this, including what does it make "unavailable". [1 mark]

Answer. *The ICMP attack uses up network capacity leading up to the Target (or the network of the Target), thereby making that network capacity unavailable. In addition, the processing of ICMP ECHO replies may use up processing (CPU) capacity of the Target, making that host unavailable.*

(d) Explain the difference between a direct DDoS attack and a reflector DDoS attack. [1 mark]

Answer. *A direct DDoS attack involves hosts under the control of the attacker directly performing an attack (by sending messages) on a target. A reflector DDoS attack involves hosts under control sending messages to uncontrolled (normal) hosts in the network, that then perform an attack on the target.*