

①

Solutions - 2004 Master copy
Network Security E4.44

(a) $\text{gcd}(65, 537, 300) = 1$

300 is relatively prime to 65, 537
[4]

(b) Euler's Totient function $\phi(n)$ is the number of elements in \mathbb{Z}_n^* where \mathbb{Z}_n^* is the set of all positive integers less than n that are relatively prime to n

$$\phi(13p) = 12(p-1) \quad [4]$$

(c) Decryption in the Odd Round uses the same process as encryption but with inverse keys

K_a, K_d are replaced with their multiplicative inverses

K_b, K_c are replaced with their additive inverses

Decryption in the Even Round is identical to encryption as the Even Round is its own inverse

[4]

②

(d) $K_a = 6C60 \text{ hex} = 27,744 \text{ decimal}$

From Fig 1.1 $K_a^{-1} = 300 \text{ decimal} = 012C \text{ hex}$

$K_b = 02A1 \text{ hex}$

Additive inverse such that ~~$K_b + k$~~ is $F05F$

K_e and K_f remain the same

[9]

(e) DES and IDEA are block ciphers

encoding 64 bits plaintext into 64 bits ciphertext

Both employ Feistel structure

DES has 4 weak keys and 12 semi-weak keys which should be avoided

Exhaustive key search for DES requires

2^{56} operations whilst an exhaustive key

search for IDEA requires 2^{128} operations

(DES has 56 bit key, IDEA has 128 bit key)

IDEA is more secure than DES but

security of DES approaches that of IDEA

if 3DES (112 bit key) is used.

[4]

2 (5)

(a) RSA public key pair is (e, n)

where $n = p \times q$, p, q being large prime numbers each about 256 bits long and e is public key exponent and is relatively prime to $\phi(n)$

Private key d is such that $de = 1 \pmod{\phi(n)}$

To encrypt a message m ($< n$)

$$c = m^e \pmod{n}$$

To decrypt take $c^d \pmod{n}$

$$= m^{de} \pmod{n} = m^{de \pmod{\phi(n)}} \pmod{n}$$

$$= m$$

To choose (e, n)

Choose a number e (e.g. 3 or $2^{16} + 1$ for convenience) and find 2 large prime numbers p and q such that e is relatively prime to $(p-1)$ and $(q-1)$

$$\text{Then } n = p \times q$$

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

and private key d is multiplicative inverse of $e \pmod{\phi(n)}$

To check whether p, q are prime use Miller-Rabin test [6]

④

(b) ~~For~~ Calculating $c = m^e \pmod n$ is computationally intensive but the choice of 3 and $2^{16} + 1$ reduces the number of exponent operations to 3 and 17 respectively. This reduces the power required to encrypt or verify a signature.

Many principals may use 3 as public exponent with different moduli (e.g. $(3, n_1)$, $(3, n_2)$ etc) with different private keys d_1 , d_2 since

$$3 d_1 = 1 \pmod{\phi(n_1)}, 3 d_2 = 1 \pmod{\phi(n_2)} \quad [6]$$

(c) The use of PKCS has the benefit of a standard for interoperability purposes and offers protection against standard attacks such as the cube root attack for small n and $e = 3$.

The Message Digest function compresses the message to be signed typically to 128 or 160 bits which reduces the computation required for a signature.

[5]

⑤

(i) Let the 3 recipients have public key pairs $(3, n_1)$, $(3, n_2)$ and $(3, n_3)$ and the message be m

Then $c_1 = m^3 \bmod n_1$, $c_2 = m^3 \bmod n_2$

and $c_3 = m^3 \bmod n_3$

The attacker ^{would} know the public key pairs $(3, n_1)$, $(3, n_2)$ etc and from the Chinese Remainder

Theorem could compute $m^3 \bmod n_1 n_2 n_3$

since $m < n_1, n_2, n_3$. decryption is simply taking the cube root

[3]

(ii) Let $c_1 = m^3 \bmod n_1$, where m is less than a third of n_1 in length.

Decryption is again simply taking the cube root.

[3]

(iii) If it is known the message is one of ten, it would be possible to encrypt all possible messages with the recipient's public key and compare to what was actually sent.

[2]

⑤

3 (a) A good message digest function is such that ^{it is known that the} if a message digest of m_1 is $f(m_1)$ then it should be computationally infeasible to find an m_2 ~~whose~~ and its message digest function $f(m_2)$. In addition it should be computationally infeasible to find two messages m_1 and m_2 such that $f(m_1) = f(m_2)$.

The first property prevents the prediction of a particular message digest from knowledge of the message digests of other messages. The second property prevents replacement of a message with another with the same message digest.

[5]

(b) The SHA-1 message digest is a 160 bit quantity (5 32-bit words). The message is processed in 512-bit blocks after padding is added. The MD is initialised to a fixed value and then each stage of the MD computation takes the current value of the MD and digests it with the previous 160 bit output. The final result is the MD for the entire message.

[5]

(c) ⁷

A MAC may be formed by concatenating a secret key K with the message and forming the message digest $MD(K \| m)$.

If the secret is at the beginning of the message it would be possible to add a further 512 bits to the message and compute the new MAC. This may be prevented by including the secret at the end of the message before application of the hash function.

[5]

(d) The following attacks could be made:

- parts of the message could be interchanged (e.g. whole 160-bit chunks or individual A_i) without affecting the MD
- it would be simple to alter parts of the message to give a desired ~~value~~ message ~~without~~ and adjust other parts without affecting the MD.

[8]

(e) From the Birthday paradox it would require only 2^{16} searches to find an m_1 and m_2 such that $f(m_1) = f(m_2)$

[2]

8

(a) Kerberos provides security facilities for authorised machines and users to access other authorised machines in a secure manner over an insecure network such as the Internet.

At login a machine A requests a session key from the Key Distribution Centre (KDC). The session key has a limited lifetime and is used in the next transaction (the request for a ticket) as a key to encrypt the timestamp. [4]

(b) A ticket granting ticket, is issued by the KDC to an authorised user when the identity of that user has been established. The TGT contains the session key, the user's ID and an expiration time all of which will be used by the KDC in issuing a ticket.

A ticket requested by A for remote login to B is supplied by the KDC encrypted under B's master key. Thus when received by B, it is able to use the contents of the ticket which are the ID of A and a new session key for use in communications between A and B.

9) The KDC master key is used to encrypt the principals' master keys held in the KDC database. It is also used to encrypt the information in a TGT so that when returned to the KDC only the KDC may decipher its contents. [6]

(c) The following are examples where V5 offers increased functionality over V4

- Names V5 allows X.500 in addition to DNS
- Delegation of Rights V5 supports forwardable and proxiable tickets (not allowed in V4)
- Ticket lifetimes - virtually unlimited in V5 but ≤ 21 hours in V4
- Renewable and postdated tickets are supported in V5 but not in V4
- V5 supports many cryptographic algorithms

It is necessary to store more key versions in V5 because of renewable and postdated tickets.

[10]

②

(d) The encrypted timestamp is an authenticator designed to prevent replay. B must reply with timestamp + 1 to provide its own authenticator under the same key K_{AB} .

[5]

①

5

- (a) (i) Non-repudiation. For a message sent by A to B, B is able to prove (e.g. in a court of law) that A actually sent the message. That is, B can prove it ~~did not~~ could not have made up the message itself.
- (ii) The Proof of submission is verification ~~provided by the sender~~ provided to the sender that the message was submitted to the electronic mail system.
- (iii) The purpose of message flow confidentiality is to prevent an observer from determining that ~~A and B~~ two parties are in communication.
- (iv) Message sequence integrity proves to the recipient that a sequence of messages have been received in the correct order and with no additions or deletions.

[4]

(b) PEM uses a rigid public key infrastructure for distributing and certifying keys. This is based on a hierarchical naming system with additional policy

⑫ Constraints. The benefits are that in a strict security CA chain certificates can be trusted absolutely. The disadvantage is that no crosslinks are allowed by policy constraints which makes PEM PKI unwieldy.

S/MIME uses a more flexible PKI in that cross-links are allowed yet maintains the trust and confidence in organisational hierarchy.

PGP's PKI is based on anarchy. The benefit to the user is that he can devise his own path in a certificate chain. The disadvantage is that he may not be able to trust the results.

[7]

(c) ~~Perfect~~ A system of offering perfect forward secrecy allows an attacker to record communications between two endpoints and yet be unable to decipher the record even if all long-term secrets of the endpoints have been captured.

One way of providing Perfect Forward Secrecy is to conduct a Diffie Hellman key exchange to generate a session key

(13) and then to 'forget' the Diffie Hellman secret numbers and the resultant key at the end of the session.

[7]

(d) 'Denial of Service / Logging' protection is designed to prevent a server being inundated by authorisation requests from (probably) forged IP addresses. Such an attack would be designed to see bring the server out of service.

One method of protection is to send a challenge to the IP address such as a number which is a function of the address and which must be returned before anything else is done. A forged IP address is unlikely to receive the challenge.

[7]

(14)

6

(a) IPsec is at layer 3 whilst SSL is forms an interface to TCP at layer 4.

The advantage of IPsec is that applications and their APIs do not require change to work with IPsec. IPsec also protects against interfering with the packet stream (e.g. by deleting or adding packets). The disadvantage of IPsec is that it changes the main operating system of the machine (below layer 4) although it may be implemented as an outboard device.

By contrast SSL requires no change to the operating system but does require a change in the API used by applications.

This allows new security features to be added without changing the operating system.

The disadvantage is that packet level attacks may not be 'seen' by SSL but may bring down the TCP session. [6]

(15)

(b) AH is Authentication Header and is an implementation of IPsec which allows for integrity protection. ESP is Encapsulating Security Payload and is an alternative implementation of IPsec providing integrity protection and/or encryption. Although both offer integrity protection AH protects the IP Header whilst ESP protects only beyond the ESP Header. Additionally, AH allows intermediate network devices to look at layer 4 ports. However, these are small benefits when set against the complexity of supporting two separate implementations.

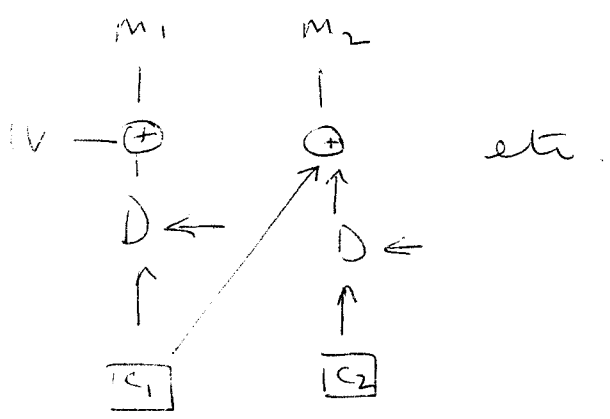
[6]

(c)

(1) The IV ensures that if the same message (or just the first part of the message) is encrypted with the same key the ciphertext will be different.

16 (c) (ii)

The decryption through CBC is as follows



If m_2 is known and is required to be changed to a particular value then it is only necessary to alter C_1 such that the m_2 becomes the desired value.

[6]

(d) (i) Integrity may be protected by adding a known pattern to the plaintext message before deciphering. After deciphering the known pattern is checked

(ii) First encrypt with one key and take the last block of ciphertext, 'the CBC residue' as a MAC and append this to the message. Then run CBC again with a new key for encryption.

Method (ii) is used to protect against any attack and is the preferred method.

(17)

Method (1) will offer protection against ~~whole~~ general manipulation of the ~~message~~ ciphertext but not against a targeted attack on certain message blocks.

[7]