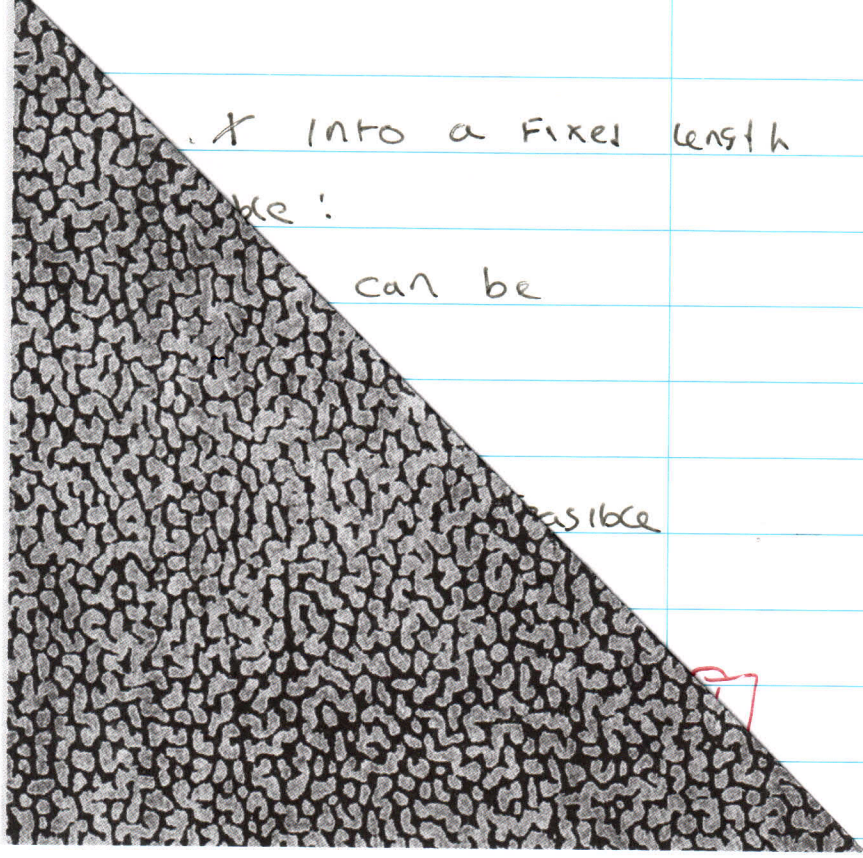
285

Coláiste na hOllscoile
Corcaigh
University College Cork



Uimhir Scrúdaithe
Examination Number 9 1 7 1 6 3

Module Code CS4614

Paper No. _____

Mír
Section _____

Do na Scrúdaitheóirí amháin **For Examiner's use only**	
1	27
2	24
3	25
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
Iomlán **Total**	76

Calculator, Please state:	No. of Books submitted
Name Casio	2
Model Fx911es	

Note: **If there are different sections on this paper, a separate Answer Book MUST be used for each section.**

Q1 A hash function maps x into a fixed length $h(x)$

A It should be unpredictable:

That is given $h(x)$ nothing can be known about x

It should be "irreversible":

Given $h(x)$ it should not be feasible to compute x

It should be free from collisions:

Given two values X, Y if $x \neq y$ then the probability of $h(x) = h(y)$ should be low

6

Q1 B

Duplicate Passwords can be seen in the File.

Given two users with two identical passwords P, P' where P = P'

It can be seen that the two users have the same password, IF one user had a password hint and another didnt that hint could be used to solve the other users password

Triple DES is reversible and an attacker need only compromise master key K_A to retrieve all passwords

ECB was used which allows patterns in the passwords to be seen.

Only applicable for larger passwords

No IV was used!

6

Q1

c They should have been stored using a secure one way hash Function with a salt value The salt value defends against a precomputation dictionary attack as every bit of salt doubles the size of dictoary the attacker needs to generate

Example: 1 bit salt (not large enough)

User = A, Password = XYZ, salt = 0 (randomly chosen)

Store

A, 0, h(XYZ^0) in File

where h is a secure one way hash such as SHA256

Attacker assuming XYZ was in dictionary would have had to precompute both h(XYZ^0) and h(XYZ^1)

As you add salt the amount of Precomputation grows exponentially

6

Q1 D

```
File encryptedFile = new File(filename);
// CipherOutputStream os = cipher.getOutputStream(
                                encryptedFile);
File PlaintextFile = new File(PtFilename);
os.write(PlaintextFile);
```

IV Pseudo code

PF = Open the File handler to the Plain File

EF = Open the File handler to the new encrypted F

3 Real For every block in PF:

WRITE BLOCK TO EF usings ciphered out stream

Q1

E The use of DES

Des is no longer considered a secure encryption algorithm. Its 56 bit keys are susceptable to a bruteforce attack. A more secure algorithm such as AES should be used

The use of ECB blocks

ECB should not be used as it allows patterns in the Plaintext to appear in the ciphertext. It is also vulnerable to cut and paste attacks where blocks. From the Encrypted File can be moved around without breaking the decryption

The use of Random() does not give us secure random numbers. Its output is Predictable

6

The lack of an IV

27

Q2
A

Assuming Eve shares a key K_{ET} Trent she can do the Followins

MSG α 1 $E \rightarrow T$ E, B, N_E

MSG α 2 $T \rightarrow E$ $\{N_E, B, K_{EB}, \{K_{EB}\}_{KBT}\}_{KET}$

Eve receiving the previous message From bob uses K_{ET} to retrieve K_{EB} and $\{K_{EB}\}_{KBT}$ For later use

IF eve wants to masquerade as Alice she can send;

MSG α 3 $A[E] \rightarrow B$ $A, \{K_{EB}\}_{KBT}$

Bob will receive the Message see its From alice and extract K_{EB} using his K_{BT}

10 Bob cannot tell K_{EB} is actually From Eve as It is Just a key

Bob will now use KEB thinking he is communicating with Alice

Eve could replace A with any other user in MSGα3 and Bob will believe he is talking to that user

Q2

B. MSG 1 A→T A, B, N_A

MSG 2 T→A $\{N_A, B, K_{AB}, \{K_{AB}, A, B, Timestamp\}_{KBT}\}_{KAT}$

MSG 3 A→B $A, \{K_{AB}, A, B, Timestamp\}_{KBT}, \{A, NA\}_{KAB}$

MSG 4 B→A $\{A, NA+1\}_{KAB}, \{B, NB\}_{KAB}$

MSG 5 A→B $\{B, NB+1\}_{KAB}$

It provides key revocation with the timestamp. Session keys will expire (Parties wont accept a key thats stale and be forced rekey after a predetermined amount of time

10

It provides mutual auth using the nonces A and B are included in the nonce so that Alices nonce looks different than bobs nonce

Q2

C She can do this by changing the protocol to be similar to kerberos

Alice requests a session key from trent using KAT. For example alice could do

Msg 1 A → T A, N_A, validity

Msg 2 T → A $\{N_A, K_{sessionAT}, validity\}_{KAT}$

Alices workstation uses KAT to retrieve $K_{sessionAT}$ from Msg 2

Trent will now send messages to A using $K_{sessionAT}$ instead of KAT and trent will discard it after the validity so it is no longer used

When Alice wants to message bob She would send the message as before to trent:

A → T A, B, N_A

Trent would then respond with the message as normal (as message 2) but use $K_{sessionAT}$ instead of KAT

4

24

285

Coláiste na hOllscoile
Corcaigh
University College Cork

3 KA

digitally signed by ICT's Private Key

Uimhir Scrúdaithe
Examination Number 9 1 7 1 6 3

Module Code CS4614

Paper No. _____

Mír
Section _____

Do na Scrúdaitheoirí amháin **For Examiner's use only**	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
Iomlán **Total**	

Calculator, Please state:	No. of Books submitted
Name Casio	2
Model FX911ES	

Note: If there are different sections on this paper, a separate Answer Book MUST be used for each section.

Q3

A

Msg 1	B → A	Hello
Msg 2	A → B	Cert A
Msg 3	B → A	$\{B, K_{AB}, N_B\}_{K_A}$
Msg 4	A → B	$\{N_B + 1\}_{K_{AB}}$

Where Cert A contains $\{K_A, A, \text{validity}\}_{SK_T}$ (digitally signed by ↓ KT's Private key)

The Browser knows K_T, our CA and has his public key installed using Cert A It can validate the public key in the cert belongs to A and has not expired it will use this K_A in Msg 3

10

The web server to get Cert A would need to securely sent it K_A and our CA K_T would need to validate K_A is owned by A

Q3

Bi B could be using a poor random number generator to generate K_{AB} or even reuse K_{AB}. The Browser b could be poorly designed or a bug intentionally/accidentally left in it (see heartbleed)

Q3

Bii Msg 1 $B \rightarrow A$ $\{ B, g, n, (g^x \bmod n), N_B \}_{K_A}$

Msg 2 $A \rightarrow B$ $g^y \bmod n$

After message 1 A, the server has g, n and $y^x \bmod n$

A generates a random prime Y (the one it sends in Msg 2)

A computes key k as $\underbrace{(g^x \bmod n)}_{\text{From M1}}^{Y} \bmod n$

After message 2 B, the browser has g, n (It generated those) and $g^y \bmod n$

B computes key k as $(g^y \bmod n)^x \bmod n$

10

Key k that the browser has computed is the same as key k that the server has computed. these are equivalent to key K_{AB}

Q3
C

Using this CA the agency could perform a Man In the middle Attack on the users browsing

The agency would either poison the DNS or trick the user by some other means (installing a proxy) to visit their webserver instead of the legitimate one

The millitary CA will issue certificates to that fake webserver and the certificate would bind that server to (for example Amazon.com)
When the user visits the site they would be presented with a certificate which they would trust and believe they are talking to the real Amazon.com
This certificate would be trusted as the Phone trusts certs signed by that agencys CA

5

25