Maynooth University
National University of Ireland Maynooth

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

JANUARY 2020 EXAMINATION

CS416

Cryptography

Dr. C. Hayes, Dr. J. Timoney, Dr. T. Dowling

Time allowed: 2 hours

Answer at least three questions
Your mark will be based on your best ***three*** answers

All questions carry equal marks

[25 marks]

1 (a) Consider the following ciphertext fragment taken from a larger document [12 marks]

5 10 15 20 25 30 35 40
TKABO KDCRW UIVCT DQKHE GTKAA OVESJ CWVHO F

Given that this cipher is a superencipherment, use the codebook on the data sheet and the information below to recover the plaintext. Show your reasoning clearly.

Most Frequent codes in the ciphertext	**Most popular words in previous decrypted documents from this source**
TKA	**THE**
VCT	**TO**
DCR	**WAS**
CWV	**NOT**
ESJ	**BUT**

(b) A software system for a trading house uses RSA encryption to encrypt requests. In order to introduce some randomness the designers select a QRn to encrypt for each Buy request and a QNRn to encrypt for each Sell request. From the information below recover the original plaintext requests. Explain your reasoning. [10 marks]

n	**100 160 063**
C1	**2**
C2	**3**
C3	**5**

(c) Consider the permutation cipher of length three ciphertext below [3 marks]

5 10 15 20 25
ASEEN YUGOC IHHEP TORET G

Recover the plaintext.

[25 Marks]

2 (a) Consider the following Java code fragment [15 marks]

```
1. String userInput=wtf:isthis:SHA3
2. Key = generatePBEKey(userInput);
.....
3. public static SecretKey generatePBEKey(String userInput){
4.String pwd;
5. String salt;
6. String hashAlgorithm;
7. String [] userSplit = userInput.split(":",3);
8. pwd=userSplit[0];
9. salt=userSplit[1];
10. hashAlgorithm=userSplit[2];
11. int pwdIterations = 65536;
12. int keySize = 128;
13. String keyAlgorithm = "AES";
14. String secretKeyFactoryAlgorithm =
"PBKDF2WithHmac"+hashAlgorithm;
15. byte[] saltBytes = salt.getBytes("UTF-8");
16. SecretKeyFactory skf =
SecretKeyFactory.getInstance(secretKeyFactoryAlgorithm);
17. PBEKeySpec spec = new PBEKeySpec(pwd.toCharArray(),
saltBytes, pwdIterations, keySize);
18. SecretKey = skf.generateSecret(spec);
19. SecretKeySpec key = new
SecretKeySpec(secretKey.getEncoded(), keyAlgorithm);
20. return key;}
```

i) Explain in detail how the **userInput** variable is used in this code fragment.

ii) Line 14 contains the term **PBKDF2WithHmac**. Explain what is meant by this term?

iii) Give a brief description, using an appropriate diagram, of the hash algorithm used in this code.

iv) Describe exactly what is returned from the method SecretKey generatePBEKey.

PLEASE SEE NEXT PAGE FOR THE REST OF QUESTION 2

2 (b) Show how a MAC function can be software engineered to produce a symmetric stream cipher. Use appropriate diagrams and notation to illustrate encryption and decryption. [6 marks]

(c) Show, using an appropriate diagram, how **DESEDE** works. Demonstrate how **DESEDE** is backward compatible with **DES**. What is the size of the keyspace of **DESEDE**? [4 marks]

[25 marks]

3 (a) "Rabin encryption is a special case of RSA encryption". Discuss and justify your arguments. [4 marks]

(b) Given the RSA public key below find the corresponding private key, d. [6 marks]

e	**57307091**
n	**105083507**

(c) The idea of a group generator is an important concept in Cryptography. [7 marks]

i) Is 6 a generator of Z_{101}^*? Justify your answer.

ii) Given that

$$5292^x \bmod 10501 = 6832$$

find x. Justify your answer.

Given your answer above what two recommendations would you make to software engineers generating parameters for the El-Gamal public key encryption system?

(d) Four El-Gamal signatures generated by the same private key x are presented below. Given that the public key (p, g, y) is (11807, 101, 2379) find the private key x used for signing. Show your calculations clearly. How would you prevent such an attack from occurring? [8 marks]

Signature	**H(m)**	**a**	**b**
1	**1001**	**6745**	**4143**
2	**1111**	**7974**	**4001**
3	**1122**	**6745**	**3392**
4	**1011**	**3789**	**4019**
Inverses			
$751^{-1} \bmod 11806 = 3317$		**$6745^{-1} \bmod 11806 = 1311$**	

[25 marks]

4 (a) A bad guy has asked you to design a ransomware system to encrypt files on a disc. She has asked the system should only require one key to be delivered to the victim to recover their files. Explain, using appropriate diagrams and cryptographic concepts, how to design this system. You should show in detail the encryption process and how the delivery of one key recovers everything. [9 marks]

(b) Explain using appropriate notation and diagrams how two users, Aimee and Bronwyn, can securely decide on a common key over an insecure channel. What problem is the security of this system based on? Given the parameters [10 marks]

(g, p) =(5999,10501)

and the Secrets

(Aimee_Secret, Bronwyn_Secret) =(3,16)

calculate the values that both users broadcast and the unique key that both users construct

(c) Software Engineering and security have an up and down relationship. Give two specific examples from the course where good Software Engineering practice can enhance security and two examples where good Software Engineering practice can reduce security. [6 marks]

DATA SHEET

Super Encipherment CodeBook

ASJ	BUT		QPI	THE
BKH	CONFUSE		TVA	WORK
CJK	ATTACK		VYQ	TO
CKX	GET		WEO	HARD
DTP	SOLVE		XAG	WITHOUT
DYR	WAS		XAN	GUESS
IUV	KEY		XUF	IMPOSSIBLE
LTZ	BY		YWT	SYSTEM
MMT	OUT		YWV	NOT
NBO	THIS		YXX	SOME
NGN	EXTRA		ZIQ	IS
QOP	TRICKY			

Base 26 encoding

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

V	W	X	Y	Z
21	22	23	24	25

ASCII characters

NUL	SOH	STX	ETX	EOT	ENQ	ACK	A-Z	a-z
0	1	2	3	4	5	6	65-90	97-122

LFSR recurrence relation

$$z_{m+i} = \sum_{j=1}^{m} c_j z_{i+j-1} \bmod 2 \quad \text{for all } i \geq 1$$

DSA Signature Scheme

r	$(g^k \bmod p) \bmod q$
s	$(h+xr)k^{-1} \bmod q$
e1	$hs^{-1} \bmod q$
e2	$rs^{-1} \bmod q$
ver	$(g^{e1} y^{e2} \bmod p) \bmod q$

El-Gamal Signature Scheme

a	$\mathbf{g^k \bmod p}$
b	$\mathbf{(h-xa)k^{-1} \bmod (p-1)}$

Chaum-van Antwerpen Signature Scheme

Public Key	**(p, α, β)**
Private Key	**a**
Signature y of message x	**$y=x^a \bmod p$**
Verification given (e_1, e_2, d)	**Valid iff $d=x^{e1} \alpha^{e2} \bmod p$**
Disavowal Consistency check given $(e_1, e_2, f_1, f_2, D, d)$	$\left(d\alpha^{-e_2}\right)^{f_1} \equiv \left(D\alpha^{-f_2}\right)^{e_1} \bmod p$

CRT

$$\text{Given } x = a_1 \bmod m_1$$
$$x = a_2 \bmod m_2$$
$$x = a_3 \bmod m_3$$
$$\ldots\ldots\ldots\ldots$$
$$x = a_n \bmod m_n$$
$$\text{Then } x = \sum_{i=1}^{n} a_i M_i N_i \bmod (M)$$
$$\text{where } M = m_1 m_2 \ldots\ldots m_n$$
$$M_i = \frac{M}{m_i}$$
$$\text{and } N_i = M_i^{-1} \bmod m_i$$

Garner's Formula

$$\text{For } x < pq$$
$$\text{If } x \equiv a(\bmod p) \text{ and } x \equiv b(\bmod q)$$
$$T = p^{-1} \bmod (q)$$
$$u = (b-a)T \bmod q$$
$$x = a + up$$

Jacobi Symbol Rules

If n is a positive odd integer and m1= m2 mod n $\left(\frac{m1}{n}\right)=\left(\frac{m2}{n}\right)$

If n is a positive odd integer then $\left(\frac{2}{n}\right)=\begin{cases}1 & \text{if } n\equiv\pm1 \bmod 8\\ -1 & \text{if } n\equiv\pm3 \bmod 8\end{cases}$

If n is a positive odd integer $\left(\frac{m1.m2}{n}\right)=\left(\frac{m1}{n}\right)\left(\frac{m2}{n}\right)$

If m and n are positive odd integers $\left(\frac{m}{n}\right)=\begin{cases}-\left(\frac{n}{m}\right) & \text{if } m\equiv n\equiv 3 \bmod 4\\ \left(\frac{n}{m}\right) & \text{otherwise}\end{cases}$