This is a collection of past exam questions related to the syllabus of the CST Part II course *Cryptography*, set (mostly) by Markus Kuhn in the previous CST Part IB and Part II courses *Introduction to Security*, *Security I*, *Security II*, and *Cryptography* since 2002.

Note that following a rearrangement of the security courses, non-cryptography material has now moved to the Part IB *Cybersecurity* syllabus. Questions that refer entirely to non-cryptography material are therefore omitted here, and (where practical) a note points out which part of a past question is not relevant to the current *Cryptography* syllabus.

3 Cryptography (mgk25)

LegacyGates Ltd has for decades sold garage-door openers that implement a simple challenge–response authentication protocol using their time-honoured 42-bit blockcipher E:

$$\begin{array}{rcl} \text{Door } D\text{:} & & \text{Key fob } F\text{:} \\ \text{pick } C \in_R \{0,1\}^{42} & \xrightarrow{\quad C \quad} & \\ & & R := E_K(C) \in \{0,1\}^{42} \\ \text{open if } R = E_K(C) & \xleftarrow{\quad R \quad} & \end{array}$$

$K \in \{0,1\}^{42}$ is a private key shared between door D and radio key fob F.

E and E^{-1} have become publicly known and burglars have started to eavesdrop some challenge–response pairs (C_1, R_1), (C_2, R_2), (C_3, R_3) and brute-force search for a K that satisfies $R_i = E_K(C_i)$. Renting servers that can try 2^{42} encryptions per day, or even store 2^{42} pairs of blocks in a lookup table, is now quite affordable.

Product teams have been asked to upgrade the protocol to 84-bit security. For "commercial reasons" the company wants to continue to use E, but with a pair of 42-bit keys (K_1, K_2). Five teams each came up with a different proposal. Your task is to evaluate if they succeeded.

For each of the following modifications of the protocol, estimate the number of test encryptions needed, and the amount of storage required, to find a pair (K_1, K_2) that will open the door, and give the corresponding search algorithm.

(*a*) $R := E_{E_{K_2}(K_1)}(C)$ [3 marks]

(*b*) $R := E_{K_1}(C) \oplus K_2$ [4 marks]

(*c*) $R := E_{K_2}(E_{K_1}(C))$ [4 marks]

(*d*) $R := E_{K_2}(E_{K_2}(E_{K_1}(C)))$ [4 marks]

(*e*) $R := E_{K_1}(E_{K_2}(E_{K_1}(C)))$ [4 marks]

Regarding proposal (*e*):

(*f*) What practical advantage would using $R := E_{K_1}(E^{-1}_{K_2}(E_{K_1}(C)))$ instead have? [1 mark]

4 Cryptography (mgk25)

(*a*) List *six* properties that an algebraic group should have to be usable for Diffie–Hellman key exchanges. [6 marks]

(*b*) Let $T : A^8 \rightarrow A^4$ be a new collision-resistant compression function approved for use in Tripos papers, where $A = \{\texttt{a}, \dots, \texttt{z}, \texttt{0}, \dots, \texttt{9}, \texttt{=}, \texttt{\&}\}$ is the "base38" alphabet used.

(*i*) Assuming a Tripos student with pocket calculator can evaluate T once per minute, and assuming all students have a brain with unlimited memory and instantaneous recall time, how many hours will it roughly take until at least half of all students can be expected to each have independently found a collision $T(x) = T(y)$ with $x \neq y$? [2 marks]

(*ii*) Use T to define a collision-resistant hash function $H : A^* \rightarrow A^4$, such that the security proof for the Merkle–Damgård construction can be applied. Describe your padding scheme and list the input blocks fed into T when you evaluate H("`love&peace`"). [6 marks]

(*iii*) Consider an ATM that receives from a bank computer authorization responses of the form (M, C), such as

$$M = \text{“}\texttt{txn=491\&pincheck=0\&limit=0}\text{”}, \quad C = H(K \| M)$$

where $K \in A^8$ is the private key shared between the bank and the ATM, and H is as in Part (*b*)(*ii*).

After recalculating and checking C, the ATM splits M into fields separated by "`&`", and then executes any variable assignments it encounters in such fields from left to right, ignoring fields that do not form an assignment. The above M confirms that the PIN provided for transaction 491 was incorrect and that the cardholder is therefore authorized to receive up to £0 in cash.

Mallory has intercepted the line between the ATM and the bank computer and can read (M, C) and replace it with a modified message (M', C'). She would like to withdraw cash without knowing the PIN. Show how she can form a message M' that ends in "`&pincheck=1&limit=1000`" and how she can calculate for that M' a matching tag $C' = H(K \| M')$ without knowing K. [6 marks]

3 Cryptography (mgk25)

(*a*) *YottaVPN*, your employer's main network-encryption product, generates a master key $K \in_{\mathsf{R}} \{0,1\}^{128}$ and an initial seed $R_0 \in_{\mathsf{R}} \{0,1\}^{80}$ randomly once, when the product is installed. It then uses

$$\text{Algorithm (A):} \qquad R_i = \mathsf{Enc}_K(R_{i-1}) \qquad\qquad \text{for } i > 0$$

to generate a stream $R_1, R_2, \ldots$ of session keys for encrypting individual network connections. That algorithm then runs continuously throughout the lifetime of the product. Your colleague suggests to replace (A) with

$$\text{Algorithm (B):} \qquad R_i = \mathsf{Enc}_K(R_{i-1}) \oplus R_{i-1} \quad \text{for } i > 0$$

because they feel that would be more secure. [Enc is a government-approved blockcipher with 80-bit blocksize and $\oplus$ is bit-wise exclusive-or.]

(*i*) For each of algorithm (A) and (B), averaged over all (K, R_0), what is the expected number of different session keys $|\{R_1, R_2, \ldots\}|$ that they will be able to generate from one (K, R_0)? State your assumptions. [5 marks]

(*ii*) What is the smallest number of different values $|\{R_1, R_2, \ldots\}|$ that could be generated by (A) and (B) from any fixed pair (K, R_0)? [2 marks]

(*iii*) Suggest another deterministic key-derivation algorithm (C), using the same blockcipher, 80-bit state and fixed parameters (K, R_0), that maximises $|\{R_1, R_2, \ldots\}|$. [2 marks]

(*iv*) Years later, a worried user discovers that, due to an operator error, the state (K, R_{65535}) of their *YottaVPN* installation was accidentally committed to a publicly accessible Git repository. Compare which other values R_i were compromised by this leak, if either algorithm (A), (B), or (C) had been used. [6 marks]

(*v*) Name a security benefit that could be claimed for algorithm (B) compared to (A). [1 mark]

(*b*) Your colleagues designed a scheme that encrypts messages $M_i \in \{0,1\}^\ell$ with one-time pads $R_i \in_{\mathsf{R}} \{0,1\}^\ell$ into ciphertexts $C_i = M_i \oplus R_i$. But to help estimate the frequency of transmission errors when transferring the R_i, they decided to occasionally replace the last random bit of any R_i with a "parity" bit, with a probability of 0.01. As a result, the probability of any R_i containing an even number of one bits is 0.505. Does this encryption scheme offer *indistinguishability in the presence of an eavesdropper*? Explain your answer. [4 marks]

4 Cryptography (mgk25)

(*a*) Consider a cyclic group $(\mathbb{G}, \bullet)$ of order q with generator g.

Briefly explain the difference between the Computational Diffie–Hellman problem and the Decision Diffie–Hellman problem for $\mathbb{G}$, and state how if one of these problems is hard for $\mathbb{G}$, what this implies for the other. [6 marks]

(*b*) While decompiling the executable of an ECDSA implementation with unknown domain parameters, you encounter a prime-number constant of the form

```
0x ffffffff ffffffff fffffffe 0626bd0c 2f33945b 7d67dbcb
```

Based on the structure of its hexadecimal representation, what rôle could this number play? Explain your answer based on how elliptic-curve groups used in cryptography can be constructed. [6 marks]

(*c*) A certification authority C would like to issue certificates that bind a user A's public key PK_A to not just that user's name, but to 10 different personal attribute values $A_0, \ldots, A_9$, e.g. forename, surname, year of birth, birthday, gender, country, postcode, street address, email, portrait photo. User A can then use such a certificate to register with a range of different online services. However, not all attributes are required, or even appropriate, to be revealed to each service: some may only need the email address, whereas others need perhaps only forename, year of birth, gender, and the photo.

User A should, therefore, be able to choose, which subset $S \subset \mathbb{Z}_{10}$ of these 10 attributes they want to reveal each time they present their certificate to a service. One solution would be that C signs for each user 2^{10} different certificates, each including a different subset of attributes. But that would be rather inefficient.

Propose a certificate format, where C generates just one digital signature for each user A, but A then can modify their certificate to remove any subset of the ten attribute values, such that the recipient still can be sure the received attribute values are authentic, while not being able to infer the value of the removed attributes (except with negligible probability in polynomial time). Explain in detail what A receives from the certification authority, and what A provides to a service that only needs a certificate covering attribute subset S. [8 marks]

3 Cryptography (mgk25)

Your colleagues need a pseudo-random permutation $P_K : \mathbb{Z}_{10^6} \leftrightarrow \mathbb{Z}_{10^6}$, over the integers in the range 0 to 999 999, where K is a 128-bit key. The standard library of their development environment offers them only a 128-bit pseudo-random permutation, in form of the blockcipher AES-128.

(*a*) Recalling that $2^{20} = 1.048576 \times 10^6$, they first decide that implementing a 20-bit pseudo-random permutation $T_K : \{0,1\}^{20} \leftrightarrow \{0,1\}^{20}$ might get them closer to a solution. How could they implement T_K using the available AES_K function? [4 marks]

(*b*) One of your colleagues then proposes to use the function

$$P'_K(m) := \langle T_K(\langle m\rangle_{20})\rangle^{-1} \bmod 10^6$$

as a "good enough" approximation of what is required.

Notation: $\langle\cdot\rangle_n : \mathbb{Z}_{2^n} \to \{0,1\}^n$ encodes non-negative integers as n-bit bitstrings and $\langle\cdot\rangle^{-1} : \{0,1\}^* \to \mathbb{N}$ does the opposite, i.e. $\langle\langle i\rangle_n\rangle^{-1} = i$ for all $0 \le i < 2^n$.

Propose a distinguisher D that can distinguish P'_K from a random permutation $R : \mathbb{Z}_{10^6} \leftrightarrow \mathbb{Z}_{10^6}$ using not more than 5000 oracle queries, and show that it achieves $\left|\mathbb{P}(D^{P'_K(\cdot)} = 1) - \mathbb{P}(D^{R(\cdot)} = 1)\right| > \frac{1}{2}$ averaged over all K. [6 marks]

(*c*) Another colleague then proposes the following algorithm:

```
function $P_K(m)$:
  $c := T_K(\langle m\rangle_{20})$
  $m := \langle c\rangle^{-1}$
  while $m \ge 10^6$:
    $c := T_K(c)$
    $m := \langle c\rangle^{-1}$
  return $m$
```

Show that this is in fact a permutation by

(*i*) explaining why this algorithm always terminates; [1 mark]

(*ii*) providing an implementation of the inverse $P_K^{-1}(m)$. [3 marks]

(*d*) What side-channel risk could the algorithm for $P_K(m)$ from part (*c*) pose, and what can an observer learn from it? [2 marks]

(*e*) Propose an alternative algorithm that reduces the risk that an observer can learn anything from this type of side channel to a negligible probability. [4 marks]

4 Cryptography (mgk25)

A building access-control smartcard uses the following authentication protocol. Let G be a generator for an elliptic-curve based cyclic group $(E(\mathbb{Z}_p, a, b), +)$ of order q. The card stores in its non-volatile memory a secret key $u \in \mathbb{Z}_q^*$ and a public card identifier $U := uG$. The door does not know u. Curve parameters were chosen such that determining u from curve point U is computationally infeasible.

When the user holds the contactless card in front of the door reader, the card picks a number $c \in \mathbb{Z}_q^*$ and the door picks a number $d \in \mathbb{Z}_q^*$, both uniformly at random. The card calculates the coordinates of elliptic-curve point $C := cG$.

They then exchange the following three messages:

card		**door**
read u, U from memory		
pick $c \in \mathbb{Z}_q^*$, $C := cG$	① $\xrightarrow{(U, C)}$	
	② $\xleftarrow{d}$	pick $d \in \mathbb{Z}_q^*$
$r := (c + du) \bmod q$	③ $\xrightarrow{r}$	

(*a*) What checks should the door perform on the received values U, C, r to verify that the card identified by U really is in possession of u? [4 marks]

(*b*) How many bits will be required to encode the values exchanged in these three messages in order to achieve a security level similar to the use of a 128-bit key in a symmetric MAC? [4 marks]

(*c*) Your colleague is concerned that the calculation of $C := cG$ in the card slows down the authentication process too much, and therefore proposes to postpone transmission of C to the third message, i.e. to change the three protocol messages from previously (U, C), d, r to now U, d, (C, r). Would this affect security? [5 marks]

(*d*) Due to supply-chain issues, the hardware manufacturer no longer can make door readers that send data to the card. Modify the original protocol such that only the card sends data to the door. The card maintains a counter m for how often it has been used, and the door remembers the highest value of m it has previously seen and will only open again when presented with a new value m higher than any seen before. Instead of receiving d in message ②, let the card calculate d in a way such that the card provides a digital signature of m, and sends (d, m) to the card. Keep message ③ the same. [5 marks]

(*e*) Which value appearing in the original protocol no longer has to be transmitted by the unidirectional variant from Part (*d*), and why? [2 marks]

3 Cryptography (mk428)

(*a*) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme that offers CCA security. Explain the concept of *forward secrecy*, why it might be useful, and why Π does not offer it. [3 marks]

(*b*) Explain how the Diffie–Hellman key exchange works, and the assumptions under which it is secure. [3 marks]

(*c*) You and your colleague are asked to design a payments system based on an authenticated symmetric encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$, a digital signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, a Diffie–Hellman group with generator g, and a key derivation function KDF. The requirements are as follows:

- Let B be a bank, and let Alice (A) be a customer of B. Say A has a digital token T (which we take to be an arbitrary bit string) that is worth money. A can deposit that money in her account by securely sending T to B.

- You may assume that the bank knows the public keys of all of its customers, and that each customer knows the public key of the bank.

- As the token T is sent over the network, it must be kept confidential from active attackers. Moreover, the protocol must provide forward secrecy.

Let $(PK_A, SK_A) \leftarrow \mathsf{Gen}$ be Alice's signature keypair, and $(PK_B, SK_B) \leftarrow \mathsf{Gen}$ be the bank's keypair. Your colleague proposes using the following scheme:

> $B \to A: \ (g^x, \mathsf{Sign}_{SK_B}(g^x))$
>
> A receives (g^x, S) and checks whether $\mathsf{Vrfy}_{PK_B}(g^x, S) = 1$.
> If this succeeds, A calculates $K = \mathsf{KDF}((g^x)^y)$ and sends:
>
> $A \to B: \ (g^y, \mathsf{Sign}_{SK_A}(g^y), A, \mathsf{Enc}_K(T))$
>
> B receives (g^y, S, N, C) where N is a customer name, looks up N's public key PK_N, and checks that $\mathsf{Vrfy}_{PK_N}(g^y, S) = 1$; if successful, B decrypts $\mathsf{Dec}_{\mathsf{KDF}(g^{xy})}(C) = T$ and credits it to the account belonging to N.

Let Mallory (M) be an active adversary who is also a customer of the bank. Show that your colleague's scheme is not secure: when Alice wants to deposit a token T in her account, M can cause his account to be credited instead. [7 marks]

(*d*) Suggest an alternative protocol that meets the requirements in part (*c*) while avoiding the problems in your colleague's scheme, and briefly justify your design. [7 marks]

4 Cryptography (mk428)

(*a*) You have intercepted a ciphertext c from the communications of an international criminal gang, and you need to decrypt it. By decompiling the gang's messaging app you learnt that c is encrypted with AES-128 in CBC mode; the plaintext is padded to a multiple of 128 bits by setting the last plaintext byte to be the padding length in bits (encoded as an 8-bit binary number), and setting the remaining bits between the end of the message and the final length byte to zero.

Moreover, you know that the gang operates a server that is reachable over the Internet; this server internally decrypts any ciphertext you send it, and always replies with "ok" (regardless of whether the decrypted data makes sense or not). You cannot break into the server, but you notice one detail: if the decrypted message has correctly formatted padding, the reply is slightly slower than if it has incorrect padding. Presumably this is because the server spends some time storing correctly formatted messages, while malformed messages are quickly discarded without being stored.

Show that, by repeatedly sending messages to the server, you can recover the entire plaintext from c. Explain your technique in detail. [8 marks]

(*b*) The gang figures out that you are decrypting their messages. They decide to continue using AES-128-CBC, but in order to prevent the attack from part (*a*), they add a check to their encryption scheme so that the server rejects any message where the ciphertext has been manipulated. Explain how to securely compute such a check using the SHA-256 hash function. [3 marks]

(*c*) As part of the new check from part (*b*), the server uses the following pseudocode:

```
// tagInMessage and correctTag are byte arrays of equal length
function checkIsMessageOk(tagInMessage, correctTag) {
    for (i = 0 to correctTag.lengthInBytes - 1) {
        if (tagInMessage[i] != correctTag[i]) {
            send "rejected" reply
            return
        }
    }
    send "ok" reply
}
```

Explain the problem with this code, and show how this problem may once again allow you to recover the entire plaintext of an encrypted message. [3 marks]

(*d*) Prove that if a hash function $H(x)$ is collision resistant, then $H(H(x))$ is collision resistant as well. [6 marks]

5 Cryptography (mgk25)

(*a*) Consider the following two alternative definitions of a MAC function, which receives as input an $(n \cdot L)$-bit long message of the form $M = M_1 \| M_2 \| \ldots \| M_L$ with $M_i \in \{0,1\}^n$ and a private key $K \in \{0,1\}^n$ picked uniformly at random, returning a tag $T \in \{0,1\}^n$. Show how neither definition provides the security property of *existential unforgeability*.

(*i*) Let F be an n-bit to n-bit pseudo-random function. Return the message tag $T = F_K(M_1) \oplus F_K(M_2) \oplus \cdots \oplus F_K(M_L)$. [4 marks]

(*ii*) Let F be a $(2n)$-bit to n-bit pseudo-random function. Return the message tag $T = F_K(\langle 1 \rangle \| M_1) \oplus F_K(\langle 2 \rangle \| M_2) \oplus \cdots \oplus F_K(\langle L \rangle \| M_L)$. [6 marks]

[*Notation:* $\|$ = concatenation of bit strings, $\oplus$ = bit-wise XOR, $\langle i \rangle$ = n-bit binary representation of non-negative integer i.]

(*b*) Your colleague proposes to construct an authenticated encryption scheme that encrypts a plain-text message M by first calculating the message authentication code $\mathrm{CMAC}_K(M) = T$, and then forms the ciphertext by encrypting $M \| T$ using CFB mode with initial vector $IV = E_K(T)$, using the same key and blockcipher E_K. Does this construction offer CCA security? Why or why not? [5 marks]

(*c*) Given a block cipher E_K with n-bit block size, where $n \geq 64$ is a power of two, how can you use E_K to construct a strong pseudo-random permutation for $\frac{n}{2}$-bit blocks? [5 marks]

6 Cryptography (mgk25)

(*a*) *CrashHash* is a cryptographic hash function invented by your colleague this morning. It zero-pads input X, splits it into n 256-bit blocks $x_1\|x_2\|\ldots\|x_n = X\|0^{(-|X|) \bmod 256}$ and then appends a length-indicator block $x_{n+1} = \langle|X|\rangle$, as in the Merkle–Damgård construction. It then iterates a 512-bit to 256-bit compression function of the form $C(K, M) = E_K(M)$, where $E_K(M)$ is a blockcipher E applied with 256-bit key K to 256-bit message block M, as

$$\begin{aligned} z_1 &= C(\langle 0\rangle, x_1) \\ z_i &= C(z_{i-1}, x_i) \qquad (1 < i \leq n+1) \end{aligned}$$

The value $H(X) = z_{n+1}$ is the hash value returned. Show that *CrashHash* is not collision resistant, even if E is replaced with an *ideal cipher.* [6 marks]

(*b*) (*i*) How can one modify an implementation of the DES encryption function to obtain the decryption function? [4 marks]

(*ii*) Name two other features of DES that made it well suited for hardware implementation. [2 marks]

(*c*) Your colleague has generated a set of $m = 200\,000$ RSA key pairs that include a modulus $n_i = p_i q_i$ where p_i and q_i are 1536-bit prime numbers (for $1 \leq i \leq m$). The corresponding p_i and q_i values were discarded immediately after key generation and are no longer available.

Due to a bug in your colleague's key-generation software, two types of fault have appeared in a random subset of the issued key pairs:

(*i*) For some key pairs i we have $p_i = q_i$.

(*ii*) For some key pairs i there exists another key pair j in that set with $p_i = p_j$ and $i \neq j$.

Suggest practical tests that can identify all public keys affected by either of these problems and state how often the algorithms involved have to be executed for this task. [4 marks]

(*d*) Calculate 7^{2000} mod 100 by hand. [4 marks]

5 Cryptography (mgk25)

(*a*) (*i*) One way to use a secure hash function H to form a message-authentication code is the construct $\mathsf{Mac}_K(M) = H(K \| M)$. What problem with that approach does the HMAC construct solve? [4 marks]

(*ii*) Why does the HMAC construct pad the key? [2 marks]

(*b*) Your opponent has started using *HomeBrew*, a new block cipher $C = E_K(M)$ that they invented last week. It uses a 96-bit key $K = K_1 \| \ldots \| K_{12}$, where each of the 12 bytes K_i $(1 \leq i \leq 12)$ is used as an 8-bit subkey in one of the 12 rounds that apply a keyed permutation f:

```
$R_0 := M$
for $i := 1$ to 12
    $R_i := f_{K_i}(R_{i-1})$
$C := R_{12}$
```

Describe an attack to find K for this type of block cipher that is practical for an adversary with a computer fast enough to execute such a block cipher around 2^{50} times and that can store and lookup around 2^{50} keys and messages. [6 marks]

(*c*) Your colleague has proposed the following digital signature algorithm. Let $(\mathbb{G}, q, g)$ be system-wide choices of a cyclic group $\mathbb{G}$ of prime order q with generator g such that the discrete logarithm problem in $\mathbb{G}$ is computationally infeasible. Further let $H : \{0,1\}^* \to \mathbb{Z}_q^*$ be a collision-resistant hash function. Pick a secret key $x \in \mathbb{Z}_q$ uniformly at random and let (y, r) with $y := g^x \in \mathbb{G}$ and $r := H(g^{H(x)})$ be the corresponding public key.

Then use as the signature of message $m \in \{0,1\}^*$ the value $s \in \mathbb{Z}_q^*$ found by solving

$$H(x) \cdot s \equiv x \cdot r + H(m) \pmod{q}$$

for $s = [H(x)]^{-1} \cdot [x \cdot r + H(m)]$. (Here a^{-1} denotes the multiplicative inverse of finite-field element $a \in \mathbb{Z}_q^*$. Your colleague considers $\mathbb{P}(s = 0)$ negligible.)

The recipient, given $(\mathbb{G}, q, g, H), (y, r), (m, s)$ verifies that signature by checking the equation

$$H\left(y^{r \cdot s^{-1}} g^{H(m) \cdot s^{-1}}\right) = r$$

Show that this signature scheme does not provide existential unforgability. [8 marks]

6 Cryptography (mgk25)

(*a*) Consider a message-authentication code Mac expected to provide *existential unforgeability under adaptive chosen-message attack.*

(*i*) What requirement does existential unforgeability impose on any padding function applied to the message by Mac and why? [4 marks]

(*ii*) What is an example of a padding function that satisfies that requirement? [2 marks]

(*b*) While reviewing the *MacGyver* burglar alarm system, you notice that a sensor S uses the following stream authentication protocol to report its status to the controller C once every second over a data wire:

$$\begin{array}{lllll}
C & \rightarrow & S: & R & \text{with } R \in_{\mathsf{R}} \{0,1\}^{128} \\
S & \rightarrow & C: & (M_1, T_1) & \text{with } T_1 = \mathsf{trunc}_{32}(\mathsf{Mac}_K(M_1, R)) \\
S & \rightarrow & C: & (M_2, T_2) & \text{with } T_2 = \mathsf{trunc}_{32}(\mathsf{Mac}_K(M_2, T_1)) \\
 & \vdots & & & \\
S & \rightarrow & C: & (M_i, T_i) & \text{with } T_i = \mathsf{trunc}_{32}(\mathsf{Mac}_K(M_i, T_{i-1})) \\
 & \vdots & & &
\end{array}$$

The controller C picks a new 128-bit random value R when the system is powered up. Each message (M_i, T_i) is sent i seconds after that. The messages M_i are normally all identical, of the form $M = 0$ meaning “no burglary has happened in the last second”. Mac is a 128-bit message-authentication code function, using a private key K shared between S and C. Because of the very limited data rate available on the alarm-wire interface, the output of Mac is truncated to the first 32 bits.

(*i*) How can an attacker, who has been observing this communication since power up, eventually predict future tags T_i for the constant message $M_i = M$? [4 marks]

(*ii*) How long will it take, on average, after powerup until the attacker can start sending simulated sensor messages? [4 marks]

(*iii*) What security implication does the predictability of message-authentication codes from a sensor have for a burglar alarm system? [2 marks]

(*iv*) How can you improve the protocol to practically eliminate the risk of that attack, without increasing the number of bits transmitted over the wire? [4 marks]

5 Cryptography (mgk25)

(*a*) The *Tripos Encryption Standard (TES)* is a block cipher optimized for use on UGPs ("undergraduate processors"). It operates on 4-bit blocks, written as hexadecimal digits (e.g., $\mathtt{a} \oplus \mathtt{9} = \mathtt{3}$). For one particular key K, it implements the following permutation:

m	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$E_K(m)$	1	b	5	c	7	e	2	a	4	9	f	d	0	3	6	8

Using this key K, decrypt the following three ciphertexts according to the indicated modes of operation. [*Note:* the XOR table at the bottom of this page may be of use.]

(*i*) ECB: `188b06` [2 marks]

(*ii*) CBC: `301b2` [3 marks]

(*iii*) CFB: `10f6d` [3 marks]

(*b*) State four advantages that counter mode has over either CBC or CFB mode. [4 marks]

(*c*) Using the same K as in Part (*a*):

(*i*) Show that the CBC-MAC tag for message `1234` is `d`. [3 marks]

(*ii*) Demonstrate that CBC-MAC with a given K is not collision resistant, by showing how to find another message, of the form 1x04, that results in the same CBC-MAC message tag (without iterating over different candidates for 4-bit block x). [5 marks]

UGP XOR accelerator:

$\oplus$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	1	0	3	2	5	4	7	6	9	8	b	a	d	c	f	e
2	2	3	0	1	6	7	4	5	a	b	8	9	e	f	c	d
3	3	2	1	0	7	6	5	4	b	a	9	8	f	e	d	c
4	4	5	6	7	0	1	2	3	c	d	e	f	8	9	a	b
5	5	4	7	6	1	0	3	2	d	c	f	e	9	8	b	a
6	6	7	4	5	2	3	0	1	e	f	c	d	a	b	8	9
7	7	6	5	4	3	2	1	0	f	e	d	c	b	a	9	8
8	8	9	a	b	c	d	e	f	0	1	2	3	4	5	6	7
9	9	8	b	a	d	c	f	e	1	0	3	2	5	4	7	6
a	a	b	8	9	e	f	c	d	2	3	0	1	6	7	4	5
b	b	a	9	8	f	e	d	c	3	2	1	0	7	6	5	4
c	c	d	e	f	8	9	a	b	4	5	6	7	0	1	2	3
d	d	c	f	e	9	8	b	a	5	4	7	6	1	0	3	2
e	e	f	c	d	a	b	8	9	6	7	4	5	2	3	0	1
f	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	0

6 Cryptography (mgk25)

(*a*) (*i*) Choose and briefly describe one major application of elliptic-curve group operations in cryptography. [4 marks]

(*ii*) What other group operation was previously (and still is) widely used for the same purpose? [2 marks]

(*iii*) What is a major advantage of elliptic curve group operations over the group operation you named in Part (*a*)(*ii*)? [4 marks]

(*b*) In the Galois field GF(2^8) modulo $x^8 + x^4 + x^3 + x^2 + 1$, calculate

(*i*) the sum 0011 1001 plus 0110 1100; [2 marks]

(*ii*) the product 0100 1011 times 0000 1001. [4 marks]

(*c*) In Lamport's one-time password scheme, the user is given a list of passwords $R_n, \ldots, R_0$ generated using the following algorithm:

```
$R_0 \leftarrow$ random
for $i := 1$ to $n$
    $R_i := h(R_{i-1})$
```

(*i*) State two properties required of function h. [2 marks]

(*ii*) Complete the password verification algorithm implemented in the server by filling in the ellipses (...) below:

```
$Q := \ldots$
while true
    $P :=$ read password
    if $\ldots$
        $\ldots$
        grant access
    else
        deny access
```

[2 marks]

12 Security II (MGK)

(*a*) Name *three* different families of algebraic groups that are commonly used in cryptographic applications of the Diffie–Hellman problem, where *any* group element (other than the neutral element) can be used as a generator. Briefly outline some of their main attributes, such as the set of elements and the group operator. [9 marks]

(*b*) You are preparing to participate in a password-cracking competition. During the competition, you will be given the 128-bit hash-function output MD5(p). You have to find p, an 8-character password, each character having been chosen uniformly at random from a known alphabet of 64 ASCII characters.

In the weeks preparing for the competition, you have access to a small cluster of GPU graphics cards that can evaluate MD5 10^9 times per second.

During the competition, you have only access to a laptop computer that can evaluate MD5 10^6 times per second.

Without any pre-computation, how long would it take to evaluate MD5 for all possible passwords p in a brute-force attack

(*i*) on the laptop? [2 marks]

(*ii*) on the GPU cluster? [2 marks]

You decide to use the GPU cluster to pre-compute a *rainbow table* for this challenge.

(*iii*) What functions other than MD5 will the GPU cluster have to evaluate as often as MD5 when building the rainbow table? [3 marks]

(*iv*) Your laptop has enough RAM for storing the rainbow table as a hash table of 2^{32} key-value pairs (x, y) with $x, y \in \{0, 1\}^{128}$. If you execute MD5 2^{50} times while generating your rainbow table, how long will your laptop need (worst case) to find a password p stored in it, given its MD5 hash value MD5(p)? Assume that the runtime is entirely dominated by the MD5 evaluations. [4 marks]

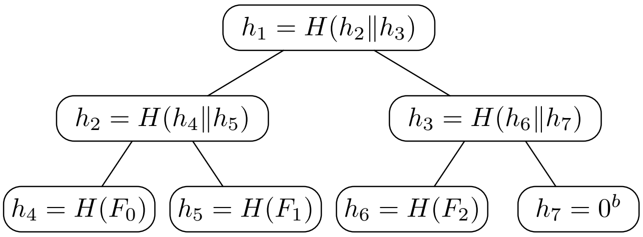
11 Security II (MGK)

(*a*) An RSA encryption routine calculates the value $m^e \bmod n$ using a square-and-multiply algorithm. During the execution of that algorithm, you can briefly hear a buzzing sound (through radio-frequency interference) on an AM radio receiver located near the computer. You record that sound, and discover that it is actually the following sequence of two different sounds A and B: $BABAABABAAB$. What is the value of e? [6 marks]

(*b*) MHash implements a hash function over file sequences $F_0, F_1, \ldots, F_{n-1} \in \{0,1\}^*$ with $n > 0$, using a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^b$:

```
MHash($n, F_0, F_1, \ldots, F_{n-1}$):
  $d := \lceil \log_2 n \rceil$
  for $i := 0$ to $n - 1$
    $h_{2^d+i} := H(F_i)$
  for $i := n$ to $2^d - 1$
    $h_{2^d+i} := 0^b$
  for $i := 2^d - 1$ downto $1$
    $h_i := H(h_{2i} \| h_{2i+1})$
  return $h_1$
```

Example calculation for $n = 3$:



(*i*) Show that MHash is not collision resistant if n is not fixed, by constructing two different sequences of files that result in the same output h_1. [8 marks]

(*ii*) Suggest an improvement to MHash to make it collision resistant. [6 marks]

8 Security I (MGK)

(*a*) *NybbleCrypt* is a block cipher optimized for use in exam questions. It has a block size of 4 bits and a key length of 64 bits. Each block can be written as a single hexadecimal digit, for example $\texttt{5} \oplus \texttt{9} = \texttt{c}$.

(*i*) The *NybbleCrypt* encryption function for a particular key K is given in the following table:

m	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$E_K(m)$	c	8	2	7	d	0	6	1	a	e	f	4	b	9	5	3

Decrypt the following messages, which were encrypted using E_K under the following modes of operation, respectively:

(A) ECB mode: `c994f88` [2 marks]

(B) CBC mode: `b144f` [3 marks]

(C) OFB mode: `eae26` [3 marks]

(*ii*) Calculate the CBC-MAC of the following message, using the same key K as in part (*a*)(*i*) above: `face` [2 marks]

(*iii*) *NybblePay* point-of-sale card terminals send 4-digit customer PINs to the bank's transaction-processing centre for verification. The bank's reply to the terminal consists of a 7-digit message in the following format:

(A) 4-digit PIN $m_1m_2m_3m_4$

(B) 2-digit result code m_5m_6: `10` if the PIN was correct, `e1` if not

(C) check digit $m_7 = m_1 \oplus \cdots \oplus m_6$ (the bit-wise XOR of previous digits)

This reply is sent OFB-encrypted using the *NybbleCrypt* blockcipher. You have intercepted such a ciphertext message: `a59defc2`. You are confident that it contains the result code $m_5m_6 =$ `e1` for an incorrect PIN. Without knowing the encryption key K, modify the ciphertext message such that after decryption it shows the result code for a correct PIN, and a matching check digit, while preserving the included PIN. [5 marks]

(*b*) *NybbleShuffle* is a transposition cipher that operates on blocks of 32768 bytes. It splits each such block into 4-bit subblocks, and then rearranges these subblocks in pseudo-random order, under the control of a secret key K, in order to form the 32768-bytes long ciphertext block that it outputs. What is the smallest number of test blocks that you have to feed into an instance of the *NybbleShuffle* cipher in order to unambiguously reconstruct the permutation of subblocks that it applies, and how do you construct these test blocks? [5 marks]

9 Security I (MGK)

(*a*) Let $\mathsf{Enc}_{K_\mathrm{E}}$ be the encryption function of an encryption scheme that provides indistinguishability under chosen plaintext attack (CPA security). Let $\mathsf{Mac}_{K_\mathrm{M}}$ be a message-authentication-code function that provides existential unforgeability. Named below are three techniques for applying these two functions together to a message M. For each of them

- briefly explain how $\mathsf{Enc}_{K_\mathrm{E}}$ and $\mathsf{Mac}_{K_\mathrm{M}}$ are combined, and
- state whether the resulting construct is likely to provide indistinguishability under chosen ciphertext attack (CCA security):

(*i*) encrypt-and-authenticate [2 marks]

(*ii*) authenticate-then-encrypt [2 marks]

(*iii*) encrypt-then-authenticate [2 marks]

(*b*) How can an attacker calling the C function parse_text below cause a buffer overflow? Explain how and why this works. [6 marks]

```
#include <stdlib.h>
#include <string.h>
#define BUFLEN 4096
int check(int n) {
  if (n > BUFLEN) abort();
  return n;
}
void parse_text(char *text, size_t len) {
  char buf[BUFLEN];
  memcpy(buf, text, check(len));
  /* ... */
}
```

(*c*) Many Unix system administrators create a personal group for each of their users with this user as the sole member.

(*i*) What is the purpose of such a group? [2 marks]

(*ii*) Such personal groups typically have the same name and integer identifier as the corresponding user identifier. Is this practice compatible with the Windows NT mechanism for identifying users and groups? [2 marks]

(*d*) Give two examples for resources where an operating system is expected to implement residual information protection and two alternative mechanisms for implementing it. What are their tradeoffs and threat assumptions? [4 marks]

[*Note:* parts (*b*)–(*d*) not in *Cryptography* syllabus]

8 Security I (MGK)

(*a*) Block ciphers usually process 64 or 128-bit blocks at a time. To illustrate how their modes of operation work, we can use instead a pseudo-random permutation that operates on the 26 letters of the English alphabet:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
m	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$E_K(m)$	D	G	W	X	T	E	R	L	Y	Z	O	J	N	S	I	Q	P	C	U	H	B	V	F	A	M	K

As the XOR operation is not defined on the set $\{\mathtt{A}, \ldots, \mathtt{Z}\}$, we replace it here during encryption with modulo-26 addition (e.g., $\mathtt{C} \oplus \mathtt{D} = \mathtt{F}$ and $\mathtt{Y} \oplus \mathtt{C} = \mathtt{A}$).

(*i*) Decrypt the following ciphertexts, which were encrypted using

(A) Electronic codebook mode: `UOMHDJT` [2 marks]

(B) Cipher feedback mode: `RVPHTUH` [4 marks]

(C) Output feedback mode: `LNMSUUY` [4 marks]

(*ii*) Determine the CBC-MAC for the message `TRIPOS`. [4 marks]

(*b*) Consider another small pseudo-random permutation, this time defined over the set of decimal digits $\{\mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots, \mathtt{9}\}$, using modulo-10 addition instead of XOR (e.g., $\mathtt{7} \oplus \mathtt{3} = \mathtt{0}$).

(*i*) You have intercepted the message `100` with appended CBC-MAC block `4`. The message represents an amount of money to be paid to you and can be of variable length. Use this information to generate a message that represents a much larger number, and provide a valid CBC-MAC digit, without knowing the pseudo-random permutation or key that the recipient will use to verify it. [4 marks]

(*ii*) What mistake did the designer of the communication system attacked in part (*b*)(*i*) make (leaving aside the tiny block size), and how can this be fixed? [2 marks]

11 Security II (MGK)

(*a*) Why does the formal security definition for collision-resistant hash functions require a key s and a security parameter n, even though most commonly used standard secure hash functions lack such input parameters? [4 marks]

(*b*) If $h_s : \{0,1\}^* \rightarrow \{0,1\}^{\ell(n)}$ is a collision-resistant hash function, do the following constructions H_s also provide collision-resistant hash functions? Explain your answers. [2 marks each]

(*i*) $H_s(x) = h_s(x) \parallel x$ (i.e. append x)

(*ii*) $H_s(x) = h_s(x) \parallel \mathrm{LSB}(x)$ (i.e. append least significant bit of x)

(*iii*) $H_s(x) = h_s(x \mid 1)$ (bitwise-or, i.e. set least significant bit of x to 1)

(*c*) Use Euler's theorem to calculate $5^{-1} \bmod 8$. [4 marks]

(*d*) The standard Digital Signature Algorithm (DSA) uses a cyclic subgroup $\mathbb{G} \subset \mathbb{Z}_p^*$ of the integers modulo a prime p, with prime order q, where q divides $p - 1$.

(*i*) Give two advantages of using a multiplicative subgroup of prime order, as opposed to just using $\mathbb{Z}_p^*$, in cryptographic schemes based on the Discrete Logarithm problem. [2 marks]

(*ii*) Why is it possible to choose q substantially smaller than p, and what is an advantage of doing so? [4 marks]

8 Security I (MGK)

(*a*) Compare and contrast the security definitions of a *pseudo-random generator* and a *pseudo-random function.* [4 marks]

(*b*) When a Windows NTFS access control entry (ACE) is inherited by a subdirectory, under which circumstances is the “inherit only” flag set or cleared, and why? [4 marks]

(*c*) What is *existential unforgeability* of a message authentication code? [4 marks]

(*d*) Which problem with CBC-MAC is fixed by ECBC-MAC, and how? [4 marks]

(*e*) A C program running on a 32-bit processor contains the following function:

```
void f(int *a, int l) {
  int *b, i;

  b = (int *) malloc(l * sizeof(int));
  if (b == NULL) return;

  for (i = 0; i < l; i++)
     b[i] = a[i];

  [...]
}
```

(*i*) How can a caller cause this function to overwrite unallocated memory? [2 marks]

(*ii*) Modify the function to remove this vulnerability. [2 marks]

[*Note:* parts (*b*) and (*e*) not in Cryptography syllabus]

9 Security I (MGK)

A smartcard application requires a stack (LIFO) for storing data records. This stack can become much larger than the tiny amount of memory available on the card. Fortunately, the card terminal has enough memory, and its API offers a class `ExternalStack` that can be invoked remotely from the card. It offers the usual four methods: a constructor to initialise an empty stack, `push(`R`)` to place a data record R onto the stack, `isempty()` to test whether the stack contains no record, and `pop()` to remove and return the top record from the stack.

The integrity of the stack is crucial for the security of the application. However, the card terminal is not tamper resistant and the adversary may have full control over the `ExternalStack` object. Therefore, you have to implement a `SecureStack` wrapper class that uses `ExternalStack` as an untrusted storage provider while guaranteeing the integrity of any data returned. The trusted on-card memory available to a `SecureStack` object is only 256 bytes.

Consider how to implement on the card a class `SecureStack` that provides the same four methods by appending additional data to records before pushing them onto `ExternalStack`, and verifying any data returned against locally held values. You have a message authentication function `mac(`K`,(`R`,...))` and a cryptographic key/nonce generator function `gen()` available.

(*a*) Why is just appending a message authentication code to each externally stored record not sufficient? [2 marks]

(*b*) Write short pseudo-code for the four methods of `SecureStack` that shows how they call `ExternalStack`, how they update the on-card check data, and under which conditions a data-integrity alarm is raised. [10 marks]

(*c*) Express the internal check value(s) of your implementation after these calls:

```
Record r, r1, r2, r3;
SecureStack s;

s.push(r1);
s.push(r2);
r = s.pop();
s.push(r3);
```

[4 marks]

(*d*) Determine an upper limit for how often the `push` method of your implementation can be called securely. [4 marks]

9 Security II (MGK)

You are working on an encryption device with your new colleague, Mallory Baish, who proposes that you use a pseudo-random generator

$$r_i = h_1(s_i), \qquad s_{i+1} = h_2(s_i)$$

where $s_0 \in G$ is the random initial state and the other $s_i \in G$ are subsequent internal states, all invisible to adversaries. The $h_1, h_2 : G \to G$ are two secure one-way functions.

Adversaries may see any of the past outputs $r_0, \ldots, r_{n-1}$. If they can predict from those, with non-negligible probability, the next value r_n, then the security of your device will be compromised.

(*a*) Give a rough estimate for the probability that an adversary can predict r_n, as a function of n and $|G|$. Explain your answer. [6 marks]

(*b*) Mallory also suggests a specific implementation:

$$\begin{aligned} h_1(x) &= f(u^x \bmod p) & p &= \text{a 2056-bit prime number} \\ h_2(x) &= f(v^x \bmod p) & u, v &= \text{two numbers from } \mathbb{Z}_p^* \\ f(x) &= x \bmod 2^{2048} & G &= \mathbb{Z}_{2^{2048}} \end{aligned}$$

(*i*) The constants p, u and v will be known to the adversary. What conditions should they fulfill so that h_1 and h_2 can reasonably be described as one-way functions, and how would you normally generate suitable numbers u and v? [*Hint:* quadratic residues] [4 marks]

(*ii*) If f were replaced with the identity function, how could an adversary distinguish the r_i emerging from this pseudo-random generator from a sequence of elements of $\mathbb{Z}_p^*$ picked uniformly at random? [4 marks]

(*iii*) After you choose a value for p, Mallory urges you to use two particular values for u and v generated in your absence. You briefly see "$v = u^e \bmod p$" scribbled on a whiteboard. You become suspicious that Mallory is trying to plant a secret backdoor into your pseudo-random generator.

Explain how Mallory could exploit such a backdoor. [6 marks]

9 Security I (MGK)

Block ciphers usually process 64 or 128-bit blocks at a time. To illustrate how their modes of operation work, we can use instead a pseudo-random permutation that operates on the 26 letters of the English alphabet:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
m	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$E_K(m)$	P	K	X	C	Y	W	R	S	E	J	U	D	G	O	Z	A	T	N	M	V	F	H	L	I	B	Q

As the XOR operation is not defined on the set $\{\mathtt{A}, \ldots, \mathtt{Z}\}$, we replace it here during encryption with modulo-26 addition (e.g., $\mathtt{C} \oplus \mathtt{D} = \mathtt{F}$ and $\mathtt{Y} \oplus \mathtt{C} = \mathtt{A}$).

(*a*) Encrypt the plaintext "TRIPOS" using:

(*i*) electronic codebook mode; [2 marks]

(*ii*) cipher-block chaining (using IV $c_0 = \mathtt{K}$); [4 marks]

(*iii*) output feedback mode (using IV $c_0 = \mathtt{K}$). [4 marks]

(*b*) Decrypt the ciphertext "BSMILVO" using cipher-block chaining. What operation should replace XOR? [4 marks]

(*c*) Your opponent is allowed to send you two plaintext messages M_0 and M_1, each n letters long. You now pick a new private key K, resulting in a new pseudo-random permutation $E_K : \{\mathtt{A}, \ldots, \mathtt{Z}\} \leftrightarrow \{\mathtt{A}, \ldots, \mathtt{Z}\}$. You also pick uniformly at random a private bit $b \in \{0, 1\}$ and return a ciphertext $C = c_0 c_1 \ldots c_n$, namely the message M_b encrypted with cipher-block chaining using the fresh E_K. Finally, your opponent has to guess your bit b.

Approximately how large must n be at least for your opponent to have a greater than 75% chance of guessing b correctly? Outline a strategy that your opponent can use to achieve this. [6 marks]

11 Security II (MGK)

(*a*) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme that operates on fixed-length messages from $\mathcal{M} = \{0,1\}^m$. Briefly explain a game that a user $\mathcal{U}$ of Π must be able to win against any polynomial-time adversary $\mathcal{A}$ with probability $\frac{1}{2} - \epsilon$ (where is ϵ is "negligible" with growing key length) for Π to be able to claim to offer "indistinguishable multiple encryptions under chosen-plaintext attack" (CPA security). [8 marks]

(*b*) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme that offers CPA security and operates on fixed-length messages $M \in \mathcal{M} = \{0,1\}^m$ with keys $K \in \mathcal{K} = \{0,1\}^\ell$. We use it to construct a new encryption scheme $\Pi' = (\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$. In which of the following cases is Π' also CPA secure? Explain your answer. [2 marks each]

(*i*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_K(M \oplus 1^m)$

(*ii*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_K(M) || \mathrm{LSB}(M)$

(*iii*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_K(M) || \mathrm{LSB}(K)$

(*iv*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_{0^\ell}(M)$

[*Note:* LSB outputs the least significant bit of its input word, || is concatenation.]

(*c*) While reviewing an implementation of AES-CBC, you discover that it simply uses the last ciphertext block from the previously encrypted message as the IV value C_0 for encrypting the next message. The implementation's author argues that as long as the IV of the very first message was chosen uniformly at random, all resulting subsequent ciphertext blocks will also be distributed uniformly at random, and therefore make good IVs. Why is this construction nevertheless not CPA secure? [4 marks]

8 Security I (MGK)

(*a*) In the Galois field GF(2^8) modulo $x^8 + x^4 + x + 1$, calculate

(*i*) the difference 1100 1010 minus 1001 0011; [2 marks]

(*ii*) the product 0100 1011 times 0000 1001. [6 marks]

(*b*) Briefly explain two advantages that arithmetic in GF(2^{128}) has over arithmetic in $\mathbb{Z}_{2^{128}}$ when designing cryptographic algorithms. [6 marks]

(*c*) Given a block cipher E_K and a corresponding decryption function D_K, provide a formula for the decryption of the following modes of operation and state for each whether the E_K or D_K calculations required during decryption can be executed in parallel: CBC, OFB, CTR. [6 marks]

12 Security II (MGK)

The RSA public-key crypto system performs calculations in the group $\mathbb{Z}_n$, with $n = pq$ being the product of two large prime numbers p and q. The public key consists of the tuple (n, e), with $\gcd(\phi(n), e) = 1$, and the corresponding private key is (n, d). A message $m \in \mathbb{Z}_n$ is encrypted via $c = m^e \bmod n$ and decrypted as $m = c^d \bmod n$.

(*a*) Given p, q, and e, how can you apply the extended Euclidian algorithm to find a suitable d? [6 marks]

(*b*) If we modified RSA to use as the public modulus a prime number instead of a composite of two large prime numbers, that is $n = p$ instead of $n = pq$, would this affect its security, and if so how? [4 marks]

(*c*) In the *UltraSecure* virtual-private network, each router knows of each of its remote communication peers the RSA public key (n, e), which all have $e = 3$ and $2^{1023} \leq n < 2^{1024}$. If router *alice* needs to establish a shared 256-bit AES secret key k with remote router *bob*, it looks up *bob*'s (n, e) and then uses this method:

- *alice* picks $k \in \{0, 1\}^{256}$ by reading 32 bytes from `/dev/random`
- *alice* interprets k as binary integer m with $0 \leq m < 2^{256}$
- *alice* sends $c = m^e \bmod n$ to *bob*
- *bob* decrypts c into m and recovers k (by removing leading zeros)

Then *alice* and *bob* secure the rest of their communication with shared secret k.

(*i*) How could an eavesdropper obtain m from c? [4 marks]

(*ii*) Suggest a better method of using RSA to establish an AES key than the one given above. [6 marks]

8 Security I (MGK)

Briefly explain

(*a*) the function of a *salt value* in a password database [3 marks]

(*b*) *two* examples of covert channels in a file system protocol that is restricted to read-only operations under a mandatory access-control policy [2 marks]

(*c*) *three* types of common software vulnerabilities, with examples [9 marks]

(*d*) *two* problems solved by Cipher Block Chaining [2 marks]

(*e*) under which conditions will user U be able to remove a directory D in Berkeley Unix [4 marks]

[*Note:* part (*b*), (*c*) and (*e*) not in Cryptography syllabus]

9 Security I (MGK)

(*a*) Name and briefly explain *three* increasingly demanding security properties expected from a *secure hash function h*. [3 marks]

(*b*) Explain how a 128-bit secure hash function h can be used to implement a one-time signature scheme, *and* how to verify such a signature. [5 marks]

(*c*) The T1000 encryption module was designed to make *TeleGroove* messages difficult to read for eavesdroppers. Such messages are character sequences of arbitrary length, encoded using a 5-bit alphabet $A = \{a_0, \ldots, a_{31}\}$. The T1000 module reads blocks of up to 1000 characters at a time into its memory $(m_0, m_1, \ldots, m_{999})$ and then outputs them again in a different order $(m_{K(0)}, m_{K(1)}, \ldots, m_{K(999)})$, according to a secret key table $K : \{0, \ldots, 999\} \to \{0, \ldots, 999\}$ that is shared with the respective recipient of each message. It repeats that process until the block ending with the last character has been processed.

(*i*) What constraint on K ensures that no information is lost? [2 marks]

(*ii*) Identify an ambiguity in the above description and propose a flaw hypothesis regarding a related residual-information vulnerability in this device. [4 marks]

(*iii*) What is the smallest number of blocks that a chosen-plaintext attacker has to send through the device to recover K? Give an example of the content of such blocks and explain how to recover K from them. [6 marks]

2011 Paper 4 Question 8

(Computer Science Tripos Part IB)

Security I (MGK)

(*a*) In Windows NTFS, each file can have an associated access control list (ACL). Each entry has a type in $\{allow, deny\} \times \{explicit, inherited\}$.

(*i*) What restriction does the Windows Explorer graphical user interface impose on the order in which these types of access-control entries can appear in an ACL? [4 marks]

(*ii*) Give *one* example of a POSIX file access-control configuration for which an equivalent NTFS ACL violates this GUI restriction. [4 marks]

(*b*) Your colleagues used a pseudo-random function $f : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ in order to construct a permutation $g : \{0,1\}^{192} \rightarrow \{0,1\}^{192}$. The argument and return values of g are split into three 64-bit registers, respectively: $g(X_1, X_2, X_3) = (Y_1, Y_2, Y_3)$. The output of g is calculated as $Y_2 = f(X_1) \oplus X_2 \oplus f(X_3)$, $Y_1 = X_1 \oplus f(Y_2)$, and $Y_3 = X_3 \oplus f(Y_2)$, where $\oplus$ denotes bit-wise exclusive or.

(*i*) Show that g is indeed a permutation. [4 marks]

(*ii*) Show how an attacker who does not know f can efficiently distinguish g from most random permutations, after evaluating g on two different inputs. [4 marks]

(*iii*) After you point out this shortcoming to your colleagues, they propose an improved variant $g'(X_1, X_2, X_3) = (Z_1, Z_2, Z_3)$ that adds another round to g: $Z_1 = Y_1$, $Z_2 = f(Y_1) \oplus Y_2 \oplus f(Y_3)$, and $Z_3 = Y_3$.

Show how this variant still does not fix the problem of efficient distinguishability from most random permutations. [4 marks]

2009 Paper 5 Question 9

(Computer Science Tripos Part IB)

Introduction to Security (MGK)

(*a*) Make the following statements correct by changing one word or number. (Negating the sentence is not sufficient.)

(*i*) The Advanced Encryption Standard defines a 16-round Feistel cipher. [1 mark]

(*ii*) Files encrypted with Cipher Block Chaining start with a zero initial vector. [1 mark]

(*iii*) Each user on a Unix system is identified by a unique prime number. [1 mark]

(*iv*) The "read" bits associated with a Unix directory affect whether the files in its subdirectory "`foo`" can be accessed. [1 mark]

(*v*) The "real user ID" associated with a Unix process determines its access rights. [1 mark]

(*b*) Name *five* examples of actions for which a Unix application will need to be invoked with *root* privileges. [5 marks]

(*c*) Explain the attack on Double DES that motivates the use of Triple DES. [6 marks]

(*d*) Under which conditions is the Vignère cipher unconditionally secure? [4 marks]

2008 Paper 3/10 Question 8

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(*a*) A source of secure, unpredictable random numbers is needed to choose cryptographic keys and nonces.

(*i*) Name *six* sources of entropy that can be found in typical desktop-computer hardware to seed secure random-number generators. [4 marks]

(*ii*) What sources of entropy can a smartcard chip, like the one in your University Card, access for this purpose? [4 marks]

(*b*) As Her Majesty's prime hacker "001", on a mission deep inside an enemy installation, you have gained brief temporary access to a secret chip, which contains a hardware implementation of the DES encryption algorithm, along with a single secret key. You connect the chip to your bullet-proof laptop and quickly manage to encrypt a few thousand 64-bit plaintext blocks of your choice, and record the resulting 64-bit ciphertext blocks. You are unable to directly read out the DES key K used in the chip to perform these encryptions and you will not be able to leave the site without knowing K. But you know that all S-boxes in the last DES round are supplied in this chip via a *separate* power-supply pin. When you create a short-circuit on that pin, the encryption progresses as normal, except that the output of all S-boxes in the last round changes to zero.

(*i*) Explain briefly the role of an S-box and the structure of a single round in DES. [4 marks]

(*ii*) How can you find K, considering that your available time and computing power will not permit you to search through more than 10^9 possible keys? [8 marks]

2007 Paper 3/10 Question 9

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(*a*) You have received a shipment of hardware random-number generators, each of which can output one 128-bit random number every 10 milliseconds. You suspect that one of these modules has been tampered with and that it actually produces only 30-bit random numbers internally, which it then converts via a pseudo-random function into the 128-bit values that it outputs.

(*i*) How does this form of tampering reduce the security of a system that uses a generated 128-bit random number as the secret key of a block cipher used to generate message authentication codes? [2 marks]

(*ii*) Suggest a test that has a more than 0.5 success probability of identifying within half an hour that a module has been tampered with in this way. [6 marks]

(*b*) Explain briefly

(*i*) the encryption and decryption steps of Cipher Feedback Mode; [3 marks]

(*ii*) why some operating systems ask the user to press a special key combination (e.g., Alt-Ctrl-Del) before each password login; [3 marks]

(*iii*) how a secure hash function h can be used to implement a one-time signature scheme; [3 marks]

(*iv*) what happens if the same private key of the scheme from (*iii*) is used *multiple times*, to sign different messages. [3 marks]

2007 Paper 4/11 Question 8

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(*a*) Your colleague wants to use a secure one-way hash function h in order to store h(password) as password-verification information in a user database for which confidentiality might become compromised. For h, she suggests to use an existing CBC-MAC routine based on AES with all bits of the initial vector and the 128-bit AES key set to zero. Is this construct a suitable one-way hash function for this application? Explain why. [8 marks]

(*b*) Explain how and under which circumstances overlong UTF-8 sequences could be used to bypass restrictions regarding which files an HTTP server serves. [8 marks]

(*c*) Name *four* techniques that can be used to make buffer-overflow attacks more difficult. [4 marks]

2006 Paper 4 Question 10 / Paper 11 Question 10

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(*a*) Alice and Bob participate in a public-key infrastructure that enables them to exchange legally binding digital signatures.

(*i*) Name *two* reasons why, for some purposes, Alice might prefer to use a message authentication code, instead of a digital signature, to protect the integrity and authenticity of her messages to Bob. [4 marks]

(*ii*) Outline a protocol for protecting the integrity and authenticity of Alice's messages to Bob that combines the benefits of a public-key infrastructure with those of using a message authentication code. [4 marks]

(*b*) Your colleague proposes a new way for constructing a message authentication code using a block cipher $E : \{0,1\}^{64} \times \{0,1\}^{128} \to \{0,1\}^{128}$. He takes the n-bit input message M, appends $p = 64 \cdot \lceil n/64 \rceil - n$ zero-bits, and splits the result into $k = (n+p)/64$ 64-bit blocks $M_1 || M_2 || \ldots || M_k = M || 0^p$. He then calculates the message authentication code as

$$C_K(M) = E_{M_1}(E_{M_2}(E_{M_3}(\ldots E_{M_k}(K)\ldots)))$$

where K is the 128-bit secret key shared between sender and recipient. Show *two* different ways in which an attacker who observes a pair $(M, C_K(M))$ can, without knowing K, create a new pair $(M', C_K(M'))$ with $M' \neq M$. [6 marks]

(*c*) Show how a 128-bit message authentication code $C_K(M)$ with 64-bit key K can be constructed for an n-bit long message M using

(*i*) a secure hash function $H : \{0,1\}^* \to \{0,1\}^{256}$, such as SHA-256; [2 marks]

(*ii*) a block cipher $E : \{0,1\}^{128} \times \{0,1\}^{256} \to \{0,1\}^{256}$. [4 marks]

2005 Paper 3 Question 9 / Paper 10 Question 11

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(*a*) A and B play a simple game. A chooses a number $R_A \in \mathbb{Z}_3$ and B chooses a number $R_B \in \mathbb{Z}_3$. Then A and B communicate their respective choice to each other *simultaneously*, meaning that the players cannot change their choice after having seen that of the opponent. These rules decide who wins the game:

$$\begin{aligned} R_A &\equiv R_B + 1 \pmod 3 \quad \Rightarrow \quad A \text{ wins} \\ R_B &\equiv R_A + 1 \pmod 3 \quad \Rightarrow \quad B \text{ wins} \end{aligned}$$

In any other case, the result of the game is a draw.

(*i*) What complication arises when this game is played at a distance, for example via email? [2 marks]

(*ii*) Suggest a cryptographic protocol that prevents cheating when this game is played via email. Your solution should not rely on a trusted third party. [6 marks]

(*iii*) What assumptions do you make about the cryptographic functions used in your solution of (*ii*)? [3 marks]

(*iv*) What assumptions do you make about the amount of computing power available to the opponent in your solution of (*ii*)? [3 marks]

(*b*) Outline briefly the purpose of an organisation's security policy and what steps should be considered in its development. [6 marks]

2004 Paper 3 Question 9 / Paper 10 Question 11

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(*a*) Explain briefly mechanisms that software on a desktop computer can use to securely generate secret keys for use in cryptographic protocols. [5 marks]

(*b*) Give two different ways of implementing residual information protection in an operating system and explain the threat addressed by each. [5 marks]

(*c*) Consider the standard POSIX file-system access control mechanism:

(*i*) Under which conditions can files and subdirectories be removed from a parent directory? [2 marks]

(*ii*) Many Unix variants implement an extension known as the "sticky bit". What is its function? [2 marks]

(*iii*) On a POSIX system that lacks support for the "sticky bit", how could you achieve an equivalent effect? [2 marks]

(*d*) VerySafe Ltd offer two vaults with electronic locks. They open only after the correct decimal code has been entered. The VS100 – a low-cost civilian model – expects a 6-digit code. After all six digits have been entered, it will either open or it will signal that the code was wrong and ask for another try. The VS110 – a far more expensive government version – expects a 40-digit code. Users of a beta-test version of the VS110 complained about the difficulty of entering such a long code correctly. The manufacturer therefore made a last-minute modification. After every five digits, the VS110 now either confirms that the code has been entered correctly so far, or it asks for the previous five digits again. Compare the security of the VS100 and VS110. [4 marks]

2003 Paper 3 Question 9 / Paper 10 Question 11

(Computer Science Tripos Part Ib, Part II (General), Diploma)

Introduction to Security (MGK)

(a) Explain the difference between mandatory and discretionary access control. [4 marks]

(b) (i) Explain the purpose and operation of cipher-block chaining (CBC). [4 marks]

(ii) Explain how to decrypt a message in CBC. [4 marks]

(c) To protect her interview partners, a journalist needs to ensure that what she records with her digital camera cannot be viewed by anyone before she returns to her home country. You were asked to design for her a camera that encrypts recordings immediately before they are stored on tape. The question arises, how to handle the encryption key. If it is stored in the camera, it could be extracted if the hardware were confiscated and analysed. A key memorised by the user might be obtained using coercion, so this is not a suitable solution either.

Suggest *two* alternative convenient ways of arranging the encryption inside the camera such that decryption of the tape is only possible on the journalist's home computer. [8 marks]

2002 Paper 3 Question 2 / Paper 10 Question 4

(Computer Science Tripos Part Ib, Part II general, Diploma)

Introduction to Security (MGK)

(a) (i) Explain the collision resistance requirement for the hash function used in a digital signature scheme. [4 marks]

(ii) Show how the DES block cipher can be used to build a 64-bit hash function. Is the result collision resistant? [4 marks]

(b) A sequence of plaintext blocks $P_1, \ldots, P_8$ is encrypted using DES into a sequence of ciphertext blocks. Where an IV is used, it is numbered C_0. Owing to a transmission error, one bit in ciphertext block C_3 changes its value, and as a consequence, the receiver obtains after decryption a corrupted plaintext block sequence $P'_1, \ldots, P'_8$. For the following modes of operation, how many bits do you expect to be wrong in each block P'_i?

(i) cipher block chaining [2 marks]

(ii) 64-bit output feedback [2 marks]

(c) (i) Explain the *Feistel principle* used by block ciphers such as DES and its purpose. [4 marks]

(ii) Using a given pseudo-random function $F : \{0,1\}^{100} \rightarrow \{0,1\}^{100}$, construct a pseudo-random permutation $P : \{0,1\}^{300} \rightarrow \{0,1\}^{300}$ by extending the Feistel principle appropriately. [4 marks]

2002 Paper 7 Question 6

(Computer Science Tripos Part II)

Security (MGK)

(a) One of the algorithms that you implement inside a smartcard requires a stack (LIFO) for storing data records. Assume that each record is a few hundred bytes long. This stack can become very large and the small amount of memory available in the card is not sufficient to hold it. Therefore you decide that the stack object will be implemented via remote method invocation in the card terminal, which has enough memory.

The externally stored stack offers the usual four methods: `init` to initialise an empty stack, `push` to place a data record onto the stack, `isempty` to test whether the stack contains no record, and `pop` to retrieve the top record from the stack.

The integrity of the stack is crucial for the security of the application and the card terminal is not tamper resistant. Consider an algorithm for an integrity-protected stack object that will be implemented on the smartcard and uses an existing secure hash function h available in the card as well as the external stack object.

Explain:

(i) Why is just adding a simple message authentication code to each externally stored record not sufficient here? [2 marks]

(ii) What check data do you attach to externally stacked records, such that the memory required in the smartcard does not grow with the stack size? What check data remains inside the card? Show the resulting internal check values and the records on the external stack (call them $Y_1, Y_2, \ldots$) after you pushed three data records X_1, X_2, X_3 onto the stack. [6 marks]

(iii) Write short pseudo-code for the methods of the protected stack object (`secure_init`, `secure_push`, `secure_pop`, `secure_isempty`) that shows how they update the on-card check data and under which conditions a tampering alarm is raised. [6 marks]

(b) In the same smartcard application, you also need an externally stored integrity-protected queue (FIFO). You decide to protect each externally stored record with a MAC for which a new key will be generated whenever the FIFO is initialised. What check data beyond the MAC key needs to be kept inside the card? What additional check data do you have to add to the records to guarantee the integrity of the FIFO? [6 marks]

2002 Paper 8 Question 6

(Computer Science Tripos Part II)

Security (MGK)

(a) Explain the concept of a *Trusted Computing Base* and outline its meaning in the context of the access control provided by a typical Unix workstation. [5 marks]

(b) An automatic teller machine (ATM) communicates with a central bank computer for PIN verification. A 32-bit CRC code is added to each packet to detect transmission errors and then the link is encrypted using a block cipher in counter mode. Describe an attack that is possible in this setup. [5 marks]

(c) (i) Describe a cryptographic protocol for a prepaid telephone chip card that uses a secure 64-bit hash function H implemented in the card. In this scheme, the public telephone needs to verify not only that the card is one of the genuine cards issued by the phone company, but also that its value counter V has been decremented by the cost C of the phone call. Assume both the card and the phone know in advance a shared secret K. [5 marks]

(ii) Explain the disadvantage of using the same secret key K in all issued phone cards and suggest a way around this. [5 marks]