**UNIVERSITY OF BRISTOL**

**JANUARY 2014 Examination Period**

**FACULTY OF ENGINEERING**

**Examination for the Degree of**
**Bachelor and Master of Engineering and Bachelor and Master of Science**

**COMS30002J**
**CRYPTOGRAPHY A**

**TIME ALLOWED:**
**2 Hours**

# Answers to COMS30002J: CRYPTOGRAPHY A

**Intended Learning Outcomes:**

On successful completion of this unit you will be able to

1. understand the Mathematical underpinnings of cryptography,
2. appreciate and apply appropriate cryptographic proofs of security,
3. understand the design and operation of standard, state-of-the-art symmetric and asymmetric cryptographic schemes,
4. appreciate basic cryptanalytic techniques, and apply this knowledge to problems such as selection of key size.

**Q1**. This question focuses on the Damgård ElGamal encryption scheme, which is a variation on the original ElGamal scheme. The scheme is defined as follows:

**Key generation** Kg generates a cyclic group $\mathsf{G}_q$ of prime order $q$ with generator $g$. A private decryption key sk consists of two elements $x, \omega \in \mathbb{Z}_q$ both sampled independently and uniformly at random. The public key consists of the group description $(\mathsf{G}_q, q, g)$ together with the group elements $h \leftarrow g^{\omega}$ and $y \leftarrow g^{x}$.

**Encryption** Enc takes as input a public key pk and a message $\mathsf{m} \in \mathsf{G}_q$, uniformly at random selects $r$ from $\mathbb{Z}_q$ and computes the ciphertext $c \leftarrow (g^r, h^r, m \cdot y^r)$.

**Decryption** Dec takes as input a private key sk and a ciphertext $c = (c_1, c_2, c_3)$. It checks whether $c_1^{\omega} = c_2$ and if and only if this is the case, it returns $m' \leftarrow c_3 c_1^{-x}$.

(a) Prove the correctness of the scheme.

*[4 marks]*

**Solution:**

**Syllabus:**

- understand the Mathematical underpinnings of cryptography,
- understand the design and operation of standard, state-of-the-art symmetric and asymmetric cryptographic schemes,

*Correctness* requires that for all keys that can be generated, all honestly generated ciphertexts decrypt to the original message. In this case, we need to show that for an honestly generated ciphertext $(c_1, c_2, c_3) \leftarrow (g^r, h^r, m \cdot y^r)$ it holds that $c_1^{\omega} = c_2$ and $m = c_3 c_1^{-x}$. These two follow from

$$c_1^{\omega} = (g^r)^{\omega} = (g^{\omega})^r = h^r = c_2$$

and

$$c_3 c_1^{-x} = m \cdot (g^x)^r (g^r)^{-x} = m g^{xr - rx} = m$$

**Marking:** 2 points for setting up what needs to be shown, 1 point each for the final justifiation (equations)

(b) The scheme is malleable. Explain what malleability means in general, demonstrate the malleability of this scheme, and comment on the implications in terms of security.

*[4 marks]*

**Solution:** A scheme is malleable if an adversary, when given a ciphertext (and in this case public key) but not the underlying plaintext, can modify the ciphertext in such a way that the "change" to the plaintext is predictable.

In this particular case, given $(c_1, c_2, c_3) = (g^r, h^r, m \cdot y^r)$ an adversary can compute $(g \cdot c_1, h \cdot c_2, y \cdot c_3)$ which is a different ciphertext for the same plaintext. (Alternatively, for any $k \in G_q$, $(c_1, c_2, k \cdot c_3)$ is an encryption of $k \cdot m$ .) Malleable schemes cannot be OW-CCA or IND-CCA secure, as the adversary can query the decryption oracle on the mauled ciphertext and learn the challenge plaintext.

**Marking:** 1 point for description of malleability, 2 points for demonstrating malleability of this scheme, 1 point for the implication (explaining how malleability can be used practically, e.g. by increasing some amount to pay, are also valid).

(c) This scheme is derived from ElGamal. Explain how Damgård ElGamal differs from ElGamal.

*[4 marks]*

**Solution:** None of the components involving $h$ is part of standard ElGamal. Thus $\omega$ need never be generated during $\mathsf{Kg}$ and both public and private key are shorter; during encryption the creation of $c_2$ is omitted (it is not part of an ElGamal ciphertext); during decryption there is no check, in ElGamal $m'$ is always returned.

**Marking:** 2 points for stating (implicitly or explicitly) the workings of ElGamal. 2 points for observing the effect of the check.

(d) ElGamal itself can be shown to be IND-CPA secure under the DDH assumption. Give the definition of IND-CPA security.

*[3 marks]*

**Solution:** The standard diagram of IND-CPA security for PKE should be given.

**Marking:** 1 mark for correct challenge creation, 1 mark for correct oracle access 1 mark for correct adversarial input. (a correct IND-CPA diagram for SE yields 2 points; the surplus encryption oracle is not penalized.)

(e) Prove that if ElGamal is IND-CPA secure, then so is Damgård ElGamal.

*[5 marks]*

**Solution:** To show the implication, we show that if Damgård ElGamal is insecure, so is ElGamal. If Damgård ElGamal is not IND-CPA secure, there must be an efficient and successful adversary $A$ against the IND-CPA property of Damgård ElGamal. We use this adversary to create an adversary $B$ against the IND-CPA property of ElGamal as follows.

When $B$ receives an ElGamal public key $(\mathsf{G}_q, q, g, y)$, it samples $\omega \leftarrow \mathbb{Z}_q$ at random and computes $h \leftarrow g^\omega$. It passes $(\mathsf{G}_q, q, g, h, y)$ as Damgård ElGamal public key to

$A$ and lets $A$ run. When $A$ outputs a message pair $(m_0, m_1)$ to be challenged on, $B$ forwards these to its own challenge oracle receiving challenge ciphertext $(c_1, c_3)$. It then computes $c_2 = c_1^{\omega}$ and provides $(c_1, c_2, c_3)$ as the challenge ciphertext to $A$. Once $A$ outputs a guess bit $b'$, $B$ outputs this bit $b'$ as its guess as well.

By inspection, $B$'s advantage is exactly the same as $A$'s advantage and its overhead (beyond running $A$) is minimal. Since we assumed $A$ efficient and successful, we conclude that $B$ is efficient and succesful as well.

**Marking:** 2 marks for setting up the reduction. 3 marks for making the reduction explicit. (4 marks for proving the reverse direction.)

**Q2**. This question focuses on building a message authentication code from an authenticated encryption scheme. To this end, let $(\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be an authenticated encryption scheme and consider the MAC scheme $(\mathsf{Kg}, \mathsf{Tag}, \mathsf{Vrfy})$ where key generation $\mathsf{Kg}$ for the MAC scheme is exactly the same as that of the authenticated encryption scheme, and

**Tagging** $\mathsf{Tag}$ takes as input a key $\mathsf{k}$ and a message $\mathsf{m}$. It computes $\tau \leftarrow \mathsf{Enc}(\mathsf{k}, \mathsf{m})$.

**Verification** $\mathsf{Vrfy}$ takes as input a $\mathsf{k}$, a message $\mathsf{m}$, and a tag $\tau$. It computes $\mathsf{m}' \leftarrow \mathsf{Dec}(\mathsf{k}, \tau)$ and accepts iff $\mathsf{m}' = \mathsf{m}$.

(a) Prove the correctness of the MAC scheme, assuming correctness of the authenticated encryption scheme.

*[2 marks]*

**Solution:** *Correctness* requires that all honestly generated tags are valid: $\mathsf{Vrfy}(\mathsf{k}, \mathsf{m}, \mathsf{Tag}(\mathsf{k}, \mathsf{m}))$ should accept for all $\mathsf{k}$ and $\mathsf{m}$. By substituting the $\mathsf{Tag}$ and $\mathsf{Vrfy}$ algorithms, for correctness we need that $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{k}, \tau)$ where $\tau = \mathsf{Enc}(\mathsf{k}, \mathsf{m})$. This is exactly what correctness of the authenticated encryption scheme gives us.

**Marking:** 1 point for explaining the correctness requirement in general, 1 point for applying it to this particular case.

(b) Comment on the efficiency and tag size of the MAC scheme.

*[2 marks]*

**Solution:**

If the authenticated encryption scheme itself is of the type mac-then-encrypt, the encrypt part of the computation is completely superfluous. However, any mac will have to process the entire message so computation is linear in the message length and the slowdown of the proposed MAC is only by a constant factor. The resulting tag size is linear in the message length, whereas it only needs to be linear in the security parameter (a much more serious shortcoming).

**Marking:** 1 point for computational efficiency 1 point for tag size (need to mention message length is too long).

(c) Katz and Lindell gave three principles of modern cryptology. Name these principles (no explanation needed) and identify each of these principles in the statement:

> The MAC scheme is existentially unforgeable under chosen message attacks (EUF-CMA secure) if the underlying authenticated encryption scheme satifies integrity of ciphertext (INT-CTXT secure).

*[4 marks]*

**Solution:**

1. Definitions: "The MAC scheme is existentially unforgeable under chosen message attacks (EUF-CMA secure)"
2. Assumptions: "if the underlying authenticated encryption scheme satifies integrity of ciphertext (INT-CTXT secure)."
3. Reductions: the statement has the format "A is secure if B is secure", which would be proved by a reduction.

**Marking:** 1 point for naming all three (proofs instead of reductions is acceptable). 1 point for each correct identification. (correctly naming and identifying the paradigms gives 2 points tops.)

(d) Give the definition of EUF-CMA security for a MAC scheme.

*[3 marks]*

**Solution:** The standard figure for EUF-CMA security of a MAC scheme should be given.

**Marking:** 1 point for the oracle access, 1 point for the adversarial I/O, 1 point for the winning condition.

(e) Compare the definition of INT-CTXT security for a symmetric encryption scheme with that of EUF-CMA security for MACs: what are the differences and what are the similarities?

*[5 marks]*

**Solution:** INT-CTXT is very close to the strong EUF-CMA notion in the presence of message recovery. In both games the adversary has access to an oracle that creates

valid tags resp. ciphertexts and the goal is to come up with one more (message–tag pair resp. ciphertext).

The minor difference is that with (standard, not strong) EUF-CMA the forgery needs to be on a fresh message, whereas with INT-CTXT the decryption may result in a message on which a ciphertext was requested (as long as the forgery ciphertext is fresh).

A potentially more pronounced difference is the availability of a verification oracle for an EUF-CMA adversary, which will always output either true or false, versus a decryption for an INT-CTXT adversary, which can return a message or $\perp$ (for ciphertexts deemed invalid). A more accurate decryption analogue of a verification oracle would output $\perp$ for invalid ciphertexts and some other, fixed symbol for valid ones. However, it turns out that for INT-CTXT the decryption oracle is redundant anyway: if an adversary were ever be able to query it with a ciphertext that is both valid (not resulting in $\perp$) and fresh, i.e. wasn't produced by the encryption oracle, then this ciphertext itself would have won the game already!

**Marking:** 1 point for the tagging/encryption oracle, 2 points for the verification oracle discrepancy, 1 point for mapping the goals, 1 point for the subtlety strong EUF-CMA versus normal EUF-CMA.

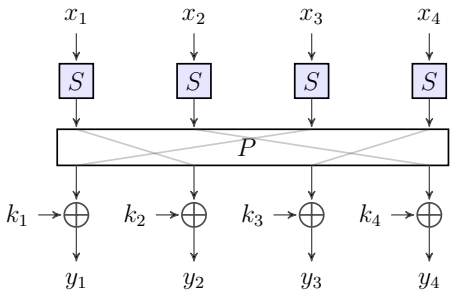
(f) Consider the MAC scheme as described above, but where the verification routine is changed as follows: $\mathsf{Vrfy}'$ takes as input a $\mathsf{k}$, a message $\mathsf{m}$, and a tag $\tau$. It computes $\mathsf{m}' \leftarrow \mathsf{Dec}(\mathsf{k}, \tau)$ and accepts iff $\mathsf{m}' \neq \perp$. How does this modification affect security? Argue your answer.

*[3 marks]*

**Solution:** The scheme is no longer unforgeable. After seeing a single valid tag for an arbitrary message, an adversary can use that tag for any other message of her choosing.

**Marking:** 3 points for a full correct answer, 2 point for a partial answer. 1 point for stating 'it becomes unforgeable'.

**Q3**. This question focuses on substitution-permutation networks. These are commonly used to build blockciphers. A blockcipher typically consists of several rounds of an SP network, with $S$ and $P$ fixed and the various round keys generated from the master key using some key schedule. Below is an illustration of a single round of an SP-network.




(a) What is a blockcipher and what is meant by its block-length? *[2 marks]*

**Solution:** A blockcipher is a family of keyed permutations with efficiently computable inverses (if given the key). The two inputs of a blockcipher are often referred to as key, resp. block and for modern blockciphers both are bitstrings. In that case the block-length refers to the bitlength of the block.

**Marking:** 1 point for blockcipher, 1 point for block-length

(b) Explain Kerckhoffs's principle in terms of a blockcipher built from an SP-network. What are the implications on the block-length and the key-length when a security level of 128-bit is desired (assuming a single round-key uniquely determines the overall blockcipher key)? *[3 marks]*

**Solution:** Kerckhoffs's principle states that the security of a cryptosystem should solely rely on the key and not on the obscurity of the algorithms involved. In this case, security should be maintained even if the adversary knows $P$, $S$, and the key schedule. It also implies that the key space should be sufficiently large. In this case $(k_1, \ldots, k_4)$ should be at least 128 bits, which implies the same for the block length.

**Marking:** 1 point for stating Kerckhoffs's principle (implicitly or explicitly), 1 point for stating what is known/unknown, 1 point for implication on key and block length.

(c) One round (as depicted above) is insufficient as a blockcipher. Give an attack that is strong yet efficient, and describe what kind of attack it is. *[3 marks]*

**Solution:** Given a single known plaintext–ciphertext pair $(x, y)$, compute $k \leftarrow S(P(x)) \oplus y$. This is a key recovery under a known plaintext attack.

(cont.)

**Marking:** 1 point for an attack, 1 point for the description, 1 point for it being efficient KR-KPA.

(d) Why is omitting the S-boxes (from a multi-round SP-network) a bad idea? *[3 marks]*

**Solution:** The resulting blockcipher becomes a linear transformation in its two inputs (key and block). Given a few known plaintext–ciphertext pairs will allow efficient reconstruction of the key (or of an equivalent key).

**Marking:** Points depending on level of detail and correctness of the answer; 2 points for observing collapse to single round; 1 point for mentioning lack of confusion.

**Q4**. This question focuses on a family of RSA-inspired signature schemes, defined as follows:

**Key generation** Kg selects 1023-bit prime numbers $p'$ and $q'$ such that $p \leftarrow 2p' + 1$ and $q \leftarrow 2q' + 1$ are both prime as well. Let $N \leftarrow pq$. Denote with $Q_N$ the group of quadratic residues modulo $N$, that is $x \in Q_N$ iff $x \in \mathbb{Z}_N^*$ and there is some $y \in \mathbb{Z}_n$ such that $x \equiv y^2 \bmod N$. Key generation also selects a function $f$ from the message space into $Q_N$ (how this function is selected is immaterial to this question). The public key consists of $N$ and the function $f$; the private key is $(p', q')$.

**Signing** Sign takes as input a private key and a message $\mathsf{m}$. It computes $y \in Q_N$ and $e$ some 128-bit prime number such that $y^e = f(\mathsf{m}) \bmod N$. The signature is the pair $(y, e)$.

**Verification** Vrfy takes as input a public key $\mathsf{pk} = (N, f)$, a message $\mathsf{m}$, and a purported signature $(y, e)$. It accepts iff $y^e = f(\mathsf{m}) \bmod N$.

(a) Show how to implement the signing algorithm, i.e. describe an efficient algorithm to compute the signature $(y, e)$ using knowledge of the private key. You may assume an efficient routine GenPrime that on input $\ell$ outputs a random prime of length $\ell$. *[4 marks]*

**Solution:**

1. Select $e$ by running $e \leftarrow \mathrm{GenPrime}(128)$.
2. Compute $d_p \leftarrow e^{-1} \bmod p'$ and $d_q \leftarrow e^{-1} \bmod q'$ using the extended GCD.
3. Compute $y_p \leftarrow f(m)^{d_p} \bmod p$ and $y_q \leftarrow f(m)^{d_q} \bmod q$.
4. Find $y \bmod N$ such that $y \equiv y_p \bmod p$ and $y \equiv y_q \bmod q$ using the Chinese Remainder Theorem.

**Marking:** 1 point per step; the direct alternative (based on $\phi(N)$) gets full marks as well.

(b) Argue why your algorithm is efficient. *[3 marks]*

**Solution:**

1. By assumption on GenPrime.
2. Extended GCD's runtime is polynomial in the input lengths.
3. Modular exponentiation can be performed efficiently using e.g. binary exponentiation.
4. The CRT can be efficiently computed by a combination of modular inverses (using XGCD) and modular multiplication.

(cont.)

**Marking:** 1 point per step, with a maximum of 3. Omitting one explanation for something that is efficient might be ok, implicitly assuming something that is inefficient as efficient will result in fewer than 3 marks.

(c) Argue why your algorithm is correct, namely that the signatures that are generated will pass verification. *[3 marks]*

**Solution:** We need to show that for the $y$ we constructed, indeed $y^e \equiv f(m) \bmod N$. This is equivalent to showing that $y^e \equiv f(m) \bmod p$ and $y^e \equiv f(m) \bmod q$ (CRT). Let us concentrate on the modulo $p$ branch, the modulo $q$ branch follows analogously. Since $f(m)$ is a quadratic residue modulo $N$, it has order dividing $p'q'$. As a result, $f(m) \bmod p$ has order dividing $p'$, meaning that $f(m)^{p'} \equiv 1 \bmod p$. As a result

$$y_p^e \equiv (f(m)^{(e^{-1} \bmod p')})^e \bmod p \equiv f(m)^{e^{-1}e \bmod p'} \bmod p \equiv f(m)^1 \bmod p \equiv f(m) \bmod p$$

Since $y \equiv y_p \bmod p$ it follows that $y^e \bmod p \equiv y_p^e \bmod p \equiv f(m) \bmod p$ which concludes the proof.

**Marking:** 1 point for getting the CRT split, 1 point for getting the orders right, 1 point for combining everything neatly.

**END OF PAPER**