UNIVERSITY OF BRISTOL

JANUARY 2017 Examination Period

FACULTY OF ENGINEERING

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

COMS-30002(J)
CRYPTOGRAPHY A

TIME ALLOWED:
2 Hours

Answers to COMS-30002(J): CRYPTOGRAPHY A

Intended Learning Outcomes:

On successful completion of this unit you will be able to

1. explain and apply the principles of modern cryptology in teh context of secure communication
2. explain and demonstrate the functionality and desired security of standard cryptographic schemes used for confidentiality and authenticity.
3. link the design and operation of standard, state-of-the-art symmetric and asymmetric cryptographic schemes to their mathematical underpinnings.
4. use basic cryptanalytic techniques to evaluate the security level of simple cryptographic schemes.

Q1. (a) A friend of yours has stumbled over the concept of the one-time pad. Unfortunately, he did not attend Crypto A, so he needs your help to make sense of the following statements. Moreover, some of the statements are misleading or even incorrect. Comment on the validity of each of the statements below, drawing particular attention to mistakes or misleading claims.

[12 marks]

A. the one time pad is useless because it just xors plaintexts and ciphertexts.

B. the one time pad provides perfect secrecy, and hence the key can be reused securely.

C. the one time pad is not secure because the key needs to be as long as the message that you encrypt.

D. the one time pad is so secure that even an adversary with unlimited computing power cannot break it.

E. the one time pad provides perfect secrecy and hence integrity of ciphertexts.

Solution:

A. Xoring the plaintext and the ciphertext gives the key, which is an operation that doesn't occur during regular use. It renders the OTP insecure against chosen plaintext attacks if the key is re-used; but key re-use is insecure anyway.

B. The OTP is perfectly secure, provided the key is not reused. Key reuse is catastrophic for security.

C. The OTP is secure provided the key is as long as the message; the issue is therefore not a security one, but a key-management one.

D. Perfect security does indeed mean that even an adversary with unlimited computing power who only observes ciphertexts cannot learn anyting about the message. However...

E. The OTP most certainly does not provide integrity of ciphertexts as all ciphertexts will be accepted as valid.

(b) Your friend has moved on to more modern cryptology, but is still struggling. He has realized that key lengths are important when implementing cryptography in real life, but isn't entirely sure about the following four statements. Again, comment on each of the statements below, drawing particular attention to mistakes or misleading ones.

[6 marks]

A. The longer the key the more secure any system will be.

B. Furthermore, it is important to have a good random number generator for key generation.

C. However, in case of public key cryptography, it is o.k. to have short public keys.

D. Even if you loose your secret key, you can still decrypt messages.

Solution:

A. In general for secure schemes yes, to increase the security level one has to increase the key length. However, especially for insecurity schemes, increasing the key length might not increase security. In particular, increasing key length is unlikely to result in a stronger security notion being achieved.

Pitfalls: Referring to the OTP as an example of why this statement is not true, by pointing out that the length of the key in the OTP is fixed to the length of the message, is less convincing. This is an o.k. example to show that the statement does not even make sense for all cryptosystems, but it is not a strong example because the OTP is quite impractical, and so it is an 'academic example' only. A stronger answer would distinguish the two cases (secure vs. insecure scheme) as in the model answer.

B. Definitely true. If keys are predictable as a result of bad randomness generation all security is lost.

C. Not true. After all, if public keys are short, then collisions are likely, meaning that two different people (with possibly different private keys) will share the same public key. As they both have a corresponding private key, the can decrypt each other's traffic (or sign on behalf of each other).

(cont.)

Pitfalls: Whilst for the RSA system it is o.k. to have short *public exponents* (i.e. e can be small), the public key would be correctly regarded as (N, e), so including the modulus. Obviously, if the length of (N, e) was short, then N itself would have to be short as well and thus RSA would become insecure (aside from possibly colliding moduli, it would become easy to factor).

Even in a setting such as ElGamal encryption where two entities can share domain parameters such as the description of the group and some generators g and h, having the same public key but different corresponding private keys, for instance if Anna's public key is $A = g^x h^y$, with private key (x, y), and Ana's public key is $A' = g^{x'} h^{y'}$, then a collision $A = A'$ causes a problem even if $(x, y) \neq (x', y')$: Anna and Ana will still be able to decrypt each other's messages (because sender Bob cannot distinguish between their respective public keys). Thus, if everyone selects from a pool of 'short' public keys, such collisions are more likely to occur, which is bad for security. (For ElGamal there is a secondary problem of representing group elements A and A' compactly for groups where the DLP is supposedly hard.)

Another popular answer was to point out that if the exponent in e in RSA is small, then it could be that the encryption of similarly short messages would be small and so taking the eth root (over the reals) would allow decryption. This again is true (assuming padding does not interfere) and this was marked "correct".

D. No, if you could decrypt without secret key, so could the adversary, implying your scheme is woefully insecure.

Q2. WhatsApp is a messaging service that introduced end-to-end encryption in 2016. There is a WhatsApp Security Whitepaper that explains how media and other attachments are transmitted. There are various steps to ensure that both the sender and receiver share two keys, namely an AES256 key and a HMAC-SHA256 key. With these in place, the main cryptographic step reads:

> The sender encrypts the attachment with the AES256 key in CBC mode with a random IV, then appends a MAC of the ciphertext using HMAC-SHA256.

(a) Unpack the terminology from the quote above by explaining what the various abbreviations (AES256, CBC, IV, MAC, HMAC, SHA256) refer to and what their general purpose is. You do not (yet) have to explain how they work.

[12 marks]

Solution: AES256 stands for the advanced encryption standard with 256 bit keys; it is a blockcipher. CBC stands for ciphertext block chaining; it is a mode of encryption based on a blockcipher. IV is the initial vector; it is used as the first block to add randomization to a mode of operation (such as CBC). MAC is shorthand for message authentication code; it produces tags to provide authentication of its input. HMAC is a specific hash-based MAC, which prescribes how to turn a keyless hash function into a keyed MAC. SHA256 is a standardized hash function, which outputs 256-bit digests of its input.

Marking: 2 points per term

(b) For *one* of the schemes or modes mentioned (AES256, CBC, HMAC, SHA256) explain at a high level how it works.

[3 marks]

Solution: AES256 is an key-alternating cipher built on top of an SP network; For CBC there is the familiar diagram; HMAC has a clean specification in terms of a general hash function; SHA256 is an iterated hash function loosely based on a blockcipher run in Davies-Meyer mode (the latter is not taught in the lectures though).

(c) Assume a receiver already has all the relevant keys. How would they decrypt some received ciphertext? (Note: describe the steps in roughly the same amount of detail as in the quote above.)

[3 marks]

Solution: The receiver parses the ciphertext as an output of the CBC-mode and a tag from HMAC-SHA256. It recomputes the tag based on the CBC-mode output; if the recomputed tag does not match the received tag, reject the ciphertext (output $\perp$), otherwise proceed by decrypting the CBC-output and return the resulting message.

(cont.)

Marking: 1 point for parsing; 1 point for the MAC check with rejection; 1 point for the CBC decryption. It is not necessary to explain CBC decryption in detail.

(d) What security goal(s) do you think WhatsApp is targeting? Describe the goal informally and link it to a formal security notion (you do not need to give the full definition or diagram of the security notion).

[4 marks]

Solution: Authenticated encryption, which itself provides confidentiality, integrity and authenticity. Here confidentiality means that an adversary intercepting ciphertexts cannot learn anything about the message; the most relevant notion here is IND-CPA, or indistinguishability under chosen plaintext attacks. Integrity and authenticity jointly refer to the the assurance that a receiver has that the message or ciphertext really originated with the purported sender (believed to be the only other person in possession of the secret key). The most relevant notion capturing this is INT-CTXT, or integrity of ciphertexts.

Marking: 1 point for each of IND-CPA and INT-CTXT, 1 point for highlighting and explaining confidentiality, 1 point for discussing integrity/authenticity.

Q3. This question addresses RSA-OAEP and a simplification thereof. Let's first consider the original.

(a) Explain the purpose of the original OAEP. In particular, which weaknesses of vanilla or textbook RSA are addressed by using RSA-OAEP instead?

[3 marks]

Solution: Vanilla RSA is deterministic (so not IND-CPA secure) and malleable (so not OW-CCA secure). OAEP addresses both of these shortcomings (achieving IND-CCA security).

Marking: 2 marks for highlighting only one of the shortcomings with correct conclusion

(b) RSA-OAEP can be proven IND-CCA secure under the RSA assumption in the random oracle model. Prove that any IND-CCA secure public key encryption scheme is also OW-CPA secure (you may ignore the random oracle).

[6 marks]

Solution: This is a fairly standard reduction, given a OW-CPA adversary create an IND-CCA adversary. The reduction (IND-CCA adversary) selects two arbitrary but distinct messages m_0 and m_1, output these to its own game. It forwards the public key and challenge ciphertext to the OW-CPA adversary who spits out a purported message m'. If $m' = m_0$ then the reduction outputs 0, otherwise output 1.

Marking: 1 points for setting up the reduction, 1 point for each notion, 1 point each for forwarding public key and ciphertext correctly, 1 point for the logic turning a message into a bit.

Now consider the following simplified version of RSA-OAEP instead. Let H be a public hash function from $\{0,1\}^{1535} \to \{0,1\}^{512}$ and, given an RSA modulus N and some $R \in \{0,1\}^{128}$, define $\mathsf{pad}_R : \{0,1\}^{512} \to \mathbb{Z}_N$ by $\mathsf{pad}_R(\mathsf{m}) = (0^{1407}||R||\mathsf{H}(0^{1407}||R) \oplus \mathsf{m})$ where this 2047-bit string is interpreted as an integer in $\{0, ...2^{2047} - 1\} \subset \mathbb{Z}_N$ (for instance, the bitstring $(0^{2045}||11)$ corresponds to 3).

Key generation Kg selects two random yet distinct 1024-bit prime numbers p and q, both congruent to 2 modulo 3. Let $N \leftarrow p \cdot q$ and $e \leftarrow 3$. Set d to the inverse of e modulo $\phi(N)$. Select a random $R \in \{0,1\}^{128}$ and publish (N, R, e) as the public key pk while keeping (N, R, d) as the private key sk.

Encryption Enc takes as input the public key $\mathsf{pk} = (N, R, e)$ and a message $\mathsf{m} \in \{0,1\}^{512}$. It computes and returns ciphertext $\mathsf{c} \leftarrow \mathsf{pad}_R(\mathsf{m})^e \bmod N$.

(c) Describe how decryption for the RSA version with padding as above works.

[2 marks]

Solution: Compute $x \leftarrow c^d \bmod N$. Parse x as $Z||R||C$ where Z has 1207 bits, R has 128 bits, and C has 512 bits. If $Z = 0^{1407}$ accept the ciphertext and output $C \oplus \mathsf{H}(0^{1407}||R)$ as message. Otherwise ($Z \neq 0^{1407}$) reject the ciphertext.

Marking: 1 point for getting x, 1 point for reconstructing the message. Note that using R from the private key is acceptable as well; recovering $0||m$ is not quite correct but carried full marks nonetheless.

(d) Explain why decryption, as described by you, works. Concentrate on the modular arithmetic involved.

[3 marks]

Solution: This is essentially the standard proof that RSA decryption works.

Marking: Using $a^{\phi(N)} \equiv 1 \pmod{N}$ requires observing that $\gcd(\mathsf{pad}_R(\mathsf{m}), N) = 1$, which is true as $\mathsf{pad}_R(\mathsf{m}) < 2^{512+128} < \min(p, q) \approx 2^{1024}$. Omitting this observation results in a maximum of 2 marks.

(e) Assume H is a secure hash function. Argue about the security of the scheme above. Highlighting multiple strengths and weaknesses may result in higher marks. Where possible, illustrate a weakness by giving an explicit attack.

[6 marks]

Solution: Several observations are possible and not all observations are required for full marks (indeed, some are mutually exclusive!)

1 pt. If invalid ciphertexts get rejected correctly, the scheme is not easily malleable.

1 pt. Key generation is fine (the modulus is sufficiently large for current practice).

2 pts. If decryption never rejects and uses R from the private key, the scheme is malleable (to be demonstrated by an explicit attack).

2 pts. (For a single user) The hash function H is superfluous, as R is fixed as part of the key and H is only ever evaluated on $\mathsf{H}(0^{1407}||R)$.

2 pts. Encryption is deterministic hence no indistinguishability (IND-CPA insecure).

3 pts. The padding and choice of e is such that there is no modular wrap-around, so decryption is possible without needing to know the private key (OW-CPA insecure).

Having only a single round of Feistel is less secure than two rounds (as OAEP), however it is not immediately clear how this concretely affects security (if R were random and e large to ensure modular wrap-around.

Pitfalls:

- Setting the primes p and q has no discernible effect on security
- Multi-user attacks are frustrated by the different Rs
- Protection against malleability does not imply OW-CCA security!

END OF PAPER