Applied Cryptography Final Examination

Spring Semester 2021

(263-4660-00L)

K.G. Paterson

Examination Rules:

(a) **DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO BY AN EXAMINER.**

(b) Closed-book, written exam, 120 minutes total.

(c) If you feel disturbed or have questions about the exam, raise your hand to call an assistant.

(d) You are permitted to leave the exam early, but not within the last 10 minutes of the exam. If you wish to leave, please raise your hand and wait for an assistant.

(e) There are four problems in the exam, worth 25 marks each. You should attempt as many problems as you can, but of course you do not need to complete every part of every problem to obtain the best possible grade.

(f) Read all the problems before beginning. In particular, Problem 4 is different in nature from the first three problems.

(g) Write your name, legi and seat number at the top of each sheet of paper that you use.

(h) Start each problem on a new sheet of paper.

(i) Write as clearly as you can and show all your work. For some questions, you may get partial credit even if the end result is wrong due to a calculation error.

(j) Write with a black or blue pen (no pencil, no green or red ink).

(k) Turn off your mobile phone. No electronic devices are allowed on desks, except for watches (and not smartwatches).

Problem 1 (Block ciphers, CBC mode and padding oracles).

(a) Give the definition of a *block cipher*. Your answer should make reference to the block size n and the key size k. [3 marks]

(b) Explain why block ciphers are usually used in *modes of operation*. [1 mark]

(c) The *Cipher Block Chaining (CBC)* mode of operation takes a sequence of plaintext blocks $P_1, P_2, \ldots, P_t$ and outputs ciphertext blocks $C_0, C_1, C_2, \ldots, C_t$, where C_0 is the initialisation vector and where $C_i = E_K(C_{i-1} \oplus P_i)$ for each $1 \leq i \leq t$. Here $E_K(\cdot)$ denotes the block cipher encryption operation with key K.

i. Write down equations that describe the decryption operation for CBC mode. [2 marks]

ii. Draw a diagram to illustrate the decryption algorithm of this mode. [2 marks]

iii. Consider a $(t+1)$-block CBC mode ciphertext $C_0, C_1, C_2, \ldots, C_t$. Suppose block C_i is replaced by $C_i \oplus \Delta$, where Δ is an n-bit block. Explain how the plaintext corresponding to the modified ciphertext relates to the original plaintext. Pay special attention in your solution to the case $i = 0$. [2 marks]

(d) The XML encryption scheme operates on byte-oriented data and uses the following padding method for CBC mode with a block cipher whose block size n (in bits) is a multiple of 8: at least 1 byte of padding and at most one complete block of padding is appended to the raw message M; if s bytes of padding are needed for some $s \geq 1$, then $s-1$ random bytes are appended to M followed by the byte encoding of integer s. So, for example, if $s = 1$, then the padding appended to the message M is just the single byte `0x01`, while if $s = 2$, then a random byte followed by `0x02` is appended.

i. Write pseudo-code showing how a typical implementation would **remove** the padding from a plaintext to recover M for this padding scheme. Your pseudo-code should work for general block sizes, and you should use a variable name `block_size` indicating the block sizes (in bytes) in your pseudo-code. Your pseudo-code should also generate an error message "`padding_error`" if the padding is invalid in some way. [3 marks]

ii. A simplified version of XML permits bytes of any value to occur in the raw messages M except for `0x00`. After CBC mode decryption and successful padding removal, a simplified XML implementation checks whether the resulting message M contains byte value `0x00`. If a byte with this value is found at any position in M, then the implementation returns an error "`parsing_error`", otherwise normal XML processing continues (and no output is returned).

In the rest of this question, assume you have access to an oracle that returns either "`padding_error`", or "`parsing_error`", or nothing (in cases when the padding removal followed by parsing both succeed).

Now suppose you have a target block of ciphertext C^* that is known to correspond to a full message block P^* (that is, P^* is a plaintext block entirely from M, not containing any padding). Suppose you also know C^*_{-1}, the ciphertext block preceding C^*. By considering two-block ciphertexts of the form $C^*_{-1} \oplus \Delta, C^*$, and using at most 256 oracle queries, explain how to construct a 2-block ciphertext of the form IV^*, C^* whose decryption does **not** cause any padding or parsing errors. Explain why your construction works. (Hint: consider making modifications to the last byte of Δ.) [4 marks]

iii. By modifying Δ in other positions, or otherwise, show how to recover the value of the last byte of plaintext (the padding byte) for the ciphertext IV^*, C^* that you obtained in the previous part of the question. As part of your answer, evaluate how many oracle queries your attack requires in the worst case. (Hint: try to create parsing errors in different positions and think about how these interact with padding removal.) [5 marks]

iv. By further modifying the ciphertext IV^*, C^*, or otherwise, explain **briefly** how you would recover the entire plaintext block P^*. As part of your answer, evaluate how many oracle queries your attack requires in the worst case. (Hint: consider modifying IV^* so that the padding byte contains `0x01`, then try to create parsing errors again.) [4 marks]

Problem 2 (AEAD and generic composition). Recall that a nonce-based AEAD scheme consists of a triple of algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ in which:

- KGen is a randomised algorithm with outputs $K \in \{0,1\}^k$.
- Enc is a deterministic algorithm with inputs key $K \in \{0,1\}^k$, nonce $N \in \mathcal{N} \subseteq \{0,1\}^*$, associated data $AD \in \mathcal{AD} \subseteq \{0,1\}^*$ and message $m \in \mathcal{M} \subseteq \{0,1\}^*$, and outputs $c \in \mathcal{C} \subseteq \{0,1\}^*$. We write $c \leftarrow \mathsf{Enc}(K, N, AD, m)$.
- Dec is a deterministic algorithm with inputs $K \in \{0,1\}^k$, $N \in \mathcal{N}$, $AD \in \mathcal{AD}$ and ciphertext $c \in \{0,1\}^*$ and outputs $m \in \mathcal{M}$ or error message $\perp$. We write $m/\perp \leftarrow \mathsf{Dec}(K, N, AD, c)$.

This question concerns the security of such AEAD schemes constructed using generic composition techniques.

(a) State the correctness definition for nonce-based AEAD. [1 mark]

(b) Security for nonce-based AEAD schemes is defined in terms of two security notions, IND-CPA and INT-CTXT security. Describe both of these notions using pictures to illustrate your answer. In each case, state clearly any restrictions on the adversary's oracle access and define the adversary's advantage. [6 marks]

(c) Let $\mathcal{SE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a nonce-based, IND-CPA secure symmetric encryption scheme in which KGen outputs k-bit keys, nonces $N \in \mathcal{N} = \{0,1\}^n$ for some n, $c \leftarrow \mathsf{Enc}(K, N, m)$ (so $\mathcal{SE}$ does not permit associated data) and $m/\perp \leftarrow \mathsf{Dec}(K, N, c)$. Let $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^t$ be a Pseudo-Random Function (PRF) with keys in $\{0,1\}^k$ and outputs in $\{0,1\}^t$. Consider the following "E&M" construction of a nonce-based AEAD scheme $\mathcal{AEAD}(\mathcal{SE}, F)$ from $\mathcal{SE}$ and F:

- $\mathcal{AEAD}(\mathcal{SE}, F).\mathsf{KGen}$: select $K_1, K_2 \leftarrow_\$ \{0,1\}^k$; return $K = (K_1, K_2)$.
- $\mathcal{AEAD}(\mathcal{SE}, F).\mathsf{Enc}$: on input (K, N, AD, m) with $K = (K_1, K_2)$, set $c = \mathsf{Enc}(K_1, N, m)$, $\tau = F(K_2, AD \,||\, m)$ and return $c' = c \,\|\, \tau$.
- $\mathcal{AEAD}(\mathcal{SE}, F).\mathsf{Dec}$: on input (K, N, AD, c') with $K = (K_1, K_2)$:
  1) return $\perp$ if c' has fewer than t bits;
  2) parse c' as $c \| \tau$ where τ has t bits;
  3) run Dec on input (K_1, N, c); if the result is $\perp$ then return $\perp$, else denote the result by m;

4) compute $\tau' = F(K_2, AD \,||\, m)$;
5) if $\tau' \neq \tau$ return $\perp$; else return m.

By exhibiting a specific adversary $\mathcal{A}_{\text{conf}}$, show that $\mathcal{AEAD}(\mathcal{SE}, F)$ is not IND-CPA secure. Quantify the resources consumed by your adversary and state its approximate advantage. (Hint: the value τ does not depend on N and is a deterministic function of AD and m.) [4 marks]

(d) Consider the modified scheme $\mathcal{AEAD}(\mathcal{SE}, F)^*$ in which $\mathcal{AEAD}(\mathcal{SE}, F)^*.\mathsf{Enc}$ works as follows: on input (K, N, AD, m) with $K = (K_1, K_2)$, set $c = \mathsf{Enc}(K_1, N, m)$, $\tau = F(K_2, N \,||\, AD \,||\, m)$ and return $c' = c \,||\, \tau$. Algorithm $\mathcal{AEAD}(\mathcal{SE}, F)^*.\mathsf{Dec}$ is also modified in the obvious way to account for the changes to the encryption algorithm. Notice that the nonce N *is* now included in the PRF input.

Assuming that $\mathcal{SE}$ is IND-CPA secure and F is a PRF, prove using a sequence of game hops that $\mathcal{AEAD}(\mathcal{SE}, F)^*$ is IND-CPA secure. Provide and justify a security bound relating the advantage of any IND-CPA adversary $\mathcal{A}$ against $\mathcal{AEAD}(\mathcal{SE}, F)^*$ to the advantages of a related PRF adversary $\mathcal{B}$ against F and a related IND-CPA adversary $\mathcal{C}$ against $\mathcal{SE}$.

(Hints: consider game G_0 in which $\mathcal{A}$ runs in the IND-CPA game against $\mathcal{AEAD}(\mathcal{SE}, F)^*$, game G_1 in which τ values in responses to $\mathcal{A}$'s encryption queries are replaced with random t-bit strings, and finally game G_2 in which $\mathcal{A}$'s encryption queries are handled by an IND-CPA adversary $\mathcal{C}$ against $\mathcal{SE}$.) [8 marks]

(e) Show that $\mathcal{AEAD}(\mathcal{SE}, F)^*$ is **not** INT-CTXT secure in general. Justify your answer by exhibiting an attack for some specific scheme $\mathcal{SE}$ that is IND-CPA secure. You do not need to prove IND-CPA security of the scheme $\mathcal{SE}$ that you use. (Hint: consider using nonce-based CTR mode for $\mathcal{SE}$ and manipulating data by moving blocks between m and AD.) [3 marks]

(f) Explain how you would further modify $\mathcal{AEAD}(\mathcal{SE}, F)^*$ to make it INT-CTXT secure whilst maintaining its IND-CPA security. Give informal reasoning for the INT-CTXT security of your proposal. [3 marks]

Problem 3 (Discrete Logarithms and KEMs). Recall that in the "finite field" discrete logarithm setting for public key cryptography, we assume that p, q are large primes with q dividing $p-1$, and we assume that g has order q modulo p, that is, $g^q = 1 \bmod p$. Given $X = g^x \bmod p$ where $x \leftarrow_\$ \mathbb{Z}_q$, the *Discrete Logarithm Problem (DLP)* is to find x when given X and the parameters (p, q, g) as input.

(a) Describe in pseudo-code an algorithm that solves the DLP using $O(q^{1/2})$ multiplications modulo p, $O(q^{1/2})$ storage, and $O(q^{1/2})$ look-ups to that storage, each look-up taking time $O(1)$ on average. (Hint: set $\ell = \lceil q^{1/2} \rceil$; then write $x = \ell \cdot i^* + j^*$ where i^* and j^* are unknown but both bounded by ℓ. Then $X = g^x = (g^\ell)^{i^*} \cdot g^{j^*} \bmod p$ and so $X \cdot (g^\ell)^{-i^*} = g^{j^*} \bmod p$. Given this equation, consider how to make use of a pre-computed list of elements $g^0, g^1, \ldots, g^\ell \bmod p$ to find i^* and j^*.) [5 marks]

(b) What does the preceding algorithm imply concerning the minimum size of q if k bits of security is required for the hardness of the DLP? Justify your answer. [2 marks]

(c) Explain briefly why p must be much larger than the above minimum size of q if k bits of security is required for the DLP. [2 marks]

(d) Recall that a Key Encapsulation Mechanism (KEM) consists of three algorithms $(\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ with a specific syntax and correctness property. OW-CPA and IND-CPA security of a KEM $\mathcal{KEM}$ can be defined in terms of two games played by adversaries $\mathcal{A}$ and $\mathcal{B}$, as shown below (in the IND-CPA game, $\mathcal{K}$ denotes the space of keys encapsulated by the KEM).

Game OW-CPA($\mathcal{A}$, $\mathcal{KEM}$)	Game IND-CPA($\mathcal{B}$, $\mathcal{KEM}$)
1 $(sk, pk) \leftarrow_\$ \mathsf{KGen}$	1 $(sk, pk) \leftarrow_\$ \mathsf{KGen}$
2 $(c^*, K^*) \leftarrow_\$ \mathsf{Encap}(pk)$	2 $b \leftarrow_\$ \{0,1\}$
3 $K \leftarrow_\$ \mathcal{A}(pk, c^*)$	3 $(c^*, K_0) \leftarrow_\$ \mathsf{Encap}(pk)$
4 Return $(K = K^*)$	4 $K_1 \leftarrow_\$ \mathcal{K}$
	5 $b' \leftarrow_\$ \mathcal{A}(pk, c^*, K_b)$
	6 Return $(b' = b)$

We define the advantage of adversary $\mathcal{A}$ in the OW-CPA game against

$\mathcal{KEM}$ to be:

$$\mathbf{Adv}^{\text{OW-CPA}}_{\mathcal{KEM}}(\mathcal{A}) = \Pr\left[\,\text{Game OW-CPA}(\mathcal{A}, \mathcal{KEM}) \Rightarrow \mathsf{true}\,\right].$$

We also define the advantage of adversary $\mathcal{B}$ in the IND-CPA game against $\mathcal{KEM}$ to be:

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{KEM}}(\mathcal{B}) = 2 \cdot \left|\Pr\left[\,\text{Game OW-CPA}(\mathcal{B}, \mathcal{KEM}) \Rightarrow \mathsf{true}\,\right] - \frac{1}{2}\right|.$$

Show using a short security argument why a KEM that is IND-CPA secure must also be OW-CPA secure (no formal advantage bounds are required). [4 marks]

(e) Recall that the Computational Diffie-Hellman Problem (CDHP) in the finite field setting is, given $X = g^x \bmod p$ and $Y = g^y \bmod p$ with $x, y \leftarrow_\$ \mathbb{Z}_q$, to compute $g^{xy} \bmod p$. (Here the parameters (p, q, g) are also given as input.) Explain how the CDHP relates to the DLP. [2 marks]

(f) Explain briefly how to build an OW-CPA secure KEM from the CDHP. Give a sketch reduction showing that OW-CPA security for your proposed KEM relies on the hardness of the CDHP. Concretely relate the advantage of an adversary in the OW-CPA security game to the probability of a related algorithm solving the CHDP .

(Hint: let KGen set $x \leftarrow_\$ \mathbb{Z}_q$ and output $(sk, pk) = (x, g^x \bmod p)$, now think about how to use a second pair $(y, g^y \bmod p)$.) [5 marks]

(g) Let $H : \mathcal{K} \to \{0,1\}^n$ be a hash function. Suppose $\mathcal{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ is a KEM. Consider the KEM $\mathcal{HKEM} = (\mathsf{H.KGen}, \mathsf{H.Encap}, \mathsf{H.Decap})$ built from $\mathcal{KEM}$ as follows:

1) $\mathsf{H.KGen}$: set $(sk, pk) \leftarrow_\$ \mathsf{KGen}$; return (sk, pk).
2) $\mathsf{H.Encap}$: on input pk, set $(c, K) \leftarrow_\$ \mathsf{Encap}(pk)$; return $(c, H(K))$.
3) $\mathsf{H.Decap}$: on input (sk, c), set $K \leftarrow \mathsf{Decap}(sk, c)$; return $H(K)$.

Sketch a proof showing that, in the Random Oracle Model, if $\mathcal{KEM}$ is OW-CPA secure, then $\mathcal{HKEM}$ is IND-CPA secure. [5 marks]

Problem 4 (Applying Crypto in TLS, Signal, and beyond).

(a) In TLS 1.2, the Handshake Protocol can use RSA encryption to transport keying material. Explain how this option is negotiated in TLS, and how it operates. Describe its shortcomings from the point of view of security and performance. [6 marks]

(b) The Logjam attack on TLS is a downgrade attack that exploits certain legacy features of the TLS protocol in combination with client-side weaknesses. Give a brief description of how the attack works, what features of TLS it exploits, and what the attack achieves. [6 marks]

(c) Describe four ways in which the design of TLS 1.3 improves over that of earlier versions of TLS. [4 marks]

(d) Describe in total three similarities or differences between the TLS 1.3 and Signal protocols with respect to authentication and key secrecy properties for the established keys. [3 marks]

(e) The wide-spread deployment of Elliptic Curve Cryptography in general and in TLS in particular took decades. Describe the reasons for this, and the eventual drivers for its adoption in TLS. [3 marks]

(f) Discuss three of the likely challenges that will be encountered as the world transitions to using post-quantum cryptography. [3 marks]

(End of exam paper.)