AMERICAN UNIVERSITY OF BEIRUT

Applied Cryptography

CMPS 297AD/396AI, Fall 2025

Instructor: Nadim Kobeissi **Website:** https://appliedcryptography.page

Problem Set 2: Symmetric Cryptography

Instructions: This problem set covers topics in provable security from parts 1.4[a], 1.5[b] and 1.6[c] of the course. Submit your solutions as a neatly formatted PDF. You are encouraged to collaborate with classmates in studying the material, but your submitted solutions must be your own work. For proofs, clearly state your assumptions, steps, and conclusions.

[a] https://appliedcryptography.page/slides/1-4.pdf
[b] https://appliedcryptography.page/slides/1-5.pdf
[c] https://appliedcryptography.page/slides/1-6.pdf

1 Pseudorandomness (20 points)

1.1 Pseudorandom Generators (10 points)

1. (3 points) Explain the limitations of the one-time pad for practical encryption and why pseudorandom generators (PRGs) are needed in modern cryptographic systems.
2. (3 points) Analyze the security implications of the following PRG construction, where G is a secure PRG:

$$H(S) = A\|B\|C\|D \text{ where } A\|B = G(S) \text{ and } C\|D = G(B)$$

   Determine whether H is a secure PRG. If not, provide a distinguisher that can tell apart $H(S)$ from a truly random string with non-negligible advantage.
3. (4 points) Consider the stream cipher RC4:
   (a) Describe the key components of RC4's design and how it generates a pseudorandom keystream.
   (b) Explain two significant weaknesses that led to RC4 being considered cryptographically broken today.
   (c) What modern alternatives would you recommend as replacements for RC4, and why?

1.2 Pseudorandom Functions and Permutations (10 points)

1. (3 points) Consider the following PRF construction: $F(K,X) = G(K) \oplus X$, where G is a secure PRG.
   (a) Is F a secure PRF? If not, describe a distinguisher that can effectively tell F apart from a random function.
   (b) Explain how this example illustrates the "Golden Rule of PRFs."
2. (4 points) For Feistel ciphers:
   (a) Explain why a 2-round Feistel cipher cannot be a secure pseudorandom permutation (PRP).
   (b) Prove that Feistel ciphers are always permutations, regardless of the security of their round functions.
3. (3 points) Compare and contrast PRFs and PRPs:
   (a) Explain the key differences in their definitions and properties.
   (b) Describe how PRPs can be "downgraded" to PRFs, but not necessarily vice versa.
   (c) Explain why collisions are inevitable for PRFs but not for PRPs.

2 Chosen-Plaintext and Chosen-Ciphertext Attacks (20 points)

2.1 CPA Security (10 points)

1. (5 points) Consider the CPA security definition:

$$\boxed{\begin{array}{l}\mathcal{L}^{\Sigma}_{\text{cpa-real}}\\ \hline K \twoheadleftarrow \Sigma.\mathcal{K}\\ \underline{\text{CPA.ENC}(M):}\\ C := \Sigma.\mathsf{Enc}(K,M)\\ \text{return } C\end{array}} \approxeq \boxed{\begin{array}{l}\mathcal{L}^{\Sigma}_{\text{cpa-rand}}\\ \hline \underline{\text{CPA.ENC}(M):}\\ C \twoheadleftarrow \Sigma.\mathcal{C}(|M|)\\ \text{return } C\end{array}}$$

   (a) Explain why deterministic encryption schemes always fail CPA security.
   (b) Construct a simple distinguisher program that can break the CPA security of any deterministic encryption scheme.
   (c) Analyze what practical security vulnerabilities exist in systems that use non-CPA-secure encryption.

2. (5 points) For each of the following encryption schemes, determine whether it achieves CPA security. If not, provide a specific attack:

   (a) $\mathsf{Enc}(K,M) = (R, F(K,R) \oplus M)$ where $R \twoheadleftarrow \{0,1\}^{\lambda}$ and F is a secure PRF.
   (b) $\mathsf{Enc}(K,M) = (R, F(K,M) \oplus R)$ where $R \twoheadleftarrow \{0,1\}^{\lambda}$ and F is a secure PRF.
   (c) AES in Electronic Codebook (ECB) mode.
   (d) AES in Counter (CTR) mode with a randomly chosen IV.

2.2 CCA Security and Authenticated Encryption (10 points)

1. (3 points) Format oracle attacks:

   (a) Explain how the null-oracle attack works against CTR mode encryption and why it's devastating despite CTR mode being CPA-secure.
   (b) Describe a real-world scenario where a format oracle might be inadvertently exposed in a cryptographic system.
   (c) Calculate the approximate number of oracle queries needed to recover a 1 KB file using the null-oracle attack, and explain why this is practical for an attacker.

2. (4 points) For the following encryption scheme constructions, determine whether each provides CCA security and/or authenticated encryption (AE). Justify your answers with brief explanations:

   (a) Encrypt-then-MAC: $C = \mathsf{Enc}(K_e, M)$, $T = \mathsf{MAC}(K_m, C)$, output (C, T)
   (b) Encrypt-and-MAC: $C = \mathsf{Enc}(K_e, M)$, $T = \mathsf{MAC}(K_m, M)$, output (C, T)
   (c) MAC-then-encrypt: $T = \mathsf{MAC}(K_m, M)$, $C = \mathsf{Enc}(K_e, M \| T)$, output C
   (d) Explain a scenario where replay attacks could succeed even against a system using authenticated encryption, and how associated data (AD) addresses this vulnerability.

3. (3 points) AES-GCM (Galois/Counter Mode):

   (a) Explain how AES-GCM combines CTR mode encryption with Galois field multiplication for authentication. What security advantages does this provide over using separate encryption and MAC algorithms?
   (b) Describe the critical security implications of nonce reuse in AES-GCM. What specific vulnerabilities arise when the same nonce is used for multiple messages?
   (c) AES-GCM is sometimes implemented with different tag lengths. Analyze the security tradeoffs when using 128-bit tags versus 64-bit or 32-bit tags.
   (d) Aside from nonce reuse, what is an unexpected vulnerability in AES-GCM that developers and engineers might not be aware of, but that might significantly impact the security of their software?

3 Collision-Resistant Hash Functions (30 points)

3.1 Hash Function Properties (15 points)

1. (5 points) Collision resistance:
   (a) Explain why collisions must exist in any hash function that maps arbitrary-length inputs to fixed-length outputs.
   (b) Using the birthday paradox, calculate approximately how many hashes must be computed to find a collision with 50% probability in a 256-bit secure hash function.
   (c) Describe a practical attack scenario where finding hash collisions would compromise a security system.
2. (5 points) Hash function construction:
   (a) Compare and contrast the Merkle-Damgård construction (used in SHA-2) and the Sponge construction (used in SHA-3).
   (b) Explain how length extension attacks work against Merkle-Damgård hash functions and why the Sponge construction is resistant to these attacks.
   (c) Describe the HMAC construction and explain how it protects against length extension attacks.
3. (5 points) Hash function evolution:
   (a) Describe the successful attacks against MD5 and SHA-1 that led to their deprecation.
   (b) Explain the concept of chosen-prefix collisions and why they are particularly dangerous for certificate authorities.
   (c) Compare the security of SHA-2 and SHA-3 against known cryptanalytic techniques.

3.2 Password Hashing (15 points)

1. (5 points) For each of the following password storage approaches, analyze the security implications if a server database is compromised:
   (a) Storing passwords in plaintext.
   (b) Encrypting passwords with a key stored on the same server.
   (c) Storing unsalted SHA-256 hashes of passwords.
   (d) Storing salted SHA-256 hashes of passwords.
   (e) Using a specialized password hashing function like Scrypt.
2. (5 points) Salting:
   (a) Explain how salt protects against precomputation attacks like rainbow tables.
   (b) Calculate the storage requirements for properly salted password hashes, assuming 10,000 users, 16-byte salts, and 32-byte hash outputs.
   (c) Describe best practices for generating and storing salts.
3. (5 points) Specialized password hashing functions:
   (a) Explain why memory-hard functions like Scrypt provide better protection against specialized hardware attacks compared to PBKDF2.
   (b) Describe how each of Scrypt's parameters (N, r, p) affect its security and performance.
   (c) Compare the relative speeds of SHA-256, PBKDF2, and Scrypt for password hashing, and explain the security implications of these speed differences.

4 Applied Cryptography Case Studies (30 points)

1. (10 points) **Block Cipher Modes Analysis**

   With reference to the block cipher modes covered in lectures 1-4, 1-5, and 1-6, analyze the following scenarios:

   (a) A secure file storage application needs to encrypt user files at rest. Compare CBC, CTR, and AES-GCM modes for this application, discussing:

   - Performance implications for large files.
   - Error propagation if parts of the ciphertext are corrupted.
   - The security implications of IV/nonce reuse.
   - Data integrity guarantees and the advantages of authenticated encryption with AES-GCM.

   (b) A real-time messaging application needs to encrypt short messages with minimal latency. Compare CBC, CTR, and AES-GCM modes for this application, discussing:

   - Parallelizability for encryption/decryption.
   - Suitability for streaming data.
   - Protection against chosen-ciphertext attacks.
   - How AES-GCM addresses authentication needs compared to unauthenticated modes.

   (c) For AES-GCM specifically:

   - Explain the security impact of nonce reuse in AES-GCM compared to nonce reuse in CTR mode.
   - Discuss the performance tradeoffs of AES-GCM compared to using separate encryption (CTR mode) and authentication (HMAC).
   - Explain how AES-GCM's authenticated encryption properties protect against attacks that would succeed against CBC or CTR modes.

2. (10 points) **Hash Function Security Analysis**

   A software update system uses hash functions to verify the integrity of downloads. The system works as follows:

   - The software vendor posts SHA-1 hashes of legitimate update files on their HTTPS website.
   - Users download the update file over HTTP (not HTTPS) for bandwidth efficiency.
   - The update client verifies the downloaded file by computing its SHA-1 hash and comparing it to the hash obtained from the HTTPS website.
   - If the hashes match, the update is installed automatically.

   Analyze this system:

   (a) Identify at least three security vulnerabilities in this design.
   (b) For each vulnerability, describe a specific attack scenario.
   (c) Propose improvements to address each vulnerability while maintaining performance and usability.
   (d) Design a more secure alternative system using modern cryptographic primitives discussed in class.

3. (10 points) **Password Management System Design**

   You are designing a password management system for a new web application with the following requirements:

   - Users must be able to securely recover their account if they forget their password.
   - The system must be resistant to offline dictionary attacks if the database is compromised.
   - The system must support high-performance authentication for a large user base.
   - The system should detect and prevent credential stuffing attacks.

   Design and analyze a complete solution:

   (a) Specify which cryptographic primitives you would use for password storage and why.

(b) Describe your password recovery mechanism and analyze its security properties.
(c) Explain how your system balances security and performance requirements.
(d) Analyze potential vulnerabilities in your design and how they are mitigated.

Bonus Challenge (20 extra points): The security of AES and other block ciphers depends on their resistance to various forms of cryptanalysis. Research and analyze one of the following advanced attacks:

1. **Side-channel attacks**: Explain how timing attacks, power analysis, or cache attacks can leak information about encryption keys in practical implementations of AES.
2. **Related-key attacks**: Describe how related-key attacks work against block ciphers and why they are significant even when normal usage involves only unrelated keys.
3. **Quantum attacks**: Analyze the impact of Grover's algorithm on the security of AES with different key sizes (128, 192, 256 bits) and discuss appropriate post-quantum key length recommendations.

Your answer should include: a description of the attack, its practical feasibility, relevant examples of successful implementations against real systems, and appropriate countermeasures.