Midterm Review - Symmetric Cryptography

Question 1  True/false

Q1.1  True or False: All cryptographic hash functions are one-to-one functions.

- [ ] True
- [ ] False

**Solution:** False. By definition, a hash function compresses an input which means you’ll always have some collisions \(\implies\) not one-to-one. Cryptographic hash functions try to make finding those collisions difficult, but they still exist.

Q1.2  True or False: If \(k\) is a 128 bit key selected uniformly at random, then it is impossible to distinguish AES\(_k(\cdot)\) from a permutation selected uniformly at random from the set of all permutations over 128-bit strings.

*Clarification made during the exam: AES\(_k(\cdot)\) refers to the encryption function of AES using key \(k\).*

- [ ] True
- [ ] False

**Solution:** True. AES is believed to be secure, which means that no known algorithm can distinguish between AES\(_k(\cdot)\) and a truly random permutation so long as \(k\) is selected uniformly at random.

Q1.3  True or False: A hash function that is one-way but not collision resistance can be securely used for password hashing.

- [ ] True
- [ ] False

**Solution:** True. Collisions don’t matter in this context as the only property we want is that an attacker can’t invert a hash.

Q1.4  True or False: A hash function whose output always ends in 0 regardless of the input can’t be collision resistant.

- [ ] True
- [ ] False
Solution: False. Consider $H(x) = SHA256(x) || 0$. This hash is collision resistant but always ends in a 0.
**Question 2  ** **AES-CBC-STAR**

Let \( E_k \) and \( D_k \) be the AES block cipher in encryption and decryption mode, respectively.

Q2.1 We invent a new encryption scheme called AES-CBC-STAR. A message \( M \) is broken up into plaintext blocks \( M_1, \ldots, M_n \) each of which is 128 bits. Our encryption procedure is:

\[
C_0 = \text{IV} \text{ (generated randomly)}, \\
C_i = E_k(C_{i-1} \oplus M_i) \oplus C_{i-1}.
\]

where \( \oplus \) is bit-wise XOR.

- Write the equation to decrypt \( M_i \) in terms of the ciphertext blocks and the key \( k \).

**Solution:** \( M_i = D_k(C_i \oplus C_{i-1}) \oplus C_{i-1}. \)

Q2.2 Mark each of the properties below that AES-CBC-STAR satisfies. Assume that the plaintexts are 100 blocks long, and that \( 10 \leq i \leq 20 \).

- Encryption is parallelizable.
- Decryption is parallelizable.
- If \( C_i \) is lost, then \( C_{i-1} \) can still be decrypted.
- If we flip the least significant bit of \( C_i \), this always flips the least significant bit in \( P_i \) of the decrypted plaintext.
- If \( C_i \) is lost, then \( C_{i+1} \) can still be decrypted.
- If \( C_i \) is lost, then \( C_{i+2} \) can still be decrypted.
- If \( C_i \) is lost, then \( C_{i-2} \) can still be decrypted.
- If we flip the least significant bit of \( C_i \), this always flips the least significant bit in \( P_{i+1} \) of the decrypted plaintext.
- If we flip a bit of \( M_i \) and re-encrypt using the same IV, the encryption is the same except the corresponding bit of \( C_i \) is flipped.
- It is not necessary to pad plaintext to the blocksize of AES when encrypting with AES-CBC-STAR.

Q2.3 Now we consider a modified version of AES-CBC-STAR, which we will call AES-CBC-STAR-STAR. Instead of generating the IV randomly, the challenger uses a list of random numbers which are public and known to the adversary. Let \( \text{IV}_i \) be the IV which will be used to encrypt the \( i \)th message from the adversary.

- Argue that the adversary can win the IND-CPA game.

**Solution:** Adversary sends two arbitrary (unequal but equal length), one-block messages \( (M, M') \) as the challenge. The resulting ciphertext is either \( C_0 = \text{IV}_0 || E_k(\text{IV}_0 \oplus M) \oplus \text{IV}_0 \) or \( C_0 = \text{IV}_0 || E_k(\text{IV}_0 \oplus M') \oplus \text{IV}_0. \)
Next the adversary sends $IV_1 \oplus IV_0 \oplus M$. The resulting ciphertext is $C_1 = IV_1 || E_k(IV_1 \oplus (IV_0 \oplus IV_1 \oplus M)) \oplus IV_1$, which simplifies to $IV_1 || E_k(IV_0 \oplus M) \oplus IV_1$. If the second block of $C_1 \oplus IV_1$ equals the second block of $C_0 \oplus IV_0$, then the challenger encrypted $M$. Otherwise the challenger encrypted $M'$. Hence we break IND-CPA with advantage significantly above $\frac{1}{2}$ (in fact such an adversary wins all the time).

An alternative solution is to send the challenger ciphertexts $M = IV_1$ and $M' = \text{anything else}$. If the challenger encrypts $M$, the message received is $E_k(0) \oplus IV_1$. Then for the second message, send $IV_2$. If the output ciphertext $\oplus IV_1 \oplus IV_2$ equals the challenge ciphertext, then the challenger encrypted $M$. Otherwise they encrypted $M'$. 
Question 3

Alice comes up with a couple of schemes to securely send messages to Bob. Assume that Bob and Alice have known RSA public keys.

For this question, Enc denotes AES-CBC encryption, H denotes a collision-resistant hash function, || denotes concatenation, and ⊕ denotes bitwise XOR.

Consider each scheme below independently and select whether each one guarantees confidentiality, integrity, and authenticity in the face of a MITM.

Q3.1 (3 points) Alice and Bob share two symmetric keys $k_1$ and $k_2$. Alice sends over the pair $[\text{Enc}(k_1, \text{Enc}(k_2, m)), \text{Enc}(k_2, m)]$.

- (A) Confidentiality
- (B) Integrity
- (C) Authenticity
- (D) —
- (E) —
- (F) —

Solution: Note that Enc denotes AES-CBC, not AES-EMAC, so we can only provide confidentiality. An attacker can forge a pair $[\text{Enc}(k_1, c_1), c_1]$ given $[\text{Enc}(k_1, c_1||c_2), c_1||c_2]$.

Q3.2 (3 points) Alice and Bob share a symmetric key $k$, have agreed on a PRNG, and implement a stream cipher as follows: they use the key $k$ to seed the PRNG and use the PRNG to generate message-length codes as a one-time pad every time they send/receive a message. Alice sends the pair $[m \oplus \text{code}, \text{HMAC}(k, m \oplus \text{code})]$.

- (G) Confidentiality
- (H) Integrity
- (I) Authenticity
- (J) —
- (K) —
- (L) —

Solution: This stream cipher scheme has confidentiality since the attacker has no way of coming up with the pseudorandomly generated one-time pads. HMAC provides the integrity and authentication.

Q3.3 (3 points) Alice and Bob share a symmetric key $k$. Alice sends over the pair $[\text{Enc}(k, m), \text{H(Enc}(k, m))]$.

- (A) Confidentiality
- (B) Integrity
- (C) Authenticity
- (D) —
- (E) —
- (F) —

Solution: Public hash functions alone do not provide integrity or authentication. Anyone can forge a pair $c, H(c)$, which will pass the integrity check and can be decrypted.
Q3.4 (3 points) Alice and Bob share a symmetric key $k$. Alice sends over the pair $[\text{Enc}(k, m), H(k||\text{Enc}(k, m))]$.

- (G) Confidentiality
- (H) Integrity
- (I) Authenticity
- (J) —
- (K) —
- (L) —

**Solution:** $H(k||\text{Enc}(k, m))$ is not a valid substitute for $HMAC$ because it is vulnerable to a length extension attack.