Exam – Cryptology 1 – 01410

16.05.2025

With solutions

Instructions and advice

- For all calculations in Part II: explain how you have computed your results. Unexplained results will receive few or no points. If you use a computer (or similar), then you need to be able to explain how the computer arrived at the answer.
- The problems have been created such that it is possible to solve them without the help of a computer or similar.
- Read all the questions first, and begin to work on the ones you find easy.

Part I – Select all that apply (36 points)

You get 2 points for every correct selection and -2 points for every incorrect selection. As for any multiple choice quiz, here only the selection counts. There is no need to describe your reasoning. The lowest number of points that you can get for any of the 5 multiple choice problems is 0.

1. Which of the following statements about unconditionally secure and pseudorandom cryptographic primitives are true? Select all that apply.
   A. A real-or-random perfectly secure symmetric key encryption scheme can only be constructed using the One-time Pad. **Solution:** No. counterexample: define the encryption algorithm as encryption with the one-time pad and then appending a 0. The decryption is the one-time pad decryption ignoring the last bit of the ciphertext.
   B. Encrypting a message using the One-time Pad symmetric key encryption algorithm using key k_1 and encrypting the resulting ciphertext using another instance of the One-time Pad with key k_2 is equivalent to applying the One-time Pad once (using another key). **Solution:** Yes
   C. If one encrypts a message using the One-time Pad symmetric key encryption algorithm and then encrypting the resulting ciphertext using the AES-128 symmetric key encryption scheme with a different key, then the resulting symmetric key encryption scheme is not perfectly secure. **Solution:** No. After encrypting with the one-time pad, any operation that doesn't depend on the message or the key will preserve perfect security.
   D. Let P be a Pseudorandom Permutation of block-size B and key-length λ, and let k_1, k_2 be keys. Let $P(k, x)$ denote applying P to input $x \in \{0,1\}^B$ with key k. Then $P(k_2, P(k_1, x))$ is a PRP with key-length 2λ and block-size B (the key is $k_1 \| k_2$). **Solution:** Yes
   E. Due to the birthday paradox, Pseudorandom permutations of block-size B can be distinguished from Pseudorandom Functions of input and output length B by an attack that obtains around than $B/2$ input/output samples. **Solution:** No, approximately $2^{B/2}$ queries are necessary by the birthday bound.

F. Pseudorandom Permutations are an abstraction of Public-key Encryption. **Solution:** No, of block ciphers

G. The AES block cipher first expands the encryption key k into multiple round keys. Then, during encryption, it applies the same function (in almost every round) repeatedly on the input, each time using a different round key. The input of the first round is the message, while every other round takes the previous round's output as input. **Solution:** Yes

H. The AES block cipher follows the Feistel design pattern. **Solution:** No.

2. Which of the following statements about Modes of Operation, Hash functions and MACs are true? Select all that apply.

A. The IND-CPA security of CTR mode fails if the initially chosen random value r is reused. **Solution:** Yes.

B. In the ECB mode of operation, each block of the input message of length B is encrypted by applying an arbitrary secure Pseudorandom Function on the block. All PRF calls use the same key, and the construction always works as long as the PRF has an output length of at least B bits. **Solution:** No, as ECB uses a PRP since we otherwise cannot decrypt later.

C. Any Pseudorandom function of output length λ and key-length λ can be used to implement a MAC scheme that is EUF-CMA secure against attackers with runtime polynomial in λ. **Solution:** Yes.

D. The length of a CBC MAC tag scales with the length of the message that is protected. **Solution:** No, the length of a CBC MAC tag is equal to the block length of the used block cipher.

E. Secure MAC schemes protect both the confidentiality and integrity of a message. **Solution:** No, they only protect the integrity

F. A cryptographic hash function is a family of deterministic functions that maps an input of variable length to a fixed output length. **Solution:** Yes

G. A cryptographic hash function can be used to make RSA IND-CCA secure using the OAEP transform. **Solution:** Yes.

H. The Merkle-Damgård construction for hash functions is prone to length-extension attacks. **Solution:** No, it employs a padding scheme to avoid length extension attacks.

3. Which of the following statements are true about Public key encryption and digital signatures? Select all that apply.

A. 4 is often used as an RSA public exponent, due to its small size. **Solution:** No. In particular, the public exponent e needs to be coprime with the Euler totient function of the modulus, i.e., with $\phi(N) = (p-1)(q-1)$ where $N = p \cdot q$ for large primes p and q. As p and q are odd, 4 actually divides $\phi(N)$.

B. If one can efficiently compute the Euler totient function $\phi(N)$ of an arbitrary number N, then RSA is insecure. **Solution:** Yes.

C. The ElGamal public key cryptosystem can only be constructed if discrete logarithms in $\mathbb{Z}_p^*$ are hard to compute. **Solution:** No. This is actually doubly false: First of all, it can be constructed from any group, even if it is not secure. And secondly, it is not known whether the hardness of the discrete logarithm problem implies the hardness of the decisional DIffie-Hellman problem (which is necessary for the *security* of ElGamal). So even if the statement was "The ElGamal public key cryptosystem can only be *secure* if discrete logarithms....", this would still be false: It could be that discrete logs are hard and decisional Diffie-Hellman is easy, so ElGamal would be broken.

D. The ElGamal public key cryptosystem is IND-CPA secure. **Solution:** Yes.

E. In a digital signature scheme, both the sender and the receiver share a secret key. **Solution:** No, digital signatures are a public-key functionality where the sender has a private key and the receiver (and everybody else) has the corresponding public verification key.

F. If there exists an efficient attacker against the Computational Diffie-Hellman problem in a certain group G, then there exists an efficient attacker against Decisional Diffie-Hellman in G. **Solution:** Yes.

G. The X3DH (Signal) key exchange protocol, as presented in the lecture, is secure against attacks with quantum computers. **Solution:** No, it uses quantum-broken Diffie-Hellman

4. Which of the following statements are true about Post-Quantum Cryptography and LWE? Select all that apply.

A. It is known how to break block ciphers with the help of a quantum algorithm for the period-finding problem. **Solution:** No.

B. It is known how to break RSA with the help of a quantum algorithm for the period-finding problem. **Solution:** Yes, using Shor's algorithm

C. Quantum computing attacks break the Diffie-Hellman key exchange protocol both in its original and elliptic curve variants. **Solution:** Yes

D. There are known polynomial-time quantum algorithms for breaking all practical cryptography. **Solution:** No, not symmetric cryptography like block ciphers

E. There is a known quantum algorithm to solve the Learning With Errors problem (underlying Regev encryption) in polynomial time. **Solution:** No

F. In Regev's encryption scheme, the message is used as the secret vector $\mathbf{s}$ in the Learning With Errors (LWE) problem (the decision LWE problem is the task of distinguishing $\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e}$ from a uniformly random pair of data with the same sizes, where $\mathbf{e}$ is small). **Solution:** No, it is added as an additional "big error" to the LWE sample

5. Which of the following statements are true in basic cryptography? Select all that apply.

A. Kerckhoffs' principle states that the confidentiality of a cryptographic key should be at least one out of several secret data that form the basis of cryptographic security. **Solution:** No, it should be the only one.

B. A cryptographic algorithm that is evaluated in an attack model that gives the attacker more power enjoys stronger security guarantees **Solution:** Yes.

C. Security definitions are necessary for mathematical proofs of cryptographic security. **Solution:** Yes.

D. If $f(n) > |p(n)|$ for some polynomial p and some integer $n > 0$, f is not negligible. **Solution:** No. It is ok if this only holds for n that are larger then some constant c (that can depend on the functions f, p).

E. IND-CPA security ensures that a scheme is secure even against attackers who can request decryptions of ciphertexts of their choice, as long as they don't include the challenge ciphertext. **Solution:** false, the description matches the definition of IND-CCA security.

F. A (properly generated) one-time pad ciphertext is a uniformly random string from the attacker's perspective. **Solution:** Yes.

G. It is fundamentally impossible to determine whether an attack against AES succeeds with non-negligible probability, as AES is not defined with a security parameter that can take any positive integer value. **Solution:** Yes.

Part II (34 points)

1. (6 points) Consider the function $F : \{0,1\}^\lambda \times \{0,1\}^B \to \{0,1\}^B$. If F was a PRP, then we would like that $L^F_{PRP-Real} \approx L^F_{PRP-Rand}$ where

Library $L^F_{PRP-Real}$

Sample $k \leftarrow K$ uniformly at random.

Lookup$(x \in \{0,1\}^B)$

```
1. $y \leftarrow F(k, x)$
2. Output $y$
```

and

Library $L^F_{PRP-Rand}$

Let T be an empty map and S be an empty set.

Lookup$(x \in \{0,1\}^B)$

```
1. If $T[x]$ is undefined then sample $T[x] \leftarrow \{0,1\}^B \setminus S$
2. Add $T[x]$ to $S$
3. Output $T[x]$
```

(a) Let $G : \{0,1\}^\lambda \times \{0,1\}^B \times \{0,1\}^B \to \{0,1\}^B \times \{0,1\}^B$ be defined as computing $G(k, x_1, x_2) = (F(k, x_1), F(k, x_2))$. Since F is assumed to be a PRP, there must exist an efficiently computable algorithm F^{-1} such that $F^{-1}(k, F(k, x)) = x$. Use the algorithm F^{-1} to construct an algorithm G^{-1} such that $G^{-1}(k, G(k, x_1, x_2)) = (x_1, x_2)$ and show that it is correct.

(b) Show that the function G defined in the previous part is not a PRP, i.e. that $L^G_{PRP-Real}$ and $L^G_{PRP-Rand}$ can easily be distinguished. Towards this, describe what happens on input (x_1, x_2) if $x_1 = x_2$ or $x_1 \neq x_2$. Then show how this can be used in an attack that distinguishes both libraries.

(c) The attack from the previous part will not work on a function G' where $G'(k, x_1, x_2) = (F(k, x_1), F(k, x_2 \oplus (1, \ldots, 1)))$ where $(1, \ldots, 1)$ is a vector of 1s of length B and $\oplus$ is the coordinate-wise XOR operation. Can you show how to modify your attack such that it still succeeds?

Solution: We are given a function $F : \{0,1\}^\lambda \times \{0,1\}^B \to \{0,1\}^B$ which is a PRP, so it is invertible with an efficient inverse F^{-1} such that

$$F^{-1}(k, F(k, x)) = x \quad \text{for all } k \in \{0,1\}^\lambda, x \in \{0,1\}^B.$$

Define $G(k, x_1, x_2) = (F(k, x_1), F(k, x_2))$. We want to construct G^{-1} such that

$$G^{-1}(k, G(k, x_1, x_2)) = (x_1, x_2).$$

Construction of G^{-1}:

$$G^{-1}(k, y_1, y_2) = (F^{-1}(k, y_1), F^{-1}(k, y_2)).$$

Correctness: Given (x_1, x_2), we compute

$$G(k, x_1, x_2) = (F(k, x_1), F(k, x_2)) = (y_1, y_2),$$

then

$$G^{-1}(k, y_1, y_2) = (F^{-1}(k, F(k, x_1)), F^{-1}(k, F(k, x_2))) = (x_1, x_2).$$

Hence, G^{-1} correctly inverts G.

To show that G is not a PRP, we demonstrate a distinguisher that can distinguish between $L^G_{PRP-Real}$ and $L^G_{PRP-Rand}$.

Observation:

- In $L^G_{PRP-Real}$: We have $G(k, x, x) = (F(k, x), F(k, x))$, i.e. the first and second parts of the output are equal if the first and second input are equal.
- In $L^G_{PRP-Rand}$: Outputs are uniformly random (under the constraint that they are distinct), so if (x, x) is the first query, then the output is uniformly random so the first and second half are equal with probability 2^{-B}.

Attack:

1. Query (x, x) to the Lookup oracle.
2. If the output is of the form (y, y) (i.e., both outputs are equal), guess "Real".
3. If not, guess "Random".

Success probability:

- In $L^G_{PRP-Real}$, we get $(F(k, x), F(k, x))$ with probability 1.
- In $L^G_{PRP-Rand}$, we get (y, y) with probability 2^{-B}.

Advantage:

$$\left|\Pr[\text{Real} \mid L^G_{PRP-Real}] - \Pr[\text{Real} \mid L^G_{PRP-Rand}]\right| - \frac{1}{2} = \frac{1}{2} - 2^{-B},$$

which is non-negligible. Thus, G is *not* a PRP.

Now consider $G'(k, x_1, x_2) = (F(k, x_1), F(k, x_2 \oplus (1, \ldots, 1)))$. Let $v = (1, \ldots, 1) \in \{0, 1\}^B$.

Modified Attack:

1. Query $(x, x \oplus v)$ and receive (y_1, y_2).
2. If $y_1 = y_2$, guess "Real"; else guess "Random".

Analysis: It suffices to observe that $G'(x, x \oplus v) = G(x, x)$, and that for a random permutation T, $(x, y) \mapsto T(x, y \oplus v)$ is also a random permutation. The attack has thus the same success probability as the attack against G.

2. (10 points) Consider the following Shifted-CBC, or sCBC, mode of encryption, a variant of the standard CBC mode.

   Let a message m be divided into n blocks:

$$m = m_1 \,\|\, m_2 \,\|\, \ldots \,\|\, m_n,$$

   where $m_i \in \{0, 1\}^B$.

Let E_k be a secure block cipher with key k, and let $IV \in \{0,1\}^B$ be a public initialization vector. The encryption algorithm Enc is defined as follows: On input key k and message $m = m_1 \,\|\, m_2 \,\|\, \dots \,\|\, m_n$, set

$$C_0 = IV$$

$$C_1 = E_k(m_1 \oplus C_0)$$

$$C_i = E_k(m_i \oplus C_{i-2}) \quad \text{for } i \geq 2$$

and output the ciphertext is $C = C_1 \,\|\, C_2 \,\|\, \dots \,\|\, C_n$.

Let us also quickly remember the libraries for the IND-CPA security game for a symmetric key encryption scheme $\Sigma = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of message space M:

Library L^{Σ}_{CPA-0}

```
Compute $k \leftarrow \Sigma.\mathsf{KG}()$.
$CTXT(m_0 \in \Sigma.M, m_1 \in \Sigma.M)$
    1. If $|m_0| \neq |m_1|$ then output $\perp$.
    2. Output $\Sigma.\mathsf{Enc}(k, m_0)$
```

and

Library L^{Σ}_{CPA-1}

```
Compute $k \leftarrow \Sigma.\mathsf{KG}()$.
$CTXT(m_0 \in \Sigma.M, m_1 \in \Sigma.M)$
    1. If $|m_0| \neq |m_1|$ then output $\perp$.
    2. Output $\Sigma.\mathsf{Enc}(k, m_1)$
```

(a) Describe a decryption algorithm for this mode of operation and prove that it achieves correctness.

(b) Consider the 2-block messages $m = m_1 \| m_1$ and $m' = m_1 \| m_2$ where $m_1 \neq m_2$ are some arbitrary message blocks. Show how these two messages can be used to construct a successful distinguisher for the IND-CPA libraries.

(c) Suppose now that the message space for this mode of operation is restricted to messages $m = m_1 \,\|\, m_2 \,\|\, \dots \,\|\, m_n$ such that $m_1 \neq m_2$, and call this variant sCBC'. Argue that this mode is IND-CPA secure for 2-block messages.

(d) Describe how to modify the decryption algorithm such that it still achieves correctness for sCBC', but never outputs $m = m_1 \,\|\, m_2 \,\|\, \dots \,\|\, m_n$ such that $m_1 = m_2$.

(e) Bonus: Show that sCBC' with your decryption algorithm from (d) is not IND-CCA secure, even when restricted to 2-block messages.

Solution: Decryption Algorithm and Correctness: Given ciphertext $C = C_1 \| C_2 \| \ldots \| C_n$ and public $IV = C_0$, we define the decryption algorithm Dec_k as follows:

Decryption:

$$m_1 = D_k(C_1) \oplus C_0$$

$$m_i = D_k(C_i) \oplus C_{i-2} \quad \text{for } i \geq 2$$

where D_k is the decryption algorithm for the block cipher E_k.

Correctness Proof:

- For $i = 1$:
$$C_1 = E_k(m_1 \oplus C_0) \Rightarrow D_k(C_1) = m_1 \oplus C_0 \Rightarrow m_1 = D_k(C_1) \oplus C_0.$$
- For $i \geq 2$:
$$C_i = E_k(m_i \oplus C_{i-2}) \Rightarrow D_k(C_i) = m_i \oplus C_{i-2} \Rightarrow m_i = D_k(C_i) \oplus C_{i-2}.$$

Hence, the decryption algorithm correctly recovers m from C.

Distinguisher for IND-CPA Game: Let the adversary query:

$$CTXT(m = m_1 \| m_1, \quad m' = m_1 \| m_2),$$

where $m_1 \neq m_2$. Let the output ciphertext be $C = C_1 \| C_2$.

Case 1: If the challenger uses m:

$$C_1 = E_k(m_1 \oplus IV)$$

$$C_2 = E_k(m_1 \oplus IV) = C_1$$

Case 2: If the challenger uses m':

$$C_1 = E_k(m_1 \oplus IV)$$

$$C_2 = E_k(m_2 \oplus IV) \neq C_1$$

Attack:

1. Submit $(m_1 \| m_1, \, m_1 \| m_2)$ to the $CTXT$ oracle.
2. Receive $C_1 \| C_2$.
3. If $C_1 = C_2$, guess 0 (i.e., L_{CPA-0}). Otherwise, guess 1 (L_{CPA-1}).

Advantage: The adversary distinguishes the two cases correctly with probability 1.

sCBC' Security under Restriction $m_1 \neq m_2$: In this case, the previous attack fails because both m_1 and m_2 must satisfy $m_1 \neq m_2$, so it is no longer possible to force $C_1 = C_2$.

Argument:

- Since E_k is a secure block cipher, the output $C_i = E_k(m_i \oplus C_{i-2})$ behaves pseudorandomly.
- Even knowing IV, and with $m_1 \neq m_2$, both message options for any IND-CPA query will produce ciphertexts indistinguishable under the pseudorandomness of E_k.

Therefore, under this restriction, sCBC' is IND-CPA secure for 2-block messages.

Modify Decryption to Reject $m_1 = m_2$: We modify the decryption algorithm as follows:

1. Decrypt as usual to obtain $m_1, m_2, \ldots, m_n$.
2. If $m_1 = m_2$, output $\perp$ (reject).
3. Otherwise, output $m_1 \,\|\, m_2 \,\|\, \ldots \,\|\, m_n$.

This ensures correctness when $m_1 \neq m_2$, and enforces rejection of forbidden messages with $m_1 = m_2$.

Bonus: sCBC' is not IND-CCA secure:

Attack:

1. Submit $(m_1 \,\|\, m_2,\ m_2 \,\|\, m_1)$ to the IND-CCA challenge oracle.
2. Get back ciphertext $C = C_1 \,\|\, C_2$.
3. Construct $C' = C_2 \,\|\, C_1$ and submit it to the decryption oracle.
4. If the decryption oracle returns $m_2 \| m_1$, guess $b = 0$, else guess $b = 1$.

Rationale: Swapping blocks flips the order of the message. For $m_1 \,\|\, m_2$, the decryption of $C_2 \,\|\, C_1$ yields $m_2 \,\|\, m_1$ (still satisfying $m_1 \neq m_2$), while for $m_2 \,\|\, m_1$, we get $m_1 \,\|\, m_2$. This enables the adversary to determine b with certainty.

Therefore, sCBC' is *not* IND-CCA secure, even for 2-block messages.

3. (10 points) Recall the RSA algorithm from the lecture:

KG()**:**

1. Sample two different odd prime numbers p, q and set the RSA modulus $N = p \cdot q$.
2. Set $\varphi(N) = (p-1)\cdot(q-1)$ and choose e such that $gcd(e, \varphi(N)) = 1$. Then compute $d = e^{-1} \bmod \varphi(N)$.
3. Output (pk, sk) where $pk = (e, N)$ is the public key and $sk = (d, N)$ is the private key.

Enc(pk, m)**:** To encrypt the message $m \in \mathbb{Z}_N^*$ using public key $pk = (e, N)$:

1. Compute $c = m^e \bmod N$
2. Output c.

Dec(sk, c)**:** To decrypt a ciphertext $c \in \mathbb{Z}_N^*$ using private key $sk = (d, N)$:

1. Compute $m = c^d \bmod N$
2. Output m.

We also quickly recap the libraries defining security of a digital signature scheme, which were $L_{sig-real}, L_{sig-fake}$ where

Library $L_{sig-real}$

```
Run $(vk, sk) \leftarrow$ KG().
$GetKey()$
        1. Return $vk$
$Getsig(m)$
```

```
    1. $\sigma \leftarrow \mathsf{Sig}(vk, m)$
    2. Return $\sigma$
$Versig(m, \sigma)$
    1. Return $\mathsf{Ver}(vk, m, \sigma)$
```

and

Library $L_{sig-fake}$

```
Run $(vk, sk) \leftarrow \mathsf{KG}()$ and let $S$ be an empty set.
$GetKey()$
    1. Return $vk$
$Getsig(m)$
    1. $\sigma \leftarrow \mathsf{Sig}(sk, m)$
    2. $S \leftarrow S \cup \{(m, \sigma)\}$
    3. Return $\sigma$
$Versig(m, \sigma)$
    1. Return $(m, \sigma) \in S$.
```

(a) Show that 17 is a possible value for the public exponent e for RSA modulus $N = 71 \cdot 89$ where 71 and 89 are the prime factors of N. Explain your reasoning!

(b) Let $N = 71 \cdot 89$, $e = 17$. Compute the encryption of the message $m = 2$.

(c) Show that $d = 5073$ is the correct decryption key for $N = 71 \cdot 89$, $e = 17$.

(d) We now define the following attempt of a signature scheme:

```
KG():
    1. Output $(vk, sk) \leftarrow RSA.\mathsf{KG}()$.
Sig(sk, m): To sign the message $m \in \mathbb{Z}_N^*$ using signing key $sk$:
    1. Compute $\sigma = m^{2d} \bmod N$
    2. Output $\sigma$.
Ver(vk, m, σ): To verify a signature $\sigma \in \mathbb{Z}_N^*$ on a message $m \in \mathbb{Z}_N^*$ using verification key $vk = (e, N)$:
    1. Output 1 if $m^2 = \sigma^e \bmod N$, otherwise output 0
```

Show that the 3 algorithms form a correct digital signature scheme.

(e) Compute the signatures of messages $m_1 = 1, m_2 = N - 1 = -1 \bmod N$ and show that this can be used to construct an efficient algorithm that can break the EUF-CMA property of the digital signature scheme, i.e. that can distinguish $L_{sig-real}$ from $L_{sig-fake}$.

Solution: For part (a), we need to check whether $e = 17$ is coprime with $\varphi(N)$ where $N = 71 \cdot 89$. Compute $\varphi(N) = (71 - 1)(89 - 1) = 70 \cdot 88 = 6160$. Now compute $\gcd(17, 6160)$. Since 17 is a prime number and does not divide 6160, we find $\gcd(17, 6160) = 1$, so 17 is a valid public exponent.

For part (b), we are given $N = 6319$ and $e = 17$, and must compute $c = 2^{17} \bmod 6319$. Compute $2^{13} = 8192 = 1873 \bmod 6319$, $2^{15} = 1873 \cdot 4 = 7492 = 1173 \bmod 6319$ and thus $2^{17} = 4692 \bmod 9319$. So $c = 2^{17} \bmod 6319 = 4692$.

For part (c), we are given $d = 5073$ and must verify that $ed \equiv 1 \bmod \varphi(N)$ with $e = 17$ and $\varphi(N) = 6160$. Compute $17 \cdot 5073 = 86241$. Now compute $86241 \bmod 6160$. Since $86241 \div 6160 \approx 14$, $6160 \cdot 14 = 86240$, and $86241 - 86240 = 1$, so indeed $17 \cdot 5073 \equiv 1 \bmod 6160$, confirming that $d = 5073$ is the correct private exponent.

For part (d), we show that this signature scheme is correct. Signing computes $\sigma = m^{2d} \bmod N$. Verification checks if $m^2 \equiv \sigma^e \bmod N$. Since $\sigma = m^{2d}$, then $\sigma^e = (m^{2d})^e = m^{2de}$. We know $de \equiv 1 \bmod \varphi(N)$, so $m^{2de} \equiv m^2 \bmod N$ for all $m \in \mathbb{Z}_N^*$, hence the verifier accepts, confirming correctness.

For part (e), let us compute the signatures of $m_1 = 1$ and $m_2 = -1 \bmod N = 6318$. Then $\sigma_1 = 1^{2d} = 1 \bmod N$, and $\sigma_2 = (-1)^{2d} = 1 \bmod N$ as $2d$ is even. So both m_1 and m_2 yield the same signature $\sigma = 1$. Now consider an adversary who queries $Getsig(1)$ and receives $\sigma = 1$, and then evaluates $Versig(-1, \sigma = 1)$. In the real signature library, $(-1)^2 = 1 = \sigma^e \bmod N$, so the verifier accepts. But in the fake signature library, the pair $(-1, 1)$ is not in S, so verification returns 0. Hence, the adversary can distinguish $L_{sig-real}$ from $L_{sig-fake}$, breaking EUF-CMA security.

4. (8 points) This problem is about the Diffie-Hellman key exchange (DHKE) protocol. For the public parameters of the protocol, let p be prime and $g \in \mathbb{Z}_p^*$ be a generator of $\mathbb{Z}_p^*$. (Reminder: a group element $g \in G$ is a generator of a multiplicative group G if it holds that $G = \{g^0, g^1, g^2, \ldots\}$)

   (a) As a toy example, for this sub-problem only, let $p = 23$ and $g = 5$. Suppose that Alice and Bob choose secret exponent $a = 3$ and $b = 13$, respectively. Compute the protocol messages that Alice and Bob send to each other, and the key output by Alice and Bob.

   (b) To generate a longer key, we can execute the DHKE twice, in parallel, and concatenate the keys. Formally describe the resulting protocol 2DHKE: Specify how Alice and Bob choose their respective protocol message, and how they compute the key from the message they receive.

   (c) Show that the protocol 2DHKE cannot be more secure than DHKE. To do that, explain how an attack against the security of DHKE (key agreement security is defined in Definition 14.4 in the book) can be used to break 2DHKE.

   (d) Bonus: Give am argument that the security of DHKE implies the security of 2DHKE. To that end, consider the libraries $\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}$ and $\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}$ from the Key Agreement security definition for 2DHKE, and an intermediate library $\mathcal{L}^{\text{2DHKE}}_{\text{int}}$ where one of the two parallel DHKE instances behaves as in $\mathcal{L}^{\text{DHKE}}_{\text{ka-real}}$ and the other one as in $\mathcal{L}^{\text{DHKE}}_{\text{ka-rand}}$. Then show that any adversary that can distinguish $\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}$ and $\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}$ succeeds at least at distinguishing $\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}$ and $\mathcal{L}^{\text{2DHKE}}_{\text{int}}$, or at distinguishing $\mathcal{L}^{\text{2DHKE}}_{\text{int}}$ and $\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}$. Finally, describe informally why distinguishing $\mathcal{L}^{\text{2DHKE}}_{\text{int}}$ from either $\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}$ or $\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}$ is enough to distinguish $\mathcal{L}^{\text{DHKE}}_{\text{ka-real}}$ and $\mathcal{L}^{\text{DHKE}}_{\text{ka-rand}}$.

Solution: For part (a), we are given $p = 23$, $g = 5$, $a = 3$, and $b = 13$. Alice computes $A = g^a \bmod p = 5^3 \bmod 23 = 2 \cdot 5 \bmod 23 = 10$, using $5^2 = 25 = 2 \bmod 23$. Bob computes $B = g^b \bmod p = 5^{13} \bmod 23 = 5 \cdot 2^6 \bmod 23 = -3 \cdot 2^4 \bmod 23 = 21$ (using $5^2 = 25 = 2 \bmod 23$, $5 * 2^2 = 20 = -3 \bmod 23$, $2^4 = 16 = -7 \bmod 23$ and $-3 \cdot -7 = 21$). Alice receives $B = 21$ and computes $s = B^a \bmod p = 21^3 \bmod 23 = -2^3 \bmod 23 = -8 \bmod 23 = 15$. Bob receives $A = 10$ and computes $s = A^b \bmod p = 10^{13} \bmod 23 = 10 \cdot (-8)^6 \bmod 23 = 10 \cdot 2^{18} \bmod 23 = 10 \cdot (-5)^3 = -10 \cdot 5 \cdot 2 \bmod 23 = 15 \bmod 23$ using similar tricks. So both share the key $s = 15$.

For part (b), the 2DHKE protocol runs two DHKE exchanges in parallel. Alice chooses two independent secret exponents $a_1, a_2 \in \mathbb{Z}_p$, and sends $A_1 = g^{a_1}$ and $A_2 = g^{a_2}$ to Bob. Bob chooses $b_1, b_2 \in \mathbb{Z}_p$ and sends $B_1 = g^{b_1}$ and $B_2 = g^{b_2}$. Alice computes shared keys $K_1 = B_1^{a_1}$ and $K_2 = B_2^{a_2}$ and outputs

$K = K_1 \| K_2$. Bob computes $K_1 = A_1^{b_1}$ and $K_2 = A_2^{b_2}$ and also outputs $K = K_1 \| K_2$.

For part (c), we argue that 2DHKE cannot be more secure than DHKE. Suppose there exists an adversary $\mathcal{A}$ that breaks DHKE, meaning $\mathcal{A}$ can distinguish the key output in DHKE from random. We can use $\mathcal{A}$ to build an adversary $\mathcal{B}$ against 2DHKE. Specifically, $\mathcal{B}$, on input a transcript (A_1, A_2, B_1, B_2) and a key $K_1 \| K_2$, runs $\mathcal{A}$ on input (A_1, B_1) and K_1. Then $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. If $K_1 \| K_2$ is the real 2DHKE key belonging to transcript (A_1, A_2, B_1, B_2), then K_1 is the real DHKE key belonging to transcript (A_1, B_1). If, on the other hand, $K_1 \| K_2$ is uniformly random then of course K_1 is uniformly random. $\mathcal{B}$ thus has the same distinguishing advantage against 2DHKE as $\mathcal{A}$ has against DHKE.

For part (d), we observe that for any distinguisher $\mathcal{D}$ such that

$$\left| \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}}] - \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}}] \right| \geq \varepsilon$$

we have that

$$\begin{aligned}
\varepsilon &\leq \left| \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}}] - \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}}] \right| \\
&= \left| \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}}] - \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{int}}}] + \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{int}}}] - \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}}] \right| \\
&\leq \left| \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}}] - \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{int}}}] \right| + \left| \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{int}}}] - \Pr[1 \leftarrow \mathcal{D}^{\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}}] \right|
\end{aligned}$$

using the triangle inequality. If both summands on the right hand side where smaller than $\varepsilon/2$, their sum would be smaller than ε, so we have that one of the two summands on the right hand side is larger or equal $\varepsilon/2$.

Now, note that distinguishing either $\mathcal{L}^{\text{2DHKE}}_{\text{ka-real}}$ or $\mathcal{L}^{\text{2DHKE}}_{\text{ka-rand}}$ from $\mathcal{L}^{\text{2DHKE}}_{\text{int}}$ is equivalent to distinguishing DHKE real key from random in one of the two parallel executions of DHKE, so an adversary against 2DHKE can be used to build an adversary against DHKE. Thus, security of DHKE implies security of 2DHKE.