Sample Solution

MIDTERM EXAM

Name: *Matriculation:*

- **DO NOT OPEN** the exam until instructed to do so. Read all the instructions first.
- The exam consists of **15 sheets** of paper. The last page ends with the sentence "This is the last page of your exam."
- The exam is closed-book, which means there are no allowed aids for this exam. We added relevant definitions on the last page of this exam.
- Use a pen with **permanent black or blue** ink. **Under no circumstances use a pencil or red or green ink.**
- The exam consists of four problems to solve. You are supposed to solve **all four** problems. The problems do not give equal points, so make sure you do not spend too much time on problems that only give a small number of points.
- You can achieve at most 40 points for this exam.
- You have exactly **90 minutes** for the exam. Use your time wisely.
- Write your solutions in the space provided beneath each subproblem. You may use the empty reverse sides of the exam if you run out of space. If you choose to do so **clearly indicate which problem the solution belongs to**. If you do not indicate clearly the problem you are solving, your solution will not be graded.
- **Be neat and write legibly**. You will be graded not only on the correctness of your answer but also on the clarity with which you express it. In particular, if you are asked to show (both prove or disprove) a statement, the closer you can come to a (correct) formal statement, the better.
- Make sure your mobile phone(s) are switched off. Calculators must not be used.
- Please place your student identification card on your desk.
- Write your name and your matriculation on each page in the corresponding fields.
- Do not fill out the table below.
- Good luck!

Problem	**1**	**2**	**3**	**4**	**Total**
Points Possible	19	7	7	7	40
Points Achieved					

I hereby confirm that I have carefully read and understood the above instructions.

Signature

Problem 1 (Multiple Choice) 19 Points

For each question, choose the letter corresponding to the statement that fits best and place it in the appropriate table below. **Each question has only one correct answer.** You will earn 1 point for each correct answer. **For each incorrect answer, 1 point will be deducted from your total score for this problem.** If you leave a question unanswered, no points will be deducted. The total score for this problem cannot be less than 0.

Question	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)	(ix)	(x)
Choice										
Solution	b	c	c	a	b	b	b	c	a	b

Question	(xi)	(xii)	(xiii)	(xiv)	(xv)	(xvi)	(xvii)	(xviii)	(xix)
Choice									
Solution	b	b	c	d	b	d	b	a	b

(i) What is the central idea behind proving the security of a scheme via a reduction?

a) Construct a simulator that mimics all adversarial actions and show that the adversary cannot distinguish it from the real scheme under any input distribution.

b) Assume an efficient adversary breaks the scheme and transform this adversary into an algorithm that violates an underlying hardness assumption.

c) Show that every attack on the scheme can be mapped to an indistinguishable execution in the random oracle model, ensuring the adversary learns no new information.

d) Prove that for any adversary, its advantage can be made negligible by restricting the adversary's access to certain oracle queries.

(ii) Which of the following protocols most closely fits the definition of a Fully Encrypted Protocol (FEP), meaning that both payload and protocol metadata are deliberately obfuscated to resist traffic classification?

a) HTTPS, because it encrypts payloads but still exposes handshake and header metadata.

b) WireGuard, because it encrypts all packets but retains highly recognizable packet-size and timing patterns.

c) OBFS4 is preferred because it deliberately hides protocol fingerprints, making the traffic appear as random, high-entropy data.

d) TLS 1.3, because it encrypts more of the handshake than previous versions.

(iii) In many public-key encryption schemes, why is randomness essential during encryption?

a) To ensure that the ciphertext grows linearly with the length of the message.

b) To prevent an adversary from predicting the public key used in each encryption.

c) To guarantee that encrypting the same plaintext under the same public key produces different ciphertexts, preventing deterministic leakage.

d) To allow the receiver to reconstruct the secret key during decryption.

(iv) Why was RC4 eventually prohibited in modern versions of TLS, despite being a stream cipher that initially avoided certain CBC-mode attacks?

a) Because practical attacks exploited statistical biases in the RC4 keystream, allowing partial plaintext recovery even over HTTPS.

b) Because RC4 required keys longer than those allowed by the TLS specification.

c) Because RC4 could not be combined with server authentication in the TLS handshake.

d) Because RC4 produced ciphertexts that were incompatible with TLS session resumption.

(v) What is the significance of AEAD in TLS 1.3?

a) It replaces the need for public key encryption.

b) It provides both confidentiality and integrity for transmitted data.

c) It allows to achieve post-quantum security, since it is immune to Shor's algorithm.

d) It provides integrity, but no confidentiality.

(vi) Why did the Heartbleed vulnerability in OpenSSL pose a severe threat to long-term TLS security, beyond simply leaking random server memory?

a) Because it allowed attackers to bypass certificate validation entirely and impersonate any server.

b) Because leaked memory could contain private keys, enabling retroactive decryption of past TLS sessions that did not use forward secrecy.

c) Because Heartbleed allowed modification of the server's cipher suite preferences during the handshake.

d) Because the vulnerability enabled an attacker to inject malicious code into the OpenSSL library at runtime.

(vii) Why do security proofs for cryptographic protocols often fail to prevent real-world attacks?

a) Because provable security always assumes that adversaries have unlimited computational power, which does not match real-world constraints.

b) Because security proofs typically reason about idealized models and assumptions, while real-world systems introduce side channels, implementation bugs, and protocol interactions not captured in those models.

c) Because real-world adversaries rarely target cryptographic components and instead focus exclusively on social engineering.

d) Because proofs only apply to symmetric-key schemes, whereas most real-world systems rely on public-key cryptography.

(viii) What is the role of HSTS in securing TLS connections?

a) It validates certificates during the handshake.

b) It encrypts session cookies for confidentiality.

c) It enforces HTTPS connections to prevent downgrade attacks.

d) It prevents Heartbleed-style memory leaks.

(ix) What is one of the primary challenges in integrating post-quantum security into TLS?

a) Post-quantum key exchange algorithms often have larger key sizes or message sizes, which can increase handshake latency and strain existing network infrastructure.

b) Hash-based signature schemes cannot be used in TLS because they provide no mechanism for certificate-based authentication.

c) Quantum-resistant algorithms require trusted hardware support, which most TLS endpoints do not provide.

d) Classical cipher suites must be removed entirely before any post-quantum mechanisms can be deployed safely.

(x) In TLS, why is a hybrid of asymmetric and symmetric cryptography used instead of relying on a single type of encryption?

a) Because using both types of encryption automatically doubles the security level of the connection.

b) Because symmetric encryption alone cannot authenticate the server, and asymmetric encryption alone would be too slow for bulk data.

c) Because symmetric keys cannot be generated securely without also encrypting them with an RSA public key.

d) Because asymmetric encryption is only secure when combined with at least one symmetric cipher.

(xi) Why is hybrid encryption commonly used instead of relying solely on public-key encryption?

a) Symmetric encryption is easier to make quantum-resistant.

b) Public-key encryption is more computationally expensive than symmetric encryption, so it is only used to encrypt a symmetric key.

c) Public-key encryption is more computationally expensive than symmetric encryption, so it is only used to encrypt a small portion of the plaintext.

d) Because combining public-key and symmetric encryption results in more secure encryption schemes.

(xii) Which of the following is necessary for a private-key encryption scheme to be CPA-secure?

a) The encryption must be deterministic to ensure consistent output given the public key and the message.

b) The encryption must be randomized so that the same message encrypts to different ciphertexts.

c) The encryption must be randomized, to obfuscate the plaintext through a uniformly random mask.

d) The encryption algorithm must use a secure pseudorandom function.

(xiii) In a CPA security proof, why is the adversary allowed to query the encryption oracle both before and after receiving the challenge ciphertext?

a) Because the adversary must help the challenger test decryption correctness.

b) To ensure the adversary learns the key distribution.

c) To model realistic settings where attackers can keep interacting with the system while attempting to distinguish ciphertexts.

d) Because the challenger needs these queries to decide the challenge bit.

(xiv) Which of the following best describes why perfect security (e.g., One-Time Pad) is impractical for general communication?

a) Perfectly secure schemes such as One-Time Pad cannot be formally proven secure.

b) Symmetric keys cannot be generated randomly in practice.

c) Perfect security provides no authentication guarantees.

d) They require secret keys as long as the message and must never reuse keys.

(xv) What does the Decisional Diffie-Hellman (DDH) assumption state?

a) It is computationally infeasible to compute g^{ab} given g^a and g^b, with uniformly random a and $b \in \mathbb{Z}_p$.

b) It is difficult to distinguish g^{xy} from g^z in a cyclic group for uniformly random $x, y, z \in \mathbb{Z}_p$.

c) It ensures that g^x and g^y are indistinguishable from random strings for uniformly random $x, y \in \mathbb{Z}_p$.

d) It assumes g^x and g^{xy} are equivalent for uniformly random $x, y \in \mathbb{Z}_p$.

(xvi) Why must a cryptographic security definition explicitly describe the adversary's capabilities?

a) So that encryption keys can be automatically generated based on the adversary's strength.

b) So that the adversary can be forced to attack in a predictable manner.

c) To ensure the scheme remains compatible with classical and quantum attackers.

d) Because the strength of the security guarantee depends on what the adversary can observe, access, or influence.

(xvii) What is the purpose of defining specific security goals for a cryptographic scheme?

a) To allow cryptographers to reuse security proofs between unrelated schemes.

b) To precisely state what the scheme should protect against, enabling meaningful evaluation of whether a construction meets those goals.

c) To limit the adversary's ability to adapt during an attack.

d) To ensure that performance is always prioritized over security.

(xviii) Why do we model collision-resistant hash functions as keyed functions?

a) Because a function that is compressing and not keyed admits a constant-time collision-finder.

b) Because the key is necessary to compute the function.

c) Because they are a subset of PRFs, and the latter are keyed.

d) Because the randomness of the key ensures unpredictable outputs.

(xix) Which feature contributes most to the Schnorr scheme's *provable security*?

a) Its efficiency in very short signature sizes.

b) Its reduction to a well-studied hardness assumption like DLP.

c) Its ability to operate without random nonces.

d) Its compatibility with any cryptographic hash function in practice.

Problem 2 (Pseudorandom Functions) 7 Points

In this exercise, we will consider two variants of pseudorandom functions:

- F is a **selectively secure PRF** if every PPT adversary $\mathcal{A}$ with oracle access to $F(k, \cdot)$ (for uniformly random k) that *generates all of its queries before seeing any of the oracle's outputs* cannot distinguish it from being given oracle access to a uniformly random function $f(\cdot)$.
- Let f be a uniformly random function, and $x_1, \ldots, x_q$ be q uniformly random and independent bitstrings. F is a **weak PRF** if every PPT adversary $\mathcal{A}$ cannot distinguish

$$x_1, F(k, x_1), \ldots, x_q, F(k, x_q) \quad \text{from} \quad x_1, f(x_1), \ldots, x_q, f(x_q),$$

with k sampled uniformly at random.

In both cases, ”cannot distinguish” is implicitly followed by ”with non-negligible advantage”. For your convenience, we added the precise definitions at the end of this exam.

(i) Are weak PRFs a superset of selective PRFs? Prove or disprove it. (2 Points)

Solution:
Weak PRFs are a superset of selective PRFs. That is, if F is selectively secure, then it is a weak PRF too.

Assume by contradiction that there exists a polynomial-bounded q and a PPT adversary $\mathcal{A}$ such that for randomly generated $x_1, \ldots, x_q$, $\mathcal{A}$ distinguishes $\{x_i, F(k, x_i)\}_{i\in[q]}\}$ from $\{x_i, f(x_i)\}_{i\in[q]}\}$ with probability $1/2 + \mu(\lambda)$, and μ is a non-negligible function. Then, we can immediately build the following adversary $\mathcal{A}'$:

1 Points: Correct intuition.

a) $\mathcal{A}'$ has oracle access to either $F(k, \cdot)$ or $f(\cdot)$, and runs $\mathcal{A}$ as a subroutine.

b) $\mathcal{A}'$ generates q random inputs $x_1, \ldots, x_q$, and queries its oracle on said inputs.

c) It then relays both the inputs and corresponding outputs to $\mathcal{A}$, and collects its guess bit b'.

d) Finally, it outputs b'.

$\mathcal{A}'$ perfectly simulates the security game defining weak pseudorandomness towards $\mathcal{A}'$, therefore b' is the correct answer with probability $1/2 + \mu(\lambda)$. Furthermore, since $\mathcal{A}$ is efficient by assumption, so is $\mathcal{A}'$.

1 Points: Correct reduction.

(ii) Are selective PRFs a superset of weak PRFs? Prove or disprove it. (**Hint:** what happens if you take a PRF F, and you modify it by fixing one output to...?) (2 Points)

Solution:
No. We can show this by constructing a PRF that is weak pseudorandom, but not selectively secure.

Let $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a pseudorandom function. Define a new function

1 Points: Correct intuition.

$$G(k, x) = \begin{cases} 0 & x = 0 \\ F(k, x) & x \neq 0 \end{cases}.$$

G is clearly NOT selectively secure: you can just query it on input 0, and check if the corresponding output is 0. This is always true for G, but happens with negligible probability when interacting with a random function.

On the other hand, the function is weak pseudorandom, since when generating inputs $x_1, \ldots, x_q$ at random, the probability of hitting 0 is negligible. Conditioned on this event, evaluating G causes to evaluate F on q random points, and the result is in turn pseudorandom.

1 Points: Exhibits counterexample.

(iii) Consider the private-key CPA-secure encryption scheme that you have seen during the lecture. That is:

- $\mathsf{KGen}(1^\lambda)$: Return a key $k \leftarrow \{0,1\}^\lambda$ of size λ chosen uniformly at random.
- $\mathsf{Enc}(k, m)$: Choose $r \leftarrow \{0,1\}^\lambda$ uniformly at random and compute

$$c = (r, s) = (r, F(k, r) \oplus m).$$

- $\mathsf{Dec}(k, c)$: Parse c as (r, s) and output the plaintext message computing

$$m = F(k, r) \oplus s.$$

Said scheme uses a PRF. What happens if you replace F with a *weak* PRF? State whether the scheme remains secure or not, and explain why. Provide a high-level intuition and a proof sketch or a counterexample. To assist you, we provide the corresponding definition at the end of the exam. (3 points)

Solution:
Yes, the scheme remains secure. This is because F is only evaluated over uniformly random inputs, and by its weak pseudorandomness the output is still pseudorandom.

1 Points: Correct Intuition

Concretely, we can prove the claim as follows:

a) We observe that a PRF and a weak PRF are computationally indistinguishable as long as they are evaluated over uniformly random inputs only. This is due to to both of them being computationally indistinguishable from the same object (a uniformly random function).

b) Then, suppose that the encryption scheme is not CPA-secure with a weak PRF. Then, there must exist a PPT adversary $\mathcal{A}$ with non-negligible advantage in the corresponding game. We can leverage $\mathcal{A}$ to construct a distinguisher $\mathcal{B}$, which can tell apart PRFs and weak PRFs. $\mathcal{B}$ simulates the CPA game towards $\mathcal{A}$, and outputs 1 iff $\mathcal{A}$ wins the simulated game.

2 Points: Correct Sketch

Problem 3 (Schnorr Signatures) 7 Points

Consider the Schnorr Signature scheme that you have seen in the lecture (whose pseudocode is available at the end of this exam). Signer Alice has poor entropy sources, and decided to issue two signatures with equal nonce (R, s_1) and (R, s_2) on two messages $m_1 \neq m_2$.

NOTE: for the purpose of this exercise, assume that the hash function H has the following property: if two inputs x, y are distinct, then $\mathsf{H}(x) = \mathsf{H}(y)$ with probability $1/p$.

(a) In a few sentences explain your intuition whether this constitutes a vulnerability. If no, explain why the scheme remains unforgeable. If yes, exhibit an attack. To assist you, we added the definition of existential unforgeability at the end of this exam. (3 Points)

Solution:
It is not secure. By construction, we know that $R = g^r$, $s_1 \equiv \mathsf{sk}c_1 + r \mod p$ and $s_2 \equiv \mathsf{sk}c_2 + r \mod p$. Their difference is then

$$s_1 - s_2 \equiv \mathsf{sk}(c_1 - c_2) \mod p.$$

With probability $1 - p^{-1}$, $c_1 \neq c_2$. Then, in particular, $c_1 - c_2$ is invertible modulo p, allowing to retrieve

$$\mathsf{sk} \equiv (c_1 - c_2)^{-1}(s_1 - s_2) \mod p.$$

1 Points: correct intuition

2 Points: correct attack

(b) Alice decides to compensate for her poor entropy sources as follows. It generates a short key k to a PRF F. Instead of generating r uniformly at random, it instead computes $r \leftarrow F(k, \mathsf{vk})$, where vk is the public key. Is the resulting scheme unforgeable? If yes, explain why; otherwise, exhibit an attack. (2 Points)

Solution:
Because vk is constant, the nonce r would be constant too. Therefore, Alice ends up using the same nonce all the time, thus enabling the previous attack.

Concretely, an attacker can request a signature on two distinct messages m_1, m_2, and obtain two signatures with equal nonce as before. Because $m_1 \neq m_2$, we can also assume that $c_1 \neq c_2$ with overwhelming probability. The rest of the attack is identical.

1 Points: Correct intuition.

1 Points: Describe attack

(c) Suppose that $f : \mathbb{G} \to \mathbb{Z}_p$ is an efficiently computable bijective function. Alice proposes one more strategy to save entropy:

a) It samples an initial $s \leftarrow\$ \mathbb{Z}_p$, and publishes $h \leftarrow g^s$.

b) At each signature, it samples $r \leftarrow \mathbb{Z}_p$.

c) Alice computes

$$(\alpha, \beta) = (g^r, h^r)$$

and obtains two nonces $r_1 = f(\alpha)$, $r_2 = f(\beta)$.

d) It then uses r_1 for one signature, and r_2 for the next one.

```
$\mathsf{KGen}(\lambda)$
    $\mathsf{sk} \leftarrow\!\!\$\, \mathbb{Z}_p$
    $\mathsf{vk} \leftarrow g^{\mathsf{sk}}$
    return $(\mathsf{sk}, \mathsf{vk})$

$\mathsf{Sign}'(\mathsf{sk}, m_1, m_2)$
    $r \leftarrow\!\!\$\, \mathbb{Z}_p$
    $(\alpha, \beta) = (g^r, h^r)$
    $r_1 \leftarrow f(\alpha), r_2 \leftarrow f(\beta)$
    $R_1 \leftarrow g^{r_1}$
    $R_2 \leftarrow g^{r_2}$
    $h_1 \leftarrow \mathsf{H}(\mathsf{vk}, R_1, m_1)$
    $h_2 \leftarrow \mathsf{H}(\mathsf{vk}, R_2, m_2)$
    $s_1 \leftarrow \mathsf{sk} \cdot h_1 + r_1$
    $s_2 \leftarrow \mathsf{sk} \cdot h_2 + r_2$
    return $((R_1, s_1), (R_2, s_2))$

$\mathsf{Vrfy}'(\mathsf{vk}, m_1, m_2, \sigma)$
    $(\sigma_1, \sigma_2) := \sigma$
    return $\mathsf{Vrfy}(\mathsf{vk}, m_1, \sigma_1) \wedge$
    $\mathsf{Vrfy}(\mathsf{vk}, m_2, \sigma_2)$
```

Figure 1: Tweaked Schnorr Signature.

For completeness, we provide the resulting scheme in Figure 1. Is the resulting scheme unforgeable? Provide your intuition. (2 Points)

Solution:
It is. First of all, observe that under the DDH assumption, α and β are pseudorandom. Furthermore, since f is bijective, r_1 and r_2 are two uniformly random elements of $\mathbb{Z}_p$.

It follows that this computationally indistinguishable from running two Schnorr signatures in a row where each time the exponent r is sampled uniformly at random. The latter is known to be unforgeable, which implies unforgeability for the tweaked scheme.

1 Points: Pseudorandomness of nonces

1 Points: Correct conclusion

Problem 4 (A Shared Secret With Picky Parties) 7 Points

We assume two parties, A and B, who want to compute a shared secret based on the hardness of Decisional Diffie-Hellman (DDH) problem. Both of them are, for some reason, obsessed with even integers. This causes them to only sample even exponents, as in the figure below[1].

[1] Recall: if $\mathbb{G}$ is a cyclic group generated by g, $[x] := g^x$.

```
A(λ)
  $x \leftarrow\!\!\$\ \mathbb{Z}_p \cap \{x \in \mathbb{Z} | x \equiv 0 \mod 2\}$
  send $[x]$ to B
  receive $[y]$ from B
  return $[xy]$
```

```
B(λ)
  $y \leftarrow\!\!\$\ \mathbb{Z}_p \cap \{x \in \mathbb{Z} | x \equiv 0 \mod 2\}$
  send $[y]$ to A
  receive $[x]$ from A
  return $[xy]$
```

(a) In a few sentences, explain your intuition about whether the resulting key exchange protocol is secure in the presence of an eavesdropper. (2 Points)

Solution:
It is secure. The original DDH key exchange is secure when sampling uniformly random exponents, and the latter are even with "good" probability. If there was an adversary that can break the tweaked scheme above, then we could recycle it to break the original scheme.

1 Points: correct intuition

1 Points: presents a valid argument

(b) Provide a formal proof to support your intuition. I.e., either provide a reduction to the hardness of DDH or give an attack showing that an eavesdropper can distinguish random keys from real ones. In either case, analyze the efficiency and the success probability of either your attack or your reduction. As an aid, we added the definitions of eavesdropper security and DDH to the last page of this exam. (4 Points)

Solution:
We assume towards contradiction that the group key-exchange protocol is not secure in the presence of an eavesdropper. I.e., we assume there exists a PPT distinguisher $\mathcal{A}$ for which

$$|\Pr[\mathcal{A}^0_{GKE}(\lambda) = 1] - \Pr[\mathcal{A}^1_{GKE}(\lambda) = 1]| \geq \epsilon(\lambda) \tag{1}$$

for some non-negligible function ϵ.

We use $\mathcal{A}$ to construct a distinguisher $\mathcal{B}$ against DDH as follows:

a) $\mathcal{B}$ collects $[x]$ and $[y]$, and then the secret $[k]$.

b) It forwards all of them to $\mathcal{A}$, and collects its decision bit b'.

c) Finally, it outputs b'.

1 Points: description of reduction

To check efficiency, we note that $\mathcal{B}$ merely forwards group elements to $\mathcal{A}$, and runs the latter without performing any postprocessing. Since $\mathcal{A}$ is PPT, $\mathcal{B}$ is efficient too.

1 Points: analysis of runtime

We now analyze the success probability of $\mathcal{B}$. We first observe that the probability of sampling an even integer in the range $[0, p-1]$ is $(p+1)/2p \geq 1/2$. Therefore, the probability that both x and y are even is more than $1/4$. Conditioned on this event, $\mathcal{B}$ perfectly simulates towards $\mathcal{A}$ the interaction with A and B in the tweaked scheme. Therefore, the success probability is at least

$$\frac{1}{4}\left(\frac{1}{2}+\epsilon(\lambda)\right)+\frac{3}{4}\cdot\frac{1}{2}=\frac{1}{2}+\frac{\epsilon(\lambda)}{4}.$$

This yields an advantage $\frac{\epsilon(\lambda)}{4}$ that is non-negligible. As this contradicts the assumption that DDH is hard to solve, $\mathcal{A}$ cannot exist and the tweaked key-exchange protocol is secure in the presence of eavesdroppers. This concludes the proof.

2 Points: analysis of advantage

(c) A and B decide to go one step further. Instead of sticking to multiples of 2, they become much more selective. Specifically, suppose $p \equiv -1 \mod c$ for a *small* constant c. A samples x uniformly at random from the set $S = \mathbb{Z}_p \cap \{x \in \mathbb{Z} | x \equiv 0 \mod (p+1)/c\}$, and B does the same. Is the scheme secure? Explain your intuition. (1 Point)

Solution:
No, it is not. The set S has cardinality

$$|S| \leq \frac{c(p+1)}{p} = c\cdot(1+1/p).$$

Since c is constant, $|S|$ is so small that an adversary can just guess which exponents A and B have sampled with non-negligible probability.

1 Points: Correct argument

Auxiliary Information – This is the last page of your exam.

In this section, we provide definitions needed to solve problems two to four.

Definition 1 (Pseudorandom Functions)**.** Let F be an efficient, keyed function. F is a *pseudorandom function (PRF)* if for all PPT distinguishers D, there exists a negligible function negl such that

$$\left|\Pr\left[D^{F(k,\cdot)}(1^\lambda)=1\right]-\Pr\left[D^{f(\cdot)}(1^\lambda)=1\right]\right| \leq \mathsf{negl}(\lambda),$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^\lambda$ and the randomness of D, and the second probability is taken over uniform choice of f and the randomness of D.

```
$\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(\lambda)$
$k \leftarrow \mathsf{KGen}(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Enc}(k,\cdot)}, \quad |m_0| = |m_1|$
$b \leftarrow \{0,1\}$
$c_b \leftarrow \mathsf{Enc}(k, m_b)$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}(k,\cdot)}(c_b)$
if $b' = b$ return $1$
else return $0$
```

Definition 2 (Selectively Secure PRFs)**.** Let F be an efficient, keyed function. F is a *selectively secure pseudorandom function (PRF)* if for all PPT distinguishers D that fix their queries before interacting with their oracle, there exists a negligible function negl such that

$$\left|\Pr\left[D^{F(k,\cdot)}(1^\lambda) = 1\right] - \Pr\left[D^{f(\cdot)}(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^\lambda$ and the randomness of D, and the second probability is taken over uniform choice of f and the randomness of D.

Definition 3 (Weak Pseudorandom Functions)**.** Let F be an efficient, keyed function. F is a *weak pseudorandom function (PRF)* if for all PPT distinguishers D and all polynomial-bounded q, there exists a negligible function negl such that

$$|\Pr[D(x_1, F(k, x_1), \ldots, x_q, F(k, x_q)) = 1] - \Pr[D(x_1, f(x_1), \ldots, x_q, f(x_q) = 1]| \leq \mathsf{negl}(\lambda),$$

where $x_1, \ldots, x_q$ are uniformly random and independent bitstrings, the first probability is taken over uniform choice of $k \in \{0,1\}^\lambda$ and the randomness of D, and the second probability is taken over uniform choice of f and the randomness of D.

Definition 4 (CPA security)**.** An encryption scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ over a message space $\mathcal{M}$ has *indistinguishable encryptions under a chosen-plaintext attack* (or is *CPA-secure*) if for every PPT adversary $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Definition 5 (Existential Unforgeability)**.** A signature scheme $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is existentially unforgeable under a chosen-message attack if for each PPT adversary $\mathcal{A}$ there exists a negligible function negl, such that

$$|\Pr[\mathsf{EUF} - \mathsf{CMA}_{\mathcal{A},\Sigma}(\lambda) = 1]| \leq \mathsf{negl}(\lambda).$$

```
$\mathsf{EUF} - \mathsf{CMA}_{\mathcal{A},\Sigma}(\lambda)$
$Q := \emptyset$
$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk},\cdot)}(\mathsf{vk})$
$b^* = \mathsf{Vrfy}(\mathsf{vk}, m^*, \sigma^*) \wedge ((m^*, \cdot) \notin Q)$
return $b^*$
```

```
Oracle $\mathcal{O}(\mathsf{sk}, m)$
$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
$Q := Q \cup \{(m, \sigma)\}$
return $\sigma$
```

Name: *Matriculation:*

```
$\mathsf{Setup}(\lambda)$
    $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\lambda)$
    $par := (\mathbb{G}, q, g)$
    return $par$

$\mathsf{KGen}(\lambda)$
    $\mathsf{sk} \leftarrow\!\!\$\ \mathbb{Z}_p$
    $\mathsf{vk} \leftarrow g^{\mathsf{sk}}$
    return $(\mathsf{sk}, \mathsf{vk})$

$\mathsf{Sign}(\mathsf{sk}, m)$
    $r \leftarrow\!\!\$\ \mathbb{Z}_p$
    $R \leftarrow g^r$
    $h \leftarrow \mathsf{H}(\mathsf{vk}, R, m)$
    $s \leftarrow \mathsf{sk} \cdot h + r$
    return $(R, s)$

$\mathsf{Vrfy}(\mathsf{vk}, m, \sigma)$
    $(R, s) := \sigma$
    $h \leftarrow \mathsf{H}(\mathsf{vk}, R, m)$
    return $g^s \overset{?}{=} R \cdot \mathsf{vk}^h$
```

Figure 2: Schnorr Signature Scheme.

Definition 6 (Eavesdropper Security of Key-Exchange Protocols)**.** A key-exchange protocol Π is *secure in the presence of an eavesdropper* if for every PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\Pr[\mathsf{GKE}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1] \leq \tfrac{1}{2} + \mathsf{negl}(\lambda).$$

```
$\mathsf{GKE}^1_{\mathcal{A},\Pi}(\lambda)$
    $(trans, k) \leftarrow \langle \mathsf{A}(1^\lambda), \mathsf{B}(1^\lambda), \mathsf{C}(1^\lambda) \rangle$
    $([\mathsf{sk_A}], [\mathsf{sk_B}], [\mathsf{sk_C}]) \leftarrow trans$
    $k' \leftarrow\!\!\$\ \{0,1\}^\lambda$
    $b' \leftarrow \mathcal{A}(k', [\mathsf{sk_A}], [\mathsf{sk_B}], [\mathsf{sk_C}])$
    return $b' == b$

$\mathsf{GKE}^0_{\mathcal{A},\Pi}(\lambda)$
    $(trans, k) \leftarrow \langle \mathsf{A}(1^\lambda), \mathsf{B}(1^\lambda), \mathsf{C}(1^\lambda) \rangle$
    $([\mathsf{sk_A}], [\mathsf{sk_B}], [\mathsf{sk_C}]) \leftarrow trans$
    $k' \leftarrow k$
    $b' \leftarrow \mathcal{A}(k', [\mathsf{sk_A}], [\mathsf{sk_B}], [\mathsf{sk_C}])$
    return $b' == b$
```

Definition 7 (Decisional Diffie-Hellman)**.** The *Decisional Diffie-Hellman (DDH) problem* is *hard relative to* Gen if for all PPT algorithms $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathcal{A}(\mathbb{G}, q, [1], [x], [y], [z]) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, [1], [x], [y], [xy]) = 1] \leq \mathsf{negl}(\lambda),$$

where in each case the probabilities are taken over the experiment in which the group generation algorithm $\mathsf{Gen}(1^\lambda)$ outputs $(\mathbb{G}, q, [1])$, and then uniform $x, y, z \in \mathbb{Z}_q$ are chosen.