Sample Solution

- **DO NOT OPEN** the exam until instructed to do so. Read all the instructions first.
- The exam consists of **11 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

Name: *Matriculation:*

Problem 1 (Multiple Choice) 19 Points

For each of the following statements, mark the statement that fits most. Each question has only one correct answer. For each correct answer you will get 1 point, **for each incorrect answer 1 point will be deducted from your total for this problem**. If you do not answer a question, no points will be deduced. You cannot receive less than 0 points for the problem overall.

(i) Why is it important to secure DNS queries with encryption protocols like DoT or DoH?

☐ DNS queries contain sensitive data, such as the contents of encrypted messages.

☐ DNS queries are the only part of the internet communication chain visible to end users.

☒ Unencrypted DNS queries expose visited domains to intermediaries, enabling tracking and censorship.

☐ Without encryption, DNS servers are unable to process requests reliably.

(ii) What is the primary privacy concern with rerouting attacks, even when HTTPS encryption is used?

☐ Rerouting attacks completely decrypt HTTPS traffic during transit.

☒ They expose metadata such as destination servers, timing, and traffic volume.

☐ Rerouting attacks require cooperation from the victim's device to succeed.

☐ HTTPS encrypts routing metadata, so rerouting attacks have no impact on privacy.

(iii) What is the fundamental privacy risk associated with DNS even when DNS over HTTPS (DoH) is used?

☐ The use of encryption inherently slows down DNS resolution, causing delays.

☒ DNS over HTTPS cannot prevent data leaks at the recursive resolver level.

☐ The initial DNS queries reveal the entire browsing history to ISPs.

☐ DoH requires public DNS servers, which are inherently insecure.

(iv) Why is the implementation of Privacy-Enhancing Technologies (PETS) often seen as challenging for widespread adoption?

☐ PETS technologies do not align with legal privacy standards in most countries.

☒ Many PETS tools require advanced technical skills and infrastructure, creating usability barriers.

☐ PETS are incompatible with modern encryption standards such as TLS.

☐ The adoption of PETS would require entirely redesigning the current internet architecture.

Name: *Matriculation:*

(v) In the CPA-Secure Encryption scheme from PRFs, what guarantees the randomness of the ciphertext?

☐ The use of a key generator that outputs random keys.

☒ A uniformly random r value that is input in a PRF to blind the message.

☐ Encryption using AES in CBC mode.

☐ The deterministic nature of PRFs.

(vi) How can HSTS fingerprinting track users even after they clear their cookies?

☐ By using malicious JavaScript injections to rewrite browser configurations.

☐ By persisting HSTS settings in the browser's local storage indefinitely.

☒ By checking cached HSTS states for predefined domains in the browser.

☐ By modifying the browser's TLS certificate chain to create a unique identifier.

(vii) What is a significant limitation of Fully Encrypted Protocols (FEPs) like obfs4?

☐ They are easily detected by modern deep packet inspection tools.

☐ They rely on DNS resolution, making them vulnerable to poisoning.

☒ They can only disguise the content, not the traffic volume.

☐ They require widespread adoption to be effective.

(viii) How does domain fronting evade censorship mechanisms?

☐ By modifying the source IP address during packet transmission.

☒ By routing requests through allowed domains while hiding the target domain in the HTTP header.

☐ By encrypting all DNS queries and responses to bypass filters.

☐ By spoofing the TLS handshake to appear as a trusted domain.

(ix) Why might privacy tools like ad blockers paradoxically increase the uniqueness of a user's fingerprint?

☐ They disable features that obfuscate the browser's default settings.

☒ They introduce distinctive patterns in blocked requests that can be tracked.

☐ They modify the browser's User-Agent string to include identifiable information.

☐ They rely on non-standard HTTP methods that are easily detected.

(x) In the Diffie-Hellman Key Exchange protocol, why is g^{xy} secure under the Decisional Diffie-Hellman assumption?

☐ Because the exponentiation is computationally intensive.

☒ Because it is computationally indistinguishable from a random element in the group.

☐ Because the values g^x and g^y are not revealed.

☐ Because g^{xy} does not depend on g^x or g^y.

Name: *Matriculation:*

(xi) What is a key difference between the DDH and DLog assumptions?

- ☐ DDH assumes the difficulty of computing discrete logarithms, while DLog assumes the indistinguishability of certain group elements.
- ☒ DDH assumes indistinguishability of group elements, while DLog assumes difficulty of computing discrete logarithms.
- ☐ DDH is harder to break than DLog.
- ☐ DLog is a decision problem while DDH is a computational problem.

(xii) In El Gamal encryption, why does correctness hold?

- ☒ Because the decryption algorithm exactly inverts the encryption steps using the private key.
- ☐ Because the ciphertext is generated using a deterministic process.
- ☐ Because the encryption scheme uses a secure PRF.
- ☐ Because $\frac{c_2}{c_1^x}$ does not depend on the randomness y.

(xiii) In the CPA-security experiment for public-key encryption, what ensures the indistinguishability of c_0 and c_1?

- ☐ The ciphertext is encrypted using a unique session key each time.
- ☐ The adversary is unable to access the private key.
- ☐ The encryption function is deterministic.
- ☒ The encryption function hides all information about the plaintext except its length.

(xiv) What is a primary limitation of the discrete logarithm assumption in certain groups?

- ☐ It is not defined for cyclic groups of prime order.
- ☐ It assumes that group elements are indistinguishable from random strings.
- ☒ In general, it does not hold in groups with composite orders.
- ☐ It cannot be used to prove the security of symmetric-key encryption schemes.

(xv) What is the primary purpose of the TLS Record Protocol?

- ☐ Negotiating cryptographic algorithms.
- ☐ Establishing a shared secret for secure communication.
- ☒ Encrypting and authenticating application data.
- ☐ Providing forward secrecy during key exchange.

(xvi) How does TLS 1.3 ensure forward secrecy?

- ☐ By using static RSA for key exchange.
- ☐ By encrypting handshake messages after ServerHello.
- ☒ By employing only ephemeral key exchange mechanisms.
- ☐ By applying AEAD ciphers to session resumption keys.

(xvii) Why is AES in its default mode not CPA-secure?

☐ It lacks sufficient key entropy.

☒ To achieve CPA security, it needs to be used in a mode of operation.

☐ It does not provide message integrity.

☐ It uses stream ciphers instead of block ciphers.

(xviii) How does TLS achieve message integrity in its record protocol?

☒ By using authenticated encryption.

☐ By using static RSA for authentication.

☐ By negotiating cipher suites during the handshake.

☐ By employing key encapsulation mechanisms.

(xix) How does Shor's algorithm threaten the security of TLS?

☐ It can decrypt AEAD ciphers.

☐ It reduces the effective key space of symmetric ciphers.

☒ It factors large integers used in RSA and ECC.

☐ It enables efficient brute-force attacks on AES.

Problem 2 (Customizing PRFs) 7 Points

Let $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a pseudorandom function. We use a slightly relaxed notion of pseudorandom function, where input, output, and key length are not necessarily equal. Consider the PRF candidate $F' : \{0,1\}^\lambda \times \{0,1\}^{\lambda-1} \to \{0,1\}^\lambda$ via

$$F'_k(x) := F_k(0\|x) \oplus F_k(1\|x).$$

(i) In a few sentences explain your intuition whether F' is also a PRF. (2 Points)

Solution:
The function is a PRF. For a reduction, it is easy to simulate the case of the PRF. In the case of a truly random function, the fact that the ranges of inputs for the two parts are disjoint ensures that there can be no correlation between them, making simulation trivial.

1 Points: correct intuition

1 Points: presents a valid argument

(ii) Provide a formal proof to support your intuition. I.e., either provide a reduction to an underlying PRF or give an attack showing that F' is no PRF. 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 PRFs to the last page of this exam. (5 Points)

Solution:
We assume towards contradiction that F' is not a PRF. I.e., we assume there exists a PPT distinguisher $\mathcal{A}$ for which

$$|\Pr[\mathcal{A}^{F'_k(\cdot)}(\lambda) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(\lambda) = 1]| \geq \epsilon(\lambda) \tag{1}$$

for some non-negligible function ϵ.

We use $\mathcal{A}$ to construct a distinguisher $\mathcal{B}$ against F as follows: On input 1^λ, $\mathcal{B}$ invokes $\mathcal{A}$ on 1^λ. For each x queried by $\mathcal{A}$, $\mathcal{B}$ queries $0\|x$ and $1\|x$ to its own oracle, receiving the answers y_0 and y_1. Then it combines the two as $y := y_0 \oplus y_1$ and returns y to $\mathcal{A}$. Eventually, $\mathcal{A}$ outputs a bit b and $\mathcal{B}$ outputs the same bit b.

2 Points: description of reduction

First, note that $\mathcal{B}$ runs the PPT algorithm $\mathcal{A}$ and answers a polynomial number of queries, using twice the number of queries. So, $\mathcal{B}$ is efficient.

1 Points: analysis of runtime

We now analyze the success probability of $\mathcal{B}$. In the case where the oracle is F_k, for each x, $\mathcal{B}$ returns the value $F_k(0\|x) \oplus F_k(1\|x)$. By construction of F', in this case, $\mathcal{B}$ therefore perfectly simulates the oracle for $\mathcal{A}$ for the case where its oracle is F'_k. Thus, we get

$$\Pr[\mathcal{B}^{F_k(\cdot)}(\lambda) = 1] = \Pr[\mathcal{A}^{F'_k(\cdot)}(\lambda) = 1]. \tag{2}$$

In the case where the oracle is a truly random function f, for each x, $\mathcal{B}$ returns the value $y := f(0\|x) \oplus f(1\|x)$. Observe that the domain of inputs for the first and second evaluation of f are disjoint. A random function that is restricted to a subset of its domain is still a random function, and because the domains are disjoint, the two functions are independent. Further, the XOR of two truly random functions is itself again a truly random function.

In this case, $\mathcal{B}$ therefore perfectly simulates the oracle for $\mathcal{A}$ for the case where its oracle is a truly random function. We thus get

$$\Pr[\mathcal{B}^{f(\cdot)}(\lambda) = 1] = \Pr[\mathcal{A}^{f(\cdot)}(\lambda) = 1]. \tag{3}$$

Combining (8) and (3) we can conclude that

2 Points: analysis of success

$$|\Pr[\mathcal{A}^{F'_k(\cdot)}(\lambda) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(\lambda) = 1]| = \tag{4}$$

$$|\Pr[\mathcal{B}^{F_k(\cdot)}(\lambda) = 1] - \Pr[\mathcal{B}^{f(\cdot)}(\lambda) = 1]| \geq \epsilon(\lambda). \tag{5}$$

As this contradicts the assumption that F is a PRF, $\mathcal{A}$ cannot exist and F' is also a PRF. This concludes the proof.

Problem 3 (Signing Large Messages) 7 Points

A limitation of many signature schemes is, that the size of the message to be signed is bounded. In setups like TLS, where we want to sign a whole transcript this could lead to problems. Hence, we define the following workaround to sign messages of arbitrary length. Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a fixed length signature scheme for message length λ. Let $\langle i \rangle$ denote the $\lambda/2$ bit encoding of the integer i[1]. We define the following new signature scheme $\Pi' = (\mathsf{KGen}, \mathsf{Sign}', \mathsf{Vrfy}')$ that is able to sign messages of length $\ell \cdot \lambda$ for arbitrary $\ell \in \mathbb{N}$ using the following construction.

```
$\mathsf{Sign}'(\mathsf{sk}, m)$
Parse $m$ as $(m_1, \ldots, m_{2\ell})$
    with $|m_i| = \lambda/2$
for $i \in \{1, \ldots, 2\ell\}$
    $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, \langle i \rangle || m_i)$
return $(\sigma_1, \ldots, \sigma_{2\ell})$
```

```
$\mathsf{Vrfy}'(\mathsf{vk}, m, \sigma)$
Parse $m$ as $(m_1, \ldots, m_{2\ell})$
    with $|m_i| = \lambda/2$
Parse $\sigma$ as $(\sigma_1, \ldots, \sigma_{2\ell})$
for $i \in \{1, \ldots, 2\ell\}$
    $b_i \leftarrow \mathsf{Vrfy}(\mathsf{vk}, \langle i \rangle || m_i, \sigma_i)$
return $b_1 \wedge \ldots \wedge b_{2\ell}$
```

(a) In a few sentences explain your intuition whether Π' is existentially unforgeable under a chosen message attack. (2 Points)

Solution:
It is not secure. A simple *mix and match* attack is possible, i.e., constructing a new signature from parts of two other signatures.

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 existential unforgeability of the underlying signature scheme or give an attack showing that Π' is not existentially unforgeable. In either case, analyze the efficiency and the success probability of either your attack or your reduction. As an aid, we added the definition of existential unforgeability of signature schemes to the last page of this exam. (5 Points)

[1] As an example, if we have $\lambda = 6$, then the number 3 is represented as 011 using this notation.

Solution:
We construct an adversary as follows: On input pk, $\mathcal{A}$ chooses two random messages $m_1 = m_1^1 \| \dots \| m_1^\ell$ and $m_2 = m_2^1 \| \dots \| m_2^\ell$ with $m_1^1 \neq m_2^1$. It queries both messages to the signing oracle resulting in signatures $\sigma_1 = (\sigma_1^1, \dots, \sigma_1^\ell)$ and $\sigma_2 = (\sigma_2^1, \dots, \sigma_2^\ell)$. Finally it outputs message $m := m_2^1 \| m_1^2 \| \dots \| m_1^\ell$ and signature $\sigma := (\sigma_2^1, \sigma_1^2, \dots, \sigma_1^\ell)$.

Clearly, $\mathcal{A}$ is efficient as it only chooses two messages and makes two queries. Further, because $m_1^1 \neq m_2^1$, the newly combined message has never been queried to the signing oracle, and according to the definition of the scheme, σ will clearly verify for m. Thus $\mathcal{A}$ succeeds with probability 1.

2 Points: description of adversary

1 Points: runtime

2 Points: success

Problem 4 (A Shared Secret Amongst Groups) 7 Points

So far, we have only considered cases where a single client wants to communicate privately with a single server. In this problem, we will extend this setup to a three-party case: We assume three parties, A, B, and C, want to compute a shared secret based on the hardness of Decisional Diffie-Hellman (DDH) problem. Below we outline the group key exchange mechanism between the three parties.

```
A(λ)
  $\mathsf{sk_A} \leftarrow\$ \mathbb{Z}_p$
  send $[\mathsf{sk_A}]$ to B and C
  receive $[\mathsf{sk_B sk_C}]$ from C
  return $[\mathsf{sk_A sk_B sk_C}]$
```

```
B(λ)
  $\mathsf{sk_B} \leftarrow\$ \mathbb{Z}_p$
  receive $[\mathsf{sk_A}]$ from A
  send $[\mathsf{sk_A sk_B}], [\mathsf{sk_B}]$ to C
  receive $[\mathsf{sk_A sk_C}]$ from C
  return $[\mathsf{sk_A sk_B sk_C}]$
```

```
C(λ)
  $\mathsf{sk_C} \leftarrow\$ \mathbb{Z}_p$
  receive $[\mathsf{sk_A}]$ from A
  receive $[\mathsf{sk_A sk_B}], [\mathsf{sk_B}]$ from B
  send $[\mathsf{sk_B sk_C}]$ to A
  send $[\mathsf{sk_A sk_C}]$ to B
  return $[\mathsf{sk_A sk_B sk_C}]$
```

For simplicity, we assume that the only public transcript are the public keys $[\mathsf{sk_A}]$, $[\mathsf{sk_B}]$, and $[\mathsf{sk_C}]$. I.e., $([\mathsf{sk_A}], [\mathsf{sk_B}], [\mathsf{sk_C}]) \leftarrow \mathit{trans}$. All other messages are sent via a pairwise secure channel, hence not visible to an adversary.

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

Solution:
It is secure. We can build a reduction from DDH to the group key exchange protocol. Since only the public keys are elements of the transcript, a reduction can tweak a DDH-triple $[x], [y], [z]$ by sampling a value $w \leftarrow\$ \mathbb{Z}_p$ and forwarding $([x], [y], [w], [w \cdot z])$ to the adversary.

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. (5 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

$$|\Pr[\mathcal{A}^0_{GKE}(\lambda) = 1] - \Pr[\mathcal{A}^1_{GKE}(\lambda) = 1]| \geq \epsilon(\lambda) \tag{6}$$

for some non-negligible function ϵ.

We use $\mathcal{A}$ to construct a distinguisher $\mathcal{B}$ against DDH as follows: On input a DDH triple $([x], [y], [z])$, $\mathcal{B}$ samples a random value $w \leftarrow\$ \mathbb{Z}_p$, and sets $[\mathsf{sk}_\mathsf{A}] = [x]$, $[\mathsf{sk}_\mathsf{B}] = [y]$, and $[\mathsf{sk}_\mathsf{C}] = [w]$. Then, $\mathcal{B}$ invokes $\mathcal{A}$ on inputs $[\mathsf{sk}_\mathsf{A}], [\mathsf{sk}_\mathsf{B}], [\mathsf{sk}_\mathsf{C}], [z \cdot w]$. Computing $[z \cdot w]$ is possible efficiently since $\mathcal{B}$ knows the integer w.

2 Points: description of reduction

Eventually, $\mathcal{A}$ outputs a bit b and $\mathcal{B}$ outputs the same bit b.

To check efficiency, we note that $\mathcal{B}$ samples a $\mathbb{Z}_p$ element, computes a single exponentiation, and forwards four group elements to $\mathcal{A}$. All of these operations are efficient. Then, $\mathcal{B}$ runs the PPT algorithm $\mathcal{A}$. So, $\mathcal{B}$ is efficient.

1 Points: analysis of runtime

We now analyze the success probability of $\mathcal{B}$. In the case where $z = x \cdot y$, $\mathcal{B}$ sends the well-formed group key to $\mathcal{A}$, therefore simulating the group key exchange game perfectly for $b = 1$. Thus, we get

$$\Pr[\mathcal{B}^{[x],[y],[x \cdot y]}(\lambda) = 1] = \Pr[\mathcal{A}^0_{GKE}(\lambda) = 1]. \tag{7}$$

In the case where z is a random integer, $w \cdot z$ is also a random integer, and hence $\mathcal{B}$ simulates the group key exchange game for $b = 1$ perfectly to the adversary. We thus get

$$\Pr[\mathcal{B}^{[x],[y],[z \leftarrow\$ \mathbb{Z}_p]}(\lambda) = 1] = \Pr[\mathcal{A}^1_{GKE}(\lambda) = 1]. \tag{8}$$

Combining (8) and (3) we can conclude that

$$|\Pr[\Pr[\mathcal{A}^1_{GKE}(\lambda) = 1] - \Pr[\mathcal{A}^0_{GKE}(\lambda) = 1| = \tag{9}$$

$$|\Pr[\mathcal{B}^{[x],[y],[z \leftarrow\$ \mathbb{Z}_p]}(\lambda) = 1] - \Pr[\mathcal{B}^{[x],[y],[x \cdot y]}(\lambda) = 1]| \geq \epsilon(\lambda). \tag{10}$$

2 Points: analysis of success

As this contradicts the assumption that DDH is hard to solve, $\mathcal{A}$ cannot exist and the group key-exchange protocol is secure in the presence of eavesdropper. This concludes the proof.

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.

Definition 2 (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

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

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

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

Definition 3 (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

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

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

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

Definition 4 (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.