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

Problem 1 (Multiple Choice) 19 Points

For each question, choose the letters corresponding to the statement that fits best and place it in the appropriate table below. **Each question has only one correct answer if not specified otherwise.** You will earn 1 point for each correct answer. For questions marked with multiple correct answers, you will receive points only if you select all correct choices and no incorrect ones. **For each incorrectly answered question, 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	a, b, d	c, d	a, b, d	a	d	c	a	c	c	d

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

(i) **This question has multiple correct answers.** Why is the key exchange security model $\mathsf{KE}^{eav}_{\mathcal{A},\Pi}(\lambda)$ below not sufficient for secure messaging?

```
$\mathsf{KE}^{eav}_{\mathcal{A},\Pi}(\lambda)$
$(trans, k) \leftarrow \langle A(1^\lambda), B(1^\lambda) \rangle$
$b \leftarrow \{0,1\}$
if $b = 0 : k' \leftarrow k$
if $b = 1 : k' \leftarrow_\$ \{0,1\}^\lambda$
$b' \leftarrow \mathcal{A}(k', trans)$
if $b' = b$ return $1$
else return $0$
```

a) $\mathsf{KE}^{eav}_{\mathcal{A},\Pi}(\lambda)$ assumes a passive adversary that only eavesdrops, but real-world adversaries can actively manipulate messages.

b) $\mathsf{KE}^{eav}_{\mathcal{A},\Pi}(\lambda)$ do not account for adversaries who control the communication channel and can delay, delete, or insert messages.

c) $\mathsf{KE}^{eav}_{\mathcal{A},\Pi}(\lambda)$ allows an adversary to impersonate legitimate parties in the protocol.

d) In real-world scenarios, adversaries may compromise a party's secret state, which is not covered by $\mathsf{KE}^{eav}_{\mathcal{A},\Pi}(\lambda)$.

(ii) **This question has multiple correct answers.** What properties does secure messaging require far beyond key exchange syntax and security?

a) Key exchange security does not require $\mathcal{A}$ to distinguish actual keys from random keys while secure messaging requires.

b) Secure messaging guarantees that an attacker can never eavesdrop on a message.

c) Secure messaging requires the security of continuous key exchange instances.

d) In the messaging security model, the adversary can corrupt the sender or receiver.

(iii) **This question has multiple correct answers.** What can adversary $\mathcal{A}$ do using its ability to make calls to $\mathsf{Oracle.Rcv}(c)$ oracle?

```
Oracle.Snd()
  $(st_R, K, c) \leftarrow \mathsf{snd}(st_S)$
  return $c$
```

```
Oracle.Rcv($c$)
  $(st_S, K) \leftarrow \mathsf{rcv}(st_R, c)$
  return $\perp$
```

a) Delaying the receiving time of a message or deleting that message.

b) Add its own created message to the receiver.

c) Decrypt the message c.

d) Alter the message sent by the sender.

(iv) What scenario is modeled by the following behavior of $\mathcal{A}$: $\mathcal{A}$ calls $\mathsf{Oracle.Snd}()$ to receive c but never calls $\mathsf{Oracle.Rcv}(c)$ in the entire game?

a) A typical package loss.

b) The adversary delays receiving time of c.

c) The sender is offline.

d) A replay attack of $\mathcal{A}$.

(v) What **is** a trivial winning condition for URKE protocols?

a) The adversary learns the real key K_0 through the receive oracle when the system is out of sync.

b) The adversary learns K_0 from incompatible (out-of-sync) states.

c) The adversary aborts without outputting b'.

d) The adversary exposes the receiver and queries the send oracle afterward.

(vi) What is **not** the reason for the need for synchronization (sync) in secure messaging?

a) It ensures that both parties always use the same cryptographic key at the same time.

b) It detects and prevents message manipulation such as reordering or loss.

c) It guarantees that messages are always delivered in real time without delay.

(vii) What is the limitation of our first ElGamal-based construction for secure messaging that uses a constant key size?

a) Compromising the state leaks all past and future exchanged keys.

b) It encrypts keys instead of messages, making it less efficient in practice.

c) The protocol lacks proper message encryption.

d) It does not handle large-scale communication or group messaging effectively.

(viii) What is a key property of a Unidirectional Ratcheted Key Exchange (URKE) protocol?

a) Messages are stored unencrypted.

b) The protocol supports backward compatibility.

c) Each exchanged key is used only once.

d) It does not require encryption for messages.

(ix) How does the Signal protocol achieve forward security during a conversation?

a) By using static key pairs for message encryption.

b) By relying on a trusted third party for key management.

c) By frequently updating independent cryptographic states.

d) By utilizing hardware-based secure key storage.

(x) Why can't post-compromise security (PCS) be achieved in unidirectional messaging?

a) Because the receiver's key updates depend on the sender.

b) Due to the lack of forward secrecy in the protocol.

c) The adversary can always recover the initial session state.

d) The receiver's state is deterministic and does not use randomness.

(xi) In the Double Ratchet Algorithm, what is the role of the Diffie-Hellman (DH) Ratchet?

a) To ensure confidentiality of initial messages.

b) To prevent message tampering.

c) To achieve post-compromise security by including fresh keys.

d) To provide integrity checks for all messages.

(xii) What is a limitation of Threema's pre-2023 encryption model?

a) Lack of forward security.

b) Over-reliance on ephemeral keys.

c) Absence of symmetric encryption.

d) Dependency on centralized key distribution.

(xiii) How does the Signal protocol ensure that an active attacker cannot violate message integrity?

a) By frequently rotating long-term keys.

b) By ensuring sender and receiver states become incompatible after message alteration.

c) By embedding a tamper-proof flag in each message.

(xiv) Which of the following statements about Tor's Directory Authorities is **true**?

a) Tor assumes no trusted entities, and all nodes are treated equally.

b) They publish the network consensus, which lists all active Tor nodes.

c) Tor uses a fully untrusted infrastructure to maintain the network consensus.

d) Clients do not need to download information from Directory Authorities before connecting to Tor.

(xv) Which of the following statements about Tor's security model is **true**?

a) Tor prevents endpoint user devices from being compromised by malware.

b) No Tor node in the network ever knows the user's real IP address.

c) Tor guarantees complete anonymity for its users.

d) Tor circuits are dynamically changed to minimize traffic analysis risks.

(xvi) Which of the following statements about Tor and tracking is **true**?

a) Using Tor prevents websites from tracking users through cookies and browser fingerprinting.

b) Tor ensures that all websites treat users as completely anonymous, regardless of their browsing behavior.

c) Users stick to a few long-term entry nodes (Guard nodes), making traffic correlation harder.

d) Once a user switches to Tor, previously visited websites cannot recognize them anymore.

(xvii) Which of the following statements about Tor circuit structure is **true**?

a) Tor circuits are built using one entry, two relays, and one exit node to reduce latency.

b) The client-encrypted request is only decrypted by the website.

c) No single node knows both the user's IP and the destination.

d) Exit relays always use the same IP address as the original user to maintain anonymity.

(xviii) Which type of attack attempts to analyze packet timing and volume at the entry and exit nodes of the Tor network?

a) Website fingerprinting

b) Sybil attacks

c) Traffic correlation attacks

d) Replay attacks

(xix) Which of the following statements about Tor exit nodes is **true**?

a) If a website does not use encryption, the exit node can see the unencrypted data.

Name: *Matriculation:*

b) Tor encrypts traffic all the way to its final destination, so exit nodes cannot see any data.

c) Exit nodes can see the destination website and always decrypt all traffic passing through them.

d) Exit nodes are randomly chosen each time a user connects, ensuring complete anonymity.

Problem 2 (Simplifying Messenger Ratchets) 7 Points

Let KDF be a key derivation function. We propose another approach for building ratchets in messaging. Instead of storing the local state

$$st'_{\mathsf{A}} = \mathsf{KDF}(st_{\mathsf{A}}, [\mathsf{esk}_{\mathsf{A}} \cdot \mathsf{epk}_{\mathsf{B}}]), \mathsf{sk}_{\mathsf{A}},$$

we propose using the local state

$$st'_{\mathsf{A}} = st_{\mathsf{A}}, [\mathsf{esk}_{\mathsf{A}} \cdot \mathsf{epk}_{\mathsf{B}}], \mathsf{sk}_{\mathsf{A}}$$

without applying the KDF iteratively on the old state as a Diffie-Hellman Ratchet. The symmetric ratched remains unchanged. While we are unaware of the security of this improved ratchet, we expect it to have better compatibility when users are in different stages of their respective ratchet. As a starting point for our security analysis, we care about sender forward security. Sender forward security is a simplified version of forward security, where the sender is corrupted at a certain point after ephemeral keys have been exchanged, and the send oracle is not called after a sender corruption. For simplicity, you can assume that when initializing the game $\mathsf{IND}^b_{\mathsf{URKE}}$, the local states already contain exchanged ephemeral information.

(i) In a few sentences, explain your intuition about whether the ratcheted key exchange achieves forward security. (2 Points)

Solution:
The key exchange is not forward secure. An adversary obtains the state at any point and can start the symmetric ratchet from the beginning. Then, the adversary can recompute the actual keys using the freshly started ratchet and compare them to the exchanged keys.

1 Points: correct intuition

1 Points: presents a valid argument

(ii) Provide formal proof to support your intuition. I.e., either provide a reduction or give an attack showing that the proposed ratchet does not achieve forward security. As an aid, we added the definitions of ratcheted key exchange to the last page of this exam. (5 Points)

Solution:
We propose the following adversary $\mathcal{A}$. As input, $\mathcal{A}$ obtains the security parameter 1^λ and has access to a send, a receive, and two expose oracles. The adversary queries the send and receive oracle until the DH ratchet is rotated at least once. Then, the adversary queries the send oracle one more time and then exposes the sender. Exposing the sender forwards the sender state st_{A} to the adversary. By the construction of our adapted ratchet, this state looks like $st'_{\mathsf{A}} = st_{\mathsf{A}}, [\mathsf{esk}_{\mathsf{A}} \cdot \mathsf{epk}_{\mathsf{B}}], \mathsf{sk}_{\mathsf{A}}$. Then, the adversary initializes a local symmetric ratchet by computing $ek_0 = \mathsf{KDF}(st'_{\mathsf{A}}, \mathsf{A}, \mathsf{B})$. This is possible since the old state of Alice is not part of the KDF. Then, $\mathcal{A}$ compares ek_0 with the key K_b returned by the send oracle after the ratched turned. $\mathcal{A}$ returns 0, iff. both keys are equal.

2 Points: description of attack

The proposed attacker has polynomial runtime since the DH ratchet is turned upon sending and receiving at most polynomially many messages. In addition, initializing the local ratchet is a single evaluation of an efficient KDF.

1 Points: analysis of runtime

Since the attacker can efficiently create a local copy of the symmetric ratchet, and there is no randomness involved in the protocol, the check if the local key and the key from the oracle are identical always is successful. Hence, the adversary can check the validity of the key with probability 1.

2 Points: analysis of success

Problem 3 (Security of Onion Encryption) 7 Points

In the lecture, we have seen that Tor uses onion encryption to secure exchanged messages. In this task, we want to prove the security of this approach. As a starting point, we use the CPA secure symmetric encryption scheme $\Pi_{\mathsf{Enc}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$. On an abstract level, the onion encryption can be realized using the encryption scheme $\Pi'_{\mathsf{Enc}} = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ defined via

```
$\mathsf{KGen}'(1^\lambda)$
  $k_1 \leftarrow \mathsf{KGen}(1^\lambda)$
  $k_2 \leftarrow \mathsf{KGen}(1^\lambda)$
  return $(k_1, k_2)$
```

```
$\mathsf{Enc}'(k_1, k_2, m)$
  $c_1 \leftarrow \mathsf{Enc}(k_1, m)$
  $c_2 \leftarrow \mathsf{Enc}(k_2, c_1)$
  return $c_2$
```

```
$\mathsf{Dec}'(k_1, k_2, c)$
  $c_1 \leftarrow \mathsf{Dec}(k_2, c)$
  $m \leftarrow \mathsf{Dec}(k_1, c_1)$
  return $m$
```

(a) Provide a reduction showing that the scheme Π'_{Enc} achieves CPA security. As an aid, we added the definition of CPA security for symmetric-key encryption schemes to the last page of this exam. (5 Points)

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

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

for some non-negligible function ϵ.

We use $\mathcal{A}$ to construct a reduction $\mathcal{B}$ against the CPA security of Π_{Enc} as follows: On input 1^λ, $\mathcal{B}$ invokes $\mathcal{A}$ on 1^λ. $\mathcal{B}$ generates a key k_1 using $k_1 \leftarrow \Pi_{\mathsf{Enc}}.\mathsf{KGen}(1^\lambda)$. For each message m queried by $\mathcal{A}$ to its encryption oracle, $\mathcal{B}$ queries $m' = \mathsf{Enc}(k_1, m)$ to it's own encryption oracle, receiving the ciphertext c. Then, $\mathcal{B}$ forwards c to $\mathcal{A}$. Eventually, $\mathcal{A}$ outputs two messages m_0, m_1 of equal length, and $\mathcal{B}$ outputs the messages m'_0, m'_1 using the above method. Eventually, $\mathcal{B}$ receives a challenge ciphertext c_b and forwards c_b to $\mathcal{A}$. Eventually, $\mathcal{A}$ outputs a bit b', and so does $\mathcal{B}$.

2 Points: description of reduction

First, note that $\mathcal{B}$ runs the PPT algorithm $\mathcal{A}$ and answers a polynomial number of queries. In addition, $\mathcal{B}$ samples a key efficiently and transforms messages using an efficient encryption algorithm. So, $\mathcal{B}$ is efficient.

1 Points: analysis of runtime

We now analyze the success probability of $\mathcal{B}$. $\mathcal{B}$ simulates the scheme Π'_{Enc} perfectly to $\mathcal{A}$ and $\mathcal{A}$ is an efficient adversary against the CPA security of Π'_{Enc}. Hence, $\mathcal{A}$ outputs the correct bit with probability

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

for some non-negligible function ϵ. This carries over to Π, since Π is part of Π' at the outer most layer such that our reduction $\mathcal{B}$ also outputs the correct bit with probability $\frac{1}{2} + \epsilon$. As this contradicts the assumption that Π is CPA secure, $\mathcal{A}$ cannot exist and Π' is also a CPA secure. This concludes the proof.

2 Points: success probabilities

(b) What happens, if Tor would use a more efficient key derivation function like this: For the three nodes along a Tor circuit, the entry node samples a random key k_1, the relay uses $k_2 = \mathsf{H}(k_1)$, and the exit node $k_3 = \mathsf{H}(k_2)$. In a few sentences, explain your intuition about whether such a circuit would still be secure. (2 Points)

Solution:
It is not secure. The entry node can decrypt the whole onion using its key by derivating key k_2 and k_3 locally.

1 Points: correct intuition
1 Points: presents a valid argument

Problem 4 (Tor Hidden Services) 7 Points

Answer to this problem in bullet points. Each correct bullet point will give you a point.

(a) Describe the steps a server has to do, to establish a Tor hidden service. (4 Points)

Solution:

- select several Tor nodes as introduction points.
- establish encrypted circuits to these introduction points.
- generate and share a public-private key pair with the introduction points.
- publish its .onion address to the distributed hash table (DHT).

1 Points: each bullet

(b) Describe, why publishing the hidden service descriptor is important. (3 Points)

Solution:

- Ensures that **only the legitimate service** can register its introduction points.
- Prevents **man-in-the-middle attacks** by verifying the descriptor signature.
- The **distributed nature of the DHT** enhances **resilience** against censorship.

1 Points: each bullet

Auxiliary Information

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

Unidirectional Ratcheted Key Exchange

```
Game.IND$^{b}_{\mathsf{URKE},\mathcal{A}}(\lambda)$
$s \leftarrow 0; r \leftarrow 0; sync \leftarrow 1$
$C[\cdot] \leftarrow \bot$
$(st_S, st_R) \leftarrow init()$
$b' \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda)$
Stop with $b'$
```

```
Oracle.Snd$^{b}()$
$(st_S, K_0, c) \leftarrow \mathsf{snd}(st_S)$
$K_1 \leftarrow\$ \mathcal{K}$
$C[s] \leftarrow c$
$s \leftarrow s + 1$
return $(K_b, c)$
```

```
Oracle.Rcv$(c)$
$(st_R, K_0) \leftarrow \mathsf{rcv}(st_R, c)$
if $c \neq C[r] \wedge sync = 1$ :
    $sync \leftarrow 0$
$r \leftarrow r + 1$
if $sync = 1$ :
    return $\bot$
return $K_0$
```

```
Oracle.ExposeS()
return $st_S$
```

```
Oracle.ExposeR()
return $st_R$
```

Tor

Definition 1. An encryption scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ 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

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

```
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
```

This is the last page of your exam.