Privacy Enhancing Technologies FS2026
Exercise Sheet 1 (version 1.2)

Florian Tramèr

Problem 1: Conceptual Questions. For each of the following statements, say whether it is TRUE or FALSE. Write at most one sentence to justify your answer.

(a) In the Random Oracle Model, we can build a commitment scheme such that recovering the committed message, or breaking binding (both with non-negligible probability) require that the attacker make an exponential number of oracle queries.

(b) If $\texttt{P} = \texttt{NP}$, then secure PRFs exist.

(c) In a FHE scheme, given a ciphertext $c = \texttt{Enc}(m)$, we have that $\texttt{Eval}(f, c) = \texttt{Enc}(f(m))$.

(d) The commitment scheme $\texttt{Commit}(m, r) = H(m, r)$ is provably secure if H is a cryptographic hash function like SHA-3.

Problem 2: Some (in)secure commitment schemes. For each of the following commitment schemes, argue whether it is statistically hiding, statistically binding, both, or neither.

(a) $\texttt{Commit} : \mathbb{Z}_q \times \{0,1\}^\lambda \to \mathbb{G}$ defined as $\texttt{Commit}(m, r) = H(r)^m$ where H is a hash function (modeled as a random oracle) from $\{0,1\}^\lambda$ to a cyclic group $\mathbb{G}$ of prime order q.

(b) $\texttt{Commit} : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ defined as $\texttt{Commit}(m, r) = m \oplus r$.

(c) The Pedersen commitment scheme, when the discrete logarithm x such that $h = g^x$ between the public generators g and h of $\mathbb{G}$ is known.

(d) The "ElGamal commitment scheme", defined as $\texttt{Commit}(m, r) = (g^r, g^m \cdot h^r)$, in the same setting as the Pedersen commitment scheme (x s.t. $h = g^x$ is known).

Problem 3: A tight proof for our cool PRF. Recall the PRF $F(k, x) = H(x)^k$ from the lecture, where H is a hash function (modeled as a random oracle) from $\mathcal{M}$ to a cyclic group $\mathbb{G}$ of prime order q, and $k \leftarrow^\$ \mathbb{Z}_q$.

In class, we showed that for any PPT adversary $\mathcal{A}$ making at most Q_{RO} random oracle queries, and a single PRF query, we can build a DDH adversary $\mathcal{B}$ such that

$$\text{AdvDDH}(\mathcal{B}, \mathbb{G}) \geq \text{AdvPRF}(\mathcal{A}, F) / Q_{RO} \ .$$

Let's now strengthen this result. For now, we still consider a PRF adversary $\mathcal{A}$ that makes a single PRF query. Intuitively, what we want is to embed the DDH challenge into *every* RO query, so that we don't have to guess which RO query is for the message that will be used to query the PRF.

(a) Consider the following simulation of $\mathcal{A}$ by $\mathcal{B}$, upon receiving a DDH challenge (X, Y, Z):

- On every RO query m_i, $\mathcal{B}$ sets $H(m_i) = X$
- Upon receiving the PRF query m_i, $\mathcal{B}$ returns Z.

- $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

What can you say about the advantage of $\mathcal{B}$?

(b) Let (X, Y, Z) be an arbitrary DDH triple. Show how to build a new triple (X', Y, Z') such that:

- X' is uniformly random in $\mathbb{G}$;
- if (X, Y, Z) is a "real" DDH triple (i.e., $(g^\alpha, g^\beta, g^{\alpha\beta})$), then (X', Y, Z') is a "real" DDH triple;
- if (X, Y, Z) is a "fake" DDH triple (i.e., $X = g^\alpha$, $Y = g^\beta$, $Z = g^\gamma$) for a random γ, then (X', Y, Z') is a "fake" DDH triple.

(c) Use your solution from (b) to build a better simulation of $\mathcal{A}$ by $\mathcal{B}$ so that:

$$\text{AdvPRF}(\mathcal{A}, F) = \text{AdvDDH}(\mathcal{B}, \mathbb{G}) .$$

(note that we still only consider a PRF adversary $\mathcal{A}$ that makes a single PRF query).

(d) **[Bonus]**: Show how to extend your solution from (c) to handle an arbitrary number of PRF queries.

Hint: Given a DDH challenge (X, Y, Z), show how to build a new triple (X', Y, Z') as in (b), but with the additional property that when the DDH challenge is fake, the value Z' is uniformly random in $\mathbb{G}$ *and independent of* X' *and* Y.

Problem 4: Vector Commitments. A vector commitment scheme is a tuple of algorithms $(\texttt{Commit}, \texttt{Open}, \texttt{Verify})$ where

- $\texttt{Commit}(m_1, \ldots, m_n)$ takes a vector of n messages in $\mathcal{M}$ and returns a commitment $c \in \mathcal{C}$.
- $\texttt{Open}(c, i, m)$ produces a proof π that m was committed to at position i in the commitment c (the opening may sometimes also need to take other messages in the commitment as input).
- $\texttt{Verify}(c, i, m, \pi)$ outputs 1 if and only if π is a valid proof that m was committed to at position i in the commitment c.

The security property we want is *position binding*: for any PPT adversary $\mathcal{A}$, it is hard to find a commitment $c \in \mathcal{C}$, an index $i \in [n]$, messages $m_1 \neq m_2 \in \mathcal{M}$ and proofs π_1, π_2 such that $\texttt{Verify}(c, i, m_1, \pi_1) = 1$ and $\texttt{Verify}(c, i, m_2, \pi_2) = 1$.

Note that we don't require a hiding property here.

For all questions below, let $H : \{0,1\}^* \to \{0,1\}^\lambda$ be a hash function modeled as a random oracle.

(a) Describe a secure vector commitment scheme where the commitment c is of size $n \cdot \lambda$ bits and an opening proof π is of size $O(1)$ bits. Argue that your scheme is position binding in the RO model.

(b) Describe a secure vector commitment scheme where the commitment c is of size λ bits and an opening proof π is of size $O(n \cdot |m|)$ bits, where $|m|$ is the size in bits of a message.

(c) Describe a secure vector commitment scheme where the commitment c is of size λ bits and an opening proof π is of size $O(\log n \cdot \lambda)$ bits. **Hint:** Build a tree!