PIR – Homework

Please answer the following questions. **We don't expect rigorous formal proofs: rather, just a high-level argument from intuition.** Please submit your answers as a **PDF** to Gradescope. Collaboration is allowed and encouraged, but you must write up your own answers and acknowledge your collaborators in your submission.

Due Date: *Monday, April 7th*

1 Commitment Schemes

Consider a commitment scheme $(\mathsf{Commit}, \mathsf{Open})$ for a finite message space $\mathcal{M}$ (for instance, $\mathcal{M} = \{0,1\}$). Recall the hiding and binding properties:

Hiding: The hiding property states that for any $m_0, m_1 \in \mathcal{M}$, $\mathsf{Commit}(m_0; r)$ is indistinguishable from $\mathsf{Commit}(m_1; s)$ for randomly sampled r and s.

Binding: The binding property states that it is hard for the sender to find $m_0, m_1 \in \mathcal{M}$, $m_0 \neq m_1$ and randomness r, s such that $\mathsf{Commit}(m_0; r) = \mathsf{Commit}(m_1; s)$.

(1) Recall the hash-based commitment scheme. The message space is the set of all strings, namely $\mathcal{M} = \{0,1\}^*$. To commit to a message $m \in \mathcal{M}$, the sender randomly samples $r \leftarrow \{0,1\}^\lambda$ where λ is the security parameter, and computes $\mathsf{Commit}(m; r) = H(r \| m)$. Why is this scheme hiding and binding (under what computational assumptions)?

(2) Recall the Pedersen commitment scheme on a cyclic group $\mathbb{G}$ of prime order q with generator g. Let the receiver first sample a random group element $h \in \mathbb{G}$. The message space is $\mathcal{M} = \mathbb{Z}_q$. To commit to a message $m \in \mathcal{M}$, the sender randomly samples $r \leftarrow \mathbb{Z}_q$ and computes $\mathsf{Commit}(m; r) = g^m \cdot h^r$. Why is this scheme hiding and binding (under what computational assumptions)?

(3) **(Extra Credit)** Recall the Merkle tree construction on a message space $\mathcal{M} = \{0,1\}^{2^d}$. For any message $m \in \mathcal{M}$, we build up a binary tree of depth d where each leaf node is a commitment to a single bit $\mathsf{Com}(m[i])$, where Com is a commitment scheme for bits. Each non-leaf node is computed as $H(\mathsf{lc} \| \mathsf{rc})$ where lc and rc are its left and right children. Finally, we use the root of the tree as the commitment for $\mathsf{Commit}(m)$. Why is this construction hiding and binding (under what computational assumptions)?

2 Crypto Notions

(1) What does each of zk, S, N, and ARG mean in zk-SNARG? What does K mean in zk-SNARK?

(2) Give a potential application of zk-SNARG/zk-SNARK in practice. (Try to come up with one that was not covered in class!)

(3) In one sentence, what is fully homomorphic encryption?

(4) Give a potential application of FHE in practice. (Try to come up with one that was not covered in class!)

3 Somewhat Homomorphic Encryption (SWHE)

(1) Intuitively speaking, what's the main reason that all the somewhat homomorphic encryption (SWHE) schemes that we see in class only support a bounded number of homomorphic operations, especially homomorphic multiplications?

(2) We construct private information retrieval (PIR) from SWHE in this project. Let's look at the tradeoffs in choosing different values for d – the dimension of the hypercube that we use to store our data. What value should we choose to minimize the number of homomorphic multiplications (optimizing computation)? What value should we choose to minimize the size of the selection vector (optimizing communication)?