NOTE: This is a non-collaborative assignment. You may not discuss the problems with any other students and you may not use any resources outside of the useful links given on the course website, the lecture notes and the course staff!

1 Two-key Encryption

When we covered secure two-party computation in class, we saw how to “encrypt,” or “garble,” a Boolean circuit. Recall that, to garble a circuit, Alice first assigns two keys to each wire \( w \) in the circuit: the 0-key \( K_w^0 \), and the 1-key \( K_w^1 \) (The exception is the output wire: \( K^0_{\text{out}} = 0 \) and \( K^1_{\text{out}} = 1 \)).

Let the \( i \)th gate in the circuit have input wires \( u \) and \( v \); and let \( w \) be its output wire. Let \( (a,b,c) \) be a row in the truth table of this gate; it means that if wire \( u \) carries the bit \( a \in \{0,1\} \) and wire \( v \) carries the bit \( b \in \{0,1\} \), then the gate will put the bit \( c \in \{0,1\} \) on the output wire \( w \). The gate’s truth table has four rows, corresponding to every possible assignment to \( (a,b) \).

To encrypt this gate, Alice forms four ciphertexts: for each possible assignment to \( (a,b) \), let \( (a,b,c) \) be a row in the truth table corresponding to this assignment; then compute \( c_{a,b} \leftarrow \text{Enc}((K_u^a, K_v^b), K_w^c) \), and output these four ciphertexts in a random order. I.e., the algorithm \( \text{Enc} \) takes a pair of keys \((K_u^a, K_v^b)\), and a message \( K_w^c \) as input, and uses the pair \((K_u^a, K_v^b)\) as an encryption key.

In order to evaluate the circuit, Bob will receive keys that correspond to his input bits, which will allow him to decrypt exactly one ciphertext per gate.

Recall that in class, we came up with the informal requirement that the underlying encryption scheme \( (\text{KeyGen}, \text{Enc}, \text{Dec}) \) must satisfy the following security properties: (1) correct decryption: if \( c \leftarrow \text{Enc}((K_1, K_2), K_3) \), then \( \text{Dec}((K_1, K_2), c) \) outputs \( K_3 \); (2) correct non-decryption: if \( c \leftarrow \text{Enc}((K_1, K_2), K_3) \), and \((K_1', K_2') \neq (K_1, K_2) \) then \( \text{Dec}((K_1', K_2'), c) \) outputs \( 1 \); (3) security: if Bob does not have both \( K_1 \) and \( K_2 \), then he learns nothing from \( c \); in particular, he does not learn if one of the keys he has is correct.

a. Formalize the security requirement (i.e. requirement (3) above).

b. Give a construction of \((\text{KeyGen}, \text{Enc}, \text{Dec})\) that satisfies correctness and security. Your construction may use, as a building block, a symmetric cryptosystem (or a block cipher or public-key cryptosystem if you prefer).

c. Prove that your construction in (b) satisfies your definition from part (a).
2 Zero-Knowledge Proofs and 2-Party Computation

Consider the following protocol for performing a zero-knowledge proof that a Boolean circuit \( C \) is satisfiable.

1. The Prover produces a garbled version of \( C \) and sends it to the Verifier.
2. The Verifier chooses a random bit \( b \) and sends it to the Prover.
3. If \( b = 0 \), the Prover reveals all the keys and randomness that were used in creating the garbled circuit, and sends these values to the Verifier.
   Let \( s_1, \ldots, s_k \) be a satisfying assignment for \( C \). If \( b = 1 \), the Prover gives the Verifier the keys \( K_{in_1}, K_{in_2}, \ldots, K_{in_k} \), where \( in_i \) is the \( i^{th} \) input wire.
4. If \( b = 0 \), the Verifier verifies that the values received from the Prover are correct. If \( b = 1 \), the Verifier evaluates the garbled circuit and accepts if it outputs 1.

a. Prove that this construction is complete. In other words, show that, if \( C \) is satisfiable and both the Prover and the Verifier follow the protocol, then the Verifier accepts.

b. Prove that this construction is sound. In other words, if \( C \) is not satisfiable, with what probability can a malicious Prover make the Verifier accept?

b. Show that this protocol is zero-knowledge. Hint: one of the things that your simulator might do is replace every gate in this circuit with a gate that always outputs 1. You might then need to use a hybrid argument to show that the simulation works, or else the encryption scheme you devised in Problem 1 is not secure.

3 Lattice-Based Cryptography

Let \( q \), \( n \), \( m \) be integers. Let \( A \) be an \( n \times m \) matrix with entries in \( \mathbb{Z}_q \); let \( a_{i,j} \) be the entry found in row \( i \), column \( j \) of \( A \). Let \( v \) be an \( m \)-dimensional vector with entries in \( \mathbb{Z}_q \). Let \( |v| \) denote the Euclidean length of \( v \); in other words \( |v| = \sqrt{\sum_{i=1}^{m} v_i^2} \), where \( v_i \) is the \( i^{th} \) entry in \( v \). Let \( 0_\ell \) denote the \( \ell \)-dimensional zero vector. We say that \( v \) is an integer solution for \( A \) if \( Av = 0_n \) (mod \( q \)); put another way, for \( 1 \leq i \leq n \), \( \sum_{j=1}^{m} a_{i,j} v_j = 0 \) (mod \( q \)). For \( \beta \in \mathbb{R}^+ \), we say that it is a \( \beta \)-short integer solution for \( A \) if \( |v| \leq \beta \). We say that it is a non-zero solution if \( v \neq 0_m \).

For certain settings of \( q \), \( n \), \( m \), \( \beta \) as a function of a security parameter \( k \), the following problem, known as the short integer solution (SIS) problem, is conjectured to be hard:

**Definition 1 ((q,n,m,β)-SIS problem)** Given an \( n \times m \) matrix \( A \) with entries drawn from \( \mathbb{Z}_q \) uniformly at random, find a non-zero \( \beta \)-short integer solution for \( A \).

Consider the following function \( H_A : \{0,1\}^m \to \mathbb{Z}_q^m \). On input an \( m \)-bit string \( x \), \( H_A \) treats it as an \( m \)-dimensional vector \( x \in \mathbb{Z}_q^m \) (since the values 0 and 1 are elements of \( \mathbb{Z}_q \)) and outputs the vector \( Ax \).

a. For what values of \( m \) (as a function of \( q \) and \( n \)) is the function \( H_A \) length-reducing?
b. Show that, given \( x \neq y \) such that \( H_A(x) = H_A(y) \), you can find a non-zero \( \sqrt{m} \)-short integer solution for \( A \) in polynomial time.

c. Give a construction of a collision-resistant hash function family whose security relies on the hardness of the \((q, n, m, \beta)\)-SIS problem, and prove its security.

4 Signatures: From Weak to Strong

A signature scheme is **weakly secure** if the probability that a ppt adversary wins the following game is negligible:

**Signing query:** On input \( 1^k \), the adversary chooses the messages \( M_1, \ldots, M_n \) to be signed.

**Response:** The signer runs the key generation and the signing algorithms and sends to the adversary the public key \( \text{pk} \) and the signatures \( \{\sigma_j\}_{j=1}^n \) on the adversary’s messages.

**Forgery:** The adversary outputs a message \( M^* \) and a signature \( \sigma^* \) and wins the game if \( M^* \) was not included in its signing query, and yet the verification algorithm accepts the signature \( \sigma^* \).

Recall that a signature scheme is **one-time secure** if the probability that a ppt adversary wins the following game is negligible:

**Key generation:** The challenger runs the key generation algorithm and generates \( (\text{pk, sk}) \).

**Signing query:** On input \( \text{pk} \), the adversary chooses a message \( m \) to be signed.

**Response:** The challenger computes a signature \( \sigma \) on \( m \) using the signing algorithm, and returns it to the adversary.

**Forgery:** The adversary outputs a message \( m^* \) and a signature \( \sigma^* \) and wins the game if \( m^* \neq m \), and yet the verification algorithm accepts the signature \( \sigma^* \).

Construct a secure signature scheme (i.e. one that satisfies the regular EUF-CMA security definition) given a weakly secure signature scheme \((\text{KeyGen}_{\text{weak}}, \text{Sign}_{\text{weak}}, \text{Verify}_{\text{weak}})\) and a one-time secure signature scheme \((\text{KeyGen}_{\text{one-time}}, \text{Sign}_{\text{one-time}}, \text{Verify}_{\text{one-time}})\). The message space for all the signature schemes here consists of all binary strings. Don’t forget to prove that your construction is correct and secure.

5 Discrete Log Based Commitment Scheme

Consider the following example of a commitment scheme based on the discrete log assumption over \( \mathbb{Z}_p \), where \( p = 2q + 1 \) for a prime \( q \). Define \( C = (\text{Setup, Commit, Open}) \) as follows:

\((p, q, g, h) \leftarrow \text{Setup}(1^k)\): Receiver chooses a safe prime \( p = 2q + 1 \) for a prime \( q \), and generators \( g, h \) of order \( q \).

\((c, (x, r)) \leftarrow \text{Commit}(x)\): For \( x \in \mathbb{Z}_q \), Committer chooses \( r \leftarrow \mathbb{Z}_q \) and sends \( c = g^x h^r \pmod{p} \) to the Receiver.
\[ \hat{x} \leftarrow \text{Open}(c, (x, r)) \]: Committer reveals \((x, r)\). Receiver outputs \(\hat{x}\), where \(\hat{x} = x\) if \(c = g^x h^r \pmod{p}\) and \(\hat{x} = 1\) otherwise.

a. Prove that this commitment scheme is unconditionally hiding.

b. Prove that this commitment scheme is computationally binding.