1 Malicious Security, Continued

To finish off our discussion of malicious security, we mention some definitional variants. Recall that an \( n \)-party protocol \( \Pi \) for computing some function \( f \) is \( t \)-secure if for all ppt adversaries \( A \) corrupting \( t \) parties, there exists some expected polynomial-time simulator \( S \) corrupting the same parties such that

\[
\begin{align*}
\{ \text{Real}^{A,\Pi}_{x,z}(1^k) \} \approx \{ \text{Ideal}^{S,f}_{x,z}(1^k) \}
\end{align*}
\]

We have the following security variants:

- One-sided security (for two-party protocols): Malicious security only holds when a specific party is corrupted (e.g., the evaluator in Yao’s 2PC protocol).

- Privacy-only: Protocol \( \Pi \) for computing some function \( f \) is \( t \)-private for malicious adversaries if for all ppt adversaries \( A \) corrupting \( t \) parties, there exists some expected polynomial time simulator \( S \) corrupting the same parties such that

\[
\begin{align*}
\{ \text{View}^{A,\Pi}_{x,z}(1^k) \} \approx \{ \text{Output}^{S,f}_{x,z}(n) \}
\end{align*}
\]

This is usually used in cases where the attacker gets no output.

2 Zero-knowledge Proofs

Let \( L \) be an \( \mathcal{NP} \)-language, and let \( R_L \) be a polynomial-time computable relation such that \( \forall x \exists w \ R_L(x,w) = 1 \iff x \in L \). A zero-knowledge (ZK) proof for \( L \) is a two-party protocol between a prover \( P \) and a verifier \( V \), such that the following three conditions hold:

1. (Completeness): \( \forall x, w, R_L(x, w) = 1 \implies \langle P(x, w), V(x) \rangle = 1 \).

2. (Soundness): \( \forall x \notin L, \forall P^*, \Pr[\langle P^*(x), V(x) \rangle = 1] \leq \epsilon(k) \). (Note that there are no restrictions on the running time of \( P^* \).)

3. (Zero-knowledge): \( \forall \text{ ppt } V^* \exists S \) running in expected polynomial time such that

\[
\begin{align*}
\{ \text{View}^{V^*}_{P(x,w),V^*(x)}(1^k) \} \approx \{ S(x) \} \end{align*}
\]
A zero-knowledge argument for $L$ is equivalent to the above definition, except soundness holds for all ppt $P^*$ (instead of $P^*$'s running time being arbitrary).

We now show a zero-knowledge proof for graph Hamiltonicity\(^1\). Since graph Hamiltonicity is $\mathcal{NP}$-complete, this implies that there exist zero-knowledge proofs for all languages in $\mathcal{NP}$.

Our zero-knowledge proof assumes the existence of a statistically binding and computationally hiding commitment scheme. We assume the reader is familiar with commitment schemes; if not, see [Gol01, §4.4.1]. The existence of such a commitment scheme is implied by one-way functions [Gol01, §4.4.1.3].

Zero-knowledge Protocol for Graph Hamiltonicity

\[
P(G, w) \\ V(G) \\
\text{com}(M') \\
b \\
\begin{array}{c}
\text{Check that } m \text{ is a valid decommitment, outputting 1 if so and 0 otherwise.}
\end{array}
\]

Let $G'$ be a random permutation $\pi$ of $G$. Let $M'$ be the adjacency matrix representation of $G'$, and let $\text{com}(M')$ be the commitment to each entry in $M'$.

If $b = 0$, let $m$ be the decommitment to all entries in $M'$. If $b = 1$, let $m$ be the decommitment to the Hamiltonian cycle in $M'$.

Completeness is straightforward to show. For soundness, we have the following claim:

**Theorem 1** If the commitment scheme $\text{com}$ is statistically binding, then the above protocol has soundness $1/2$.

**Proof** This follows from the fact that the commitment scheme is statistically binding, and thus cannot be broken. Thus, if $P^*$ can answer correctly for both $b = 0$ and $b = 1$, then $G$ must have a Hamiltonian cycle.

Finally, we have the following theorem for the zero-knowledge property:

**Theorem 2** If the commitment scheme $\text{com}$ is computationally hiding, then the above protocol is zero-knowledge.

**Proof** Fix a ppt verifier $V^*$. We construct a simulator $S(G, z)$, which takes as input a graph $G$ and an auxiliary string $z$, as follows:

- Do the following at most $k$ times:
  
  1. Choose $b \leftarrow \{0, 1\}$.

---

2. If \( b = 0 \), let \( M' \) be the adjacency matrix representation of a random permutation of \( G \), and send \( \text{com}(M') \) to \( V^* \).

3. If \( b = 1 \), let \( M' \) be the adjacency matrix representation of a random permutation of an \( n \) vertex Hamiltonian cycle, and send \( \text{com}(M') \) to \( V^* \).

4. If \( V^* \) sends \( b' = b \), then open \( \text{com}(M') \) accordingly and output the transcript.

5. If \( V \) sends \( b' \neq b \), then repeat.

We claim that \( \{ \mathcal{S}(G,z) \}_{G,z} \approx \{ \mathcal{V}(P(x,w),V^*(x,z))^{(1^k)} \}_{G,z} \). We prove this via a hybrid argument. Consider the following hybrid \( \text{Hybrid}(G,w,z) \):

- Do the following at most \( k \) times:
  1. Choose \( b \leftarrow \{0,1\} \).
  2. Compute \( \text{com}(M') \) as in the real protocol and send it to \( V^* \).
  3. If \( V^* \) sends \( b' = b \), then open \( \text{com}(M') \) accordingly and output the transcript.
  4. If \( V \) sends \( b' \neq b \), then repeat.

**Claim 3** \( \{ \text{Hybrid}(G,w,z) \}_{G,z} \approx \{ \mathcal{V}(P(x,w),V^*(x,z))^{(1^k)} \}_{G,z} \).

**Proof** Because of the uniform choice of \( b \), the probability that \( \text{Hybrid} \) never succeeds is \( 2^{-k} \). Conditioned on succeeding, \( \text{Hybrid} \) is equal to \( \mathcal{V} \), and thus the above claim holds.

**Claim 4** \( \{ \text{Hybrid}(G,w,z) \}_{G,z} \approx \{ \mathcal{S}(G,z) \}_{G,z} \).

**Proof** We prove this by reduction to the hiding property of the commitment scheme. Let \( D \) be a distinguisher between \( \text{Hybrid} \) and \( \mathcal{S} \) that succeeds with probability \( \varepsilon(k) \). Let \( \text{com}(\cdot,\cdot) \) be a “left-right” commitment oracle which returns either a commitment to its left input or a commitment to its right input. Define an attacker \( A^{\text{com}(\cdot,\cdot)} \), which takes as input a graph \( G \), a witness \( w \), and an auxiliary string \( z \), as follows:

- Repeat \( k \) times:
  1. Choose \( b \leftarrow \{0,1\} \).
  2. If \( b = 0 \) then commit to a random permutation of \( G \) as above.
  3. If \( b = 1 \) then commit to the Hamiltonian cycle in a random permutation of \( G \), and then for all other indices in the adjacency matrix \( E \) input the pair \((E_{i,j},0)\) to the commitment oracle.
  4. If \( V^* \) sends \( b' = b \), then open the commitments and run \( D \) on the resulting transcript, and stop, outputting what \( D \) outputs.

- Output \( \bot \).
If $\text{com}(\cdot, \cdot)$ commits to the left input, then the transcript is distributed exactly as in $\text{Hybrid}$; if $\text{com}(\cdot, \cdot)$ commits to the right input, then the transcript is distributed exactly as in $S$. Thus, $A$ succeeds in distinguishing the commitments with probability $\varepsilon(k)$, and thus by the assumed security of the commitment scheme it must be that $\varepsilon(k) \leq \text{negl}(k)$.

Thus, we have that $\{S(G, z)\}_{G, z} \approx \{\text{View}^{V^*}_{P(w), V^*(x, z)}(1^k)\}_{G, z}$, completing the proof.

References