CS 294 – Secure Computation March 15, 2016

Lecture 17: Resettable Zero-Knowledge, Resettable Sound Zero Knowledge

Instructor: Sanjam Garg *Scribe: Katia Patkin*

1 Resettable Zero-Knowledge

1.1 Introduction

A zero-knowledge proof (ZK) is a proof that enables to prove the truth of a given statement, without revealing anything else. Resettable zero-knowledge (rZK) is the strongest form of zero-knowledge known today. Basically, a rZK proof is a proof that remains ZK even if a polynomial-time verifier is able to make the prover execute the proof multiple times with the same random tape. More specifically, (1) The verifier can reset the prover. In each execution, the verifier can choose whether the prover execute the protocol with a new random tape or with a tape that was previously used. (2) The verifier can arbitrarily interleave executions. The verifier can start a new execution in the middle of an old one, or resume the old one whenever. (3) The prover is oblivious. From the point of view of the prover, it always executes a single instance of the protocol. rZK is a strengthening of the concurrent zero-knowledge (cZK), in which a malicious verifier acts like in rZK protocol, except it can not reset the prover's random tapes.

1.2 Model

Given a prover P, a common input x, an auxiliary input w to P, and P's random tapes ϕ. A potentially adversarial verifier interacts with the deterministic prover strategy $P_{x,w,\phi}$ that is determined by uniformly selecting and fixing ϕ. The adversary verifier may invoke and interact with many instances of $P_{x,w,\phi}$, each interaction with instances of $P_{x,w,\phi}$, is called a *session*. In each session $P_{x,w,\phi}$ "execute" a single session strategy P, that is $P_{x,w,\phi}$'s actions are oblivious of other sessions. On the other hand, the adversary's actions can depend on other sessions. Given the described above we extend the model to be interleaved that is the adversary may concurrently initiate and interact with $P_{x,w,\phi}$ in many sessions. This reminds the concurrent zero knowledge model, however to make this model imply cZK, the adversary may interact polynomially many times with each P_{x_i,w_i,ϕ_j}, where ϕ_j are random and independent of each other.

Definition 1 (rZK - standard model) *An interactive proof system (P, V) for a language L is said to be* `resettable zero-knowledge` *if for every probabilistic polynomial-time (PPT) adversary V^* there exists a PPT simulator M^* so that following two distribution ensembles are computational indistinguishable: Let each distribution be indexed by a sequence of common inputs $\bar{x} = x_1, \ldots, x_{poly(n)} \in L \cap \{0,1\}^n$ and a corresponding sequence of prover's auxiliary-inputs $\bar{w} = w_1, \ldots, w_{poly(n)}$,*

`Distribution 1` *is defined by the following random process which depends on P and V^*.*

- *Randomly select and fix $t = poly(n)$ random-tapes $\phi_1, \ldots, \phi_t$ for P resulting in deterministic strategies $P^{(i,j)} = P_{x_i,w_i,\phi_j}$ defined by $P_{x_i,w_i,\phi_j}(\alpha) = P(x_i, w_i, \phi_j, \alpha)$, for $i, j \in \{1, \ldots, t\}$,* *where α is a message from V^*.*

- *V^* interacts with $\{P^{(i,j)}\}_{i,j}$ in interleaved form.*
- *Once V^* decides it is done interacting with $P^{(i,j)}$, it produces an output based on its view of the interactions. Lets denote this output by $\langle P(\bar{w}), V^*\rangle(\bar{x})$.*

`Distribution 2` *The output of $M^*(\bar{x})$.*

1.3 Protocol

Let (P, V) be protocols in which the first verifier-message determines all its subsequent messages. This means that the verifier can only either abort or send a predetermined message. The first verifier-message is a sequence of commitments that are decommitted in subsequent verifier steps. In such a case the verifier's subsequent steps are either to send an illegal decommitment (essentially aborting) or decommit to the predetermined value. Each subsequent message of the verifier-message is comprised of two parts: (1) The main part - the revealed value and (2) The authenticator - the extra decommitment information, that establishes the validity of this value. The main value, determined by the commitment and the other part the authenticator, decommitment information, may change. If the prover's subsequent actions depend on the authenticator to be valid, it will abort for invalid authenticator, and for valid authenticator its actions depend only on the main part. In the resettable setting the first message in a session is always sent by the verifier and specifies the incarnation of P. The second message is sent by the prover, and is called the `prover initialization message`. The third message is sent by the verifier and is called the `determining message` of the session. The prover's randomness is achieved with PRF (pseudorandom function). The PRF is applied to the determining message and the input. When P receives the determining message, it sets the random tape to the result of applying the PRF.

Given com is some commitment scheme.

- $V \to P$: Send $\mathsf{FM} = \mathsf{com}(\sigma), \{\mathsf{com}(\sigma^b_{i,j})\}_{i\in[k],j\in[k],b\in\{0,1\}}$.
- P: The strategy $P_{x_i,w_i,\phi_j}(\alpha)$.
- Generating r using different randomness: $r_{i,j} = \mathsf{PRF}_{\phi_j}(x_i, w_i, \mathsf{FM})$ (Notice that now even the rewind does not help the verifier)
- For every $i \in [k]$:
  - $P \to V$: Send $r_{i,1}, \ldots, r_{i,k}$
  - $V \to P$: Decommit to $\sigma^{r_{i,j}}_{i,j}$ for all $j \in [k]$

1.4 Proof

We want to show that to simulate the view of V^* it is enough to simulate the view of $V^*_{NR,NA}$, where NR is Non Repeating and NA is Non Aborting, where $V^*_{NR,NA}$ is just the case of the concurrent ZK, because the first message determines the rest of the messages. [The proof for cZK we saw in the previous class]. The proof is comprised of two steps first we show that the view of the cheating verifier V^* is the same as the view of the V^*_{NR} and in the second step we show that the view of V^*_{NR} is the same as the view of $V^*_{NR,NA}$.

- Step 1($V^* \approx V^*_{NR}$): The simulator answer for every repeating query from V^* on it is own, hence removing the repetitions. No need to use the prover. As a result we get a guarantee that $P^{(i,j)}$ is done only once.
- Step 2 ($V^*_{NR} \approx V^*_{NR,NA}$): The simulator can simulate a view where for every aborting query, it answers $\perp$.

In more details, we construct W^* that will serve as a "mediator" between adversary V^* and prover P. When V^* starts a new session with determining message that is different from all previous ones, W^* will pass the messages of this session between V^* and P. When V^* "replays" an existing session, W^* will responds to V^* using the answers of P in this session, without involving P. Finally W^* outputs whatever V^* outputs. The construction of W^* (where $P^{(i,j,k)} = P_{x_i,w_i,\phi_{j,k}}$):

1. V^* initiates a new session with some $P^{(i,j)}$. W^* initiates a new session $P^{(i,j,k)}$ where k is a new index. Next it obtains the prover initialization and forwards the message to V^*.
2. V^* sends a new determining message (α) to $P^{(i,j)}$. W^* sends the message to one of the sessions $P^{(i,j,\cdot)}$, that awaits a determining message. W^* forwards the response to V^*. This session becomes the active session of (i, j, α) and stores the prover's response.
3. V^* repeats a first message to $P^{(i,j)}$. (The determining message is α). W^* forwards to V^* the relevant message from its storage in the active session of (i, j, α). W^* does not communicate with any session of P.
4. V^* sends a valid message to $P^{(i,j)}$. (The determining message is α). Let l denote the index of the current message sent by V^*. There are two cases:
    - The current session is the first session of V^* with $P^{(i,j)}$ with determining message α and V^*'s l^{th} message is valid: W^* forwards the current message to the active session of (i, j, α) it forwards P's to V^* and stores it.
    - The current session is *NOT* the first session of V^* with $P^{(i,j)}$ with determining message α and V^*'s l^{th} message is valid: W^* forwards to V^* the proper response from its storage, without communicating with any session of P.
5. V^* sends an invalid message to $P^(i, j)$. W^* replies with P's `abort` message to V^*. W^* does not forward this invalid message to any session of P.
6. V^* terminates: W^* output the same output message as V^* and halts.

We get that the output of W^* is computationally indistinguishable from the output of V^*.

2 Resettable Sound Zero Knowledge

2.1 Introduction

The rsZK is similar to the rZK however here the prover can reset the verifier to use the same random tape in multiple concurrent executions. Informally, an interactive proof achieves resettable soundness if a prover cannot convince a verifier of an incorrect statement with non-negligible probability.

2.2 Definition

Given a specified verifier program V and a common input x, and V's random coins r. $V_{x,r}$ determined by uniformly selecting and fixing r. The adversary may invoke and interact with $V_{x,r}$ while r is uniformly selected and fixed once and for all. Each such interaction is called a session. The adversary and $V_{x,r}$ can have polynomially-many sessions. $V_{x,r}$ performs the "single session strategy" V and oblivious to other sessions. The actions of the adversary may depend on other sessions. The aim of the cheating prover, is to convince $V_{x,r}$ to accept x in one of these sessions, while $x \notin L$. Since the verifier's randomness is fixed the adversary can "effectively rewind" the verifier to any point in a prior interaction and continue from this point. The model we look at is extending the above explanation and the adversary can to interact (many times) with several random independent incarnations of V. That is, the adversary may interact many times with different V_{x_i,r_j}, where the r_j's are independently and randomly selected and the x_i's are chosen dynamically by the adversary.

Definition 2 (resettable verifier-main model) *: A resetting attack of a cheating prover P^* on a resettable verifier V is defined by the following two-step random process, indexed by a security parameter n.*

1. *Uniformly select and fix $t = \texttt{poly}(n)$ random-tapes, denoted $r_1, \ldots, r_t$, for V, resulting in deterministic strategies $V^{(j)}(x) = V_{x,r_j}$ defined by $V_{x,r_j}(\alpha) = V(x, r_j, \alpha)$, where $x \in \{0,1\}^n$ and $j \in [t]$. Each $V^{(j)}(x)$ is called an incarnation of V .*

2. *On input 1^n, machine P^* is allowed to initiate poly(n)-many interactions with the $V^{(j)}(x)$'s. The activity of P^* proceeds in rounds. In each round P^* chooses $x \in \{0,1\}^n$ and $j \in [t]$, thus defining $V^{(j)}(x)$, and conducts a complete session with it.*

Let P and V be some pair of interactive machines, and suppose that V is implementable in PPT. Then for (P, V) *resettable-soundness* is that for every resetting attack, the probability that in some session the corresponding $V^{(j)}(x)$ has accepted but $x \notin L$ is negligible.

2.3 Resettably-Sound Zero-Knowledge Arguments

Proposition 1 *Let $L \in NP$ and R be a corresponding witness relation. Suppose that (P, V) is a constant-round public-coin argument of knowledge for R, and let f_s be a collection of PRFs. Assume, wlog, that on common input x, in each round, the verifier V sends a uniformly distributed $|x|$-bit string. Let W_s be a deterministic verifier program that, on common input $x \in \{0,1\}^{|s|}$, emulates V except that it determines the current round message by applying f_s to the transcript so far. Let W be defined so that on common input x and uniformly random-tapes $s \in \{0,1\}^{|x|}$, it acts as $W_s(x)$. Then:*

1. *(P, W) is a resettably-sound argument for L.*

2. *If (P, V) is zero-knowledge (witness-indistinguishable) then so is (P, W).*

The transformation above may yield something interesting only when applied to protocols that do not have a black-box simulator. Using Barak's protocol, constant-round public-coin zero-knowledge argument of knowledge for NP, which uses a non-black-box simulator, we obtain resettably-sound zero-knowledge arguments (of knowledge) for NP.
For detailed proof of the proposition see `https://eprint.iacr.org/2001/063.pdf`