Cryptographic Protocols
Notes for Lecture 3

Scribe: Sandro Coretti

About the notes: These notes serve as written reference for the topics not covered by the papers that are handed out during the lecture. The material contained therein is thus a strict subset of what is relevant for the final exam.

This week, the notes discuss the definition of (perfect) zero-knowledge and a proof that the three-move protocols we have encountered so far (graph isomorphism, Fiat-Shamir, Guillou-Quisqater, Schnorr) are perfectly zero-knowledge [Mau09, Theorem 2].

3.1 Definition of Zero-Knowledge

Intuitively, an interactive proof \((P, V)\) between a prover \(P\) and verifier \(V\) is zero-knowledge if after interacting with \(P\), any verifier \(V'\) has no more information than before executing the protocol. This is captured by the notion of a simulator \(S\) that reproduces \(V'\)'s view in the proof without actually communicating with \(P\).

More precisely, consider the following two random experiments:

1. Prover \(P\) interacts with \(V'\); let \(Z\) be the random variable corresponding to the resulting transcript and \(P_Z\) its distribution.

2. Simulator \(S\) interacts with \(V'\) and outputs a transcript; let \(Z'\) denote the corresponding random variable and \(P_{Z'}\) its distribution.

**Definition 3.1.** An interactive proof \((P, V)\) is (perfectly) zero-knowledge if for every efficient \(V'\) there exists an efficient simulator \(S\) (with access to \(V'\)) producing a transcript \(Z'\) that is distributed identically to the transcript \(Z\) in the actual interaction between \(P\) and \(V'\), i.e.,

\[
P_Z = P_{Z'}.
\]

The interactive proof is honest-verifier zero-knowledge (HVZK) if the simulator exists for (the honest) verifier \(V\).

In this course, when proving the zero-knowledge property, there will always be a single simulator \(S\) that works for all verifiers \(V'\). This is referred to as black-box simulation.

3.2 Honest-Verifier Zero-Knowledge and \(c\)-simulatability

The HVZK property is perhaps not very interesting per se, but it is a useful tool in proving (perfect) zero-knowledge. All three-move protocols in this course satisfy the even stronger
notion of c-simulatability.

**Definition 3.2.** A three-move protocol round of an interactive proof $(P, V)$ with challenge space $C$ is c-simulatable\(^1\) if for any value $c$ one can efficiently generate a triple $(t, c, r)$ with the same distribution as occurring in the protocol (between $P$ and the honest $V$) conditioned on the challenge being $c$.

In other words, there has to exist an efficient algorithm that given any $c \in C$ produces values $t$ and $r$ with a distribution $\hat{P}_{TR} \mid C$ such that $\hat{P}_{TR} \mid C(t, r, c) = P_{TR} \mid C(t, r, c)$ for all $t, c, r$, where $P_{TR}(t, r, c)$ is the distribution occurring in the actual protocol conditioned on the challenge being $c$.

It is easily seen that c-simulatability implies HVZK: the honest-verifier simulator simply chooses $c \in C$ uniformly at random and generates $t$ and $r$ according to the c-simulatability.

### 3.3 Proving the Zero-Knowledge Property

In this section we show that an interactive proof $(P, V)$ consisting of independent perfectly HVZK three-move rounds is perfectly zero-knowledge if, additionally, the challenge space $C$ is not too large. To that end, we first show how to formalize the behavior of the algorithms $P$, and $V'$ for said type of protocols.

#### 3.3.1 Behaviors

The behavior of the prover algorithm is fully defined by specifying the distribution of the first message $T$ and that of the response $R$ given the first message $T$ and the challenge $C$, i.e., by conditional probability distributions

$$p_P^T(t) \quad \text{and} \quad p_{R \mid TC}^P(r, t, c),$$

where $p_P^T(t)$ is the probability that the prover outputs $t$ as its first message and $p_{R \mid TC}^P(t, c, r)$ is the probability that $P$ outputs $r$ as the response given that the first message and the challenge were $t$ and $c$, respectively. Thus,

$$\sum_t p_P^T(t) = 1$$

and, for any $t$ and $c$,

$$\sum_r p_{R \mid TC}^P(r, t, c) = 1.$$

Contrary to $P$, the verifier $V'$ may not behave identically in every round, and the way it chooses the challenge $c_i$ in the $i^{th}$ round may depend on the entire transcript $u_{i-1} := (t_1, c_1, r_1, \ldots, t_{i-1}, c_{i-1}, r_{i-1})$ of rounds 1 to $i - 1$ and on the first message $t_i$ in the $i^{th}$ round. Therefore, its behavior is specified by a sequence of functions

$$p_{C_i \mid T_i U_{i-1}}^{V'}(c_i, t_i, u_{i-1}),$$

for $i = 1, 2, 3, \ldots$. Again, for any $t_i$ and $u_{i-1}$,

$$\sum_{c_i} p_{C_i \mid T_i U_{i-1}}^{V'}(c_i, t_i, u_{i-1}) = 1.$$
### 3.3.2 Perfect Zero-Knowledge

**Theorem 3.1.** An interactive proof \((P, V)\) consisting of \(k\) independent perfectly HVZK three-move rounds is perfectly HVZK. If, additionally, in every round \(V\) chooses the challenge uniformly at random from the same polynomially bounded challenge space \(C\), the protocol is perfectly zero-knowledge.

Note that Theorem 3.1 is a slightly more general than Theorem 2 in [Mau09] in that it works for any HVZK protocol and not only for c-simulatable ones.

**Proof.** The interactive proof \((P, V)\) is easily seen to be HVZK.

Consider now a potentially dishonest verifier \(V'\). The simulator \(S\) has black-box rewinding access to \(V'\). This means that \(S\) cannot see the code of \(V'\) (hence, it uses it as a black-box), but \(S\) may rewind \(V'\) at any point to an earlier state in its computation.

Simulator \(S\) creates a transcript in the following round-by-round fashion. Assume triples \(u_{i-1} = (t_1, c_1, r_1, \ldots, t_{i-1}, c_{i-1}, r_{i-1})\) for the first \(i - 1\) rounds have already been generated and that \(V'\) is in the corresponding state. For the \(i^{th}\) round, \(S\) proceeds as follows:

1. Generate a triple \((t'_i, c'_i, r'_i)\) according to the HVZK simulation.
2. Pass \(t'_i\) to \(V'\) and receive the challenge \(c''_i\) for the \(i^{th}\) round.
3. If \(c''_i = c'_i\), store the triple \((t_i, c_i, r_i) := (t'_i, c'_i, r'_i)\). Otherwise, rewind \(V'\) to the point before it received \(t'_i\) and repeat the simulation attempt.

The expected number of trials for every round is \(|C|\), which is polynomial by assumption.

In both the random experiment corresponding to the actual interaction between \(P\) and \(V'\) as well as that corresponding to the simulation, denote by \((T_i, C_i, R_i)\) the random variables corresponding to the transcript triples and let \(U_i := (T_1, C_1, R_1, \ldots, T_{i-1}, C_{i-1}, R_{i-1})\).

It remains to prove that the distribution \(P_{U_k}\) of the transcript in the actual interaction between \(P\) and \(V'\) is the same as the distribution \(\hat{P}_{U_k}\) of the simulated transcript. The proof is by induction on the number of rounds. The basis \(i = 1\) is trivial. Assume \(P_{U_{i-1}} = \hat{P}_{U_{i-1}}\). Clearly, it is sufficient to prove that

\[
P_{T_iC_iR_i|U_{i-1}} = \hat{P}_{T_iC_iR_i|U_{i-1}}.
\]

Hence, for the remainder of the proof, fix a \(u := u_{i-1}\) with non-zero probability \(P_{U_{i-1}}(u_{i-1})\).

Consider first the actual interaction between \(P\) and \(V'\) and note that

\[
P_{T_iC_iR_i|U_{i-1}}(t, c, r, u) = P_{T_i|U_{i-1}}(t, u) \cdot P_{C_i|T_iU_{i-1}}(c, t, u) \cdot P_{R_i|C_iT_iU_{i-1}}(r, c, t, u)
\]

\[= p^P_T(t) \cdot p^{V'}_{C_i|T_iU_{i-1}}(c, t, u) \cdot p^P_{R_i|TC}(r, t, c)
\]

for any \((t, c, r)\).

In the simulation, any candidate triple \((T'_i, C'_i, R'_i)\) (in isolation) in the \(i^{th}\) round is distributed according to

\[
\hat{P}_{T'_iC'_iR'_i}(t, c, r) = \frac{1}{|C|} \cdot p^P_T(t) \cdot p^P_{R_i|TC}(r, t, c),
\]

since it is generated via the perfect HVZK simulation. Moreover,

\[
\hat{P}_{T'_iC'_iR'_i}(t, c, r) \cdot p^{V'}_{C_i|T_iU_{i-1}}(c, t, u) = \frac{1}{|C|} \cdot P_{T_iC_iR_i|U_{i-1}}(t, c, r)
\]

is the probability that \(t, c, r\) is chosen as candidate triple and \(V'\) chooses (the same) \(c\) as the challenge, where the equality follows using (1) and (2).
Denote by $J_i$ (the random variable corresponding to) the number of candidate triples that need to be sampled in the $i^{th}$ round. Then, by total probability,

$$
\hat{P}_{T,C,R_i|U_{i-1}}(t,c,r,u) = \sum_j \hat{P}_{T,C,R_i|J_i,U_{i-1}}(t,c,r,j,u) \cdot \hat{P}_{J_i|U_{i-1}}(j,u) \quad (4)
$$

for any $t,c,r$. Note that for each $j$, $\hat{P}_{T,C,R_i|J_i,U_{i-1}}(t,c,r,j,u)$ is the probability that the $i^{th}$ simulated triple is $(t,c,r)$ conditioned on there being $j$ simulation attempts in the $i^{th}$ round (and on $u$). Hence, using (3),

$$
\hat{P}_{T,C,R_i|J_i,U_{i-1}}(t,c,r,j,u) = \frac{1}{|C|} \frac{1}{|R_i|} P_{T,C,R_i|U_{i-1}}(t,c,r,u) = P_{T,C,R_i|U_{i-1}}(t,c,r,u)
$$

for every $j$. Inserting this into (4) concludes the proof.

\[\square\]

References