ETH Zurich, Department of Computer Science
SS 2021

Prof. Ueli Maurer
Dr. Martin Hirt
Konstantin Gegier
Chen-Da Liu Zhang

Cryptographic Protocols
Notes 3

Scribe: Sandro Coretti (modified by Chen-Da Liu Zhang and Konstantin Gegier)

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-Quisquater, Schnorr) are perfectly zero-knowledge [Mau15, 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 *any* (poosibly dishonest) verifier V', after interacting with P, 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. One can hence argue that if V' thinks the protocol execution with P might be of some kind of usefulness for V', then V' could achieve exactly the same without talking to P (namely, by the simulation).

More precisely, consider the following two random experiments, the *real protocol execution* and the *simulation.* To differentiate between the two experiments we use $\hat{P}$ to denote probabilities in the second random experiment.

1. Prover P interacts with V' on common input z; let U denote the random variables corresponding to the transcript and let P_U denote its distribution.
2. Simulator S, for input z, outputs a transcript. Again, let U denote the corresponding random variable for the transcript and let $\hat{P}_U$ denote its distribution.[1]

Definition 3.1. *An interactive proof (P, V) is* (perfectly) zero-knowledge (ZK) *if for every efficient V' there exists an efficient*[2] *simulator S, which on every instance z produces a transcript U that is distributed identically to the transcript U in the actual interaction between P and V' (on common input z), i.e., such that*

$$P_U = \hat{P}_U.$$

[1]On the slide with the definition of zero-knowledge, the transcript is denoted as U and the simulated transcript as U'. Here, we use the same symbol (namely U) for both types of transcripts, and instead use different symbols for the distribution (P vs. $\hat{P}$).

[2]Efficient is usually interpreted as expected polynomial time.

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

We point out that for the above definition to be precise, the mathematical type of a verifier V' must be defined. Most treatments specify this type as a probabilistic polynomial-time Turing machine (PPT). Such technicalities could actually be avoided and are not discussed in this course.

In this course, and in almost all cases in the literature, when proving the zero-knowledge property of a protocol, there is a single simulator S (with access to the verifier V' as an oracle) which is able to simulate the transcript. This is referred to as *black-box* simulation.

Definition 3.2. *An interactive proof (P, V) is* (perfectly) black-box zero-knowledge (BB-ZK) *if there exists an efficient simulator S, with rewinding*[3] *oracle access to the verifier V', for which the condition of the previous definition holds.*

3.2 Some Relevant (Conditional) Probability Distributions

We discuss the relevant (conditional) probability distributions occurring in the later analyses. Since the distributions generally do not themselves constitute a random experiment but should rather only be understood as the mathematical objects of (conditional) probability distributions, we denote them by a small letter p. We denote as a superscript the object (P, V, or V') that determines the distribution.

- p_T^P denotes the prover P's distribution in choosing the first protocol message T.
- $p_{R|TC}^P$ denotes the prover P's distribution in choosing the reply R when given the first protocol message T and the challenge C.
- p_C^V denotes the honest verifier's distribution in choosing the challenge C (which is usually the uniform distribution).
- $p_{C|T}^{V'}$ denotes, for any (possibly dishonest) verifier V', the conditional distribution according to which it chooses the challenge C after seeing the first message T.

3.3 Honest-Verifier Zero-Knowledge and c-Simulatability

The HVZK property mentioned above is perhaps not very interesting per se, but it is a useful tool on the way to proving the (perfect) zero-knowledge property.

A crucial property, which all three-move protocols in this course satisfy, is c-simulatability:

Definition 3.3. *A three-move protocol round of an interactive proof (P, V) with challenge space $\mathcal{C}$ is* c-simulatable[4] *if for every instance z and 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, i.e., if the conditional distribution $p_{TR|C}$ is efficiently samplable.*

Note that we have

$$p_{TR|C}(t, r, c) \;=\; p_T^P(t) \cdot p_{R|TC}^P(r, t, c)$$

for all t, r, c. The basic idea of c-simulatability is that, while $p_{R|TC}^P$ is *not* efficiently samplable without knowledge of the secret x (which the real prover knows), the combined distribution

[3]Rewinding means that V' can, multiple times, be reset to an intermediate point of its computation. The rewinding capability is essential.

[4]This is also called *special HVZK* in the literature.

$p_{TR|C}$ can nevertheless be efficiently samplable, namely by first choosing r uniformly at random and then determining t such that the protocol's verification equation is satisfied (for the given c).

Lemma 3.1. *A three-move c-simulatable protocol is HVZK.*

Proof. The honest-verifier simulator simply chooses $c \in \mathcal{C}$ randomly according to the distribution p_C^V, and generates t and r according to $p_{TR|C}$ (which is possible due to the c-simulatable property). Hence we have

$$\hat{P}_{TCR}(t,c,r) \;=\; p_C^V(c) \cdot p_T^P(t) \cdot p_{R|TC}^P(r,t,c) \;=\; P_{TCR}(t,c,r),$$

showing that the distribution of the simulated transcript (T,C,R) is identical to the distribution of the real-protocol transcript (T,C,R). $\square$

3.4 Proving the General Zero-Knowledge Property

Lemma 3.2. *An HVZK three-move protocol (as discussed) with uniformly random challenge (by an honest verifier) from a polynomially bounded challenge space $\mathcal{C}$ is (black-box) zero-knowledge.*

Proof. Consider a potentially dishonest verifier V'. The simulator S has *black-box rewinding access* to V'. This means that the simulator can repeatedly invoke V' to generate a challenge c for a given first message t.

The simulator S creates a transcript as follows:

1. Generate a triple (t,c,r) according to the HVZK simulation.
2. Invoke V' with input t to obtain a challenge value; call it $\overline{c}$.
3. If $\overline{c} = c$, output the triple (t,c,r). Otherwise go to step 1 (and rewind V' to its start).

We need to show (1) that the distribution of the simulated transcript is correct (i.e., equal to the distribution of the transcript generated in the real protocol execution) and (2) that the simulator is efficient.

To prove the first claim, note that the transcript in the actual protocol execution between P and V' is distributed according to

$$P_{TCR}(t,c,r) \;=\; p_T^P(t) \cdot p_{C|T}^{V'}(c,t) \cdot p_{R|TC}^P(r,t,c).$$

We denote the random variables corresponding to an *attempted* simulation of a transcript (step 1 above) as T', C', and R'. The HVZK simulator outputs a transcript (t,c,r) with the distribution

$$\hat{P}_{T'C'R'}(t,c,r) \;=\; p_T^P(t) \cdot \frac{1}{|\mathcal{C}|} \cdot p_{R|TC}^P(r,t,c).$$

Then, in step 2 above, V' is invoked to generate a challenge value $\overline{c}$, according to the distribution $p_{C|T}^{V'}(\overline{c},t)$. Let us denote by $\mathcal{E}$ the event that $\overline{c} = c$ (in step 3). The (partial) distribution of (T',C',R'), and the event $\mathcal{E}$ occurring, is[5]

$$\hat{P}_{T'C'R'\mathcal{E}}(t,c,r) \;=\; p_T^P(t) \cdot \frac{1}{|\mathcal{C}|} \cdot p_{R|TC}^P(r,t,c) \cdot p_{C|T}^{V'}(c,t)$$

[5] $\hat{P}_{T'C'R'\mathcal{E}}(t,c,r)$ denotes the probability that $T' = t$, $C' = c$, $R' = r$, and $\mathcal{E}$ occurs.

It is easy to see (by summing over all t, c, and r) that

$$\hat{P}(\mathcal{E}) \;=\; \frac{1}{|\mathcal{C}|}$$

and hence that the distribution of the transcript (the first successful attempt) is

$$\hat{P}_{TCR}(t,c,r) \;=\; \frac{\hat{P}_{T'C'R'\mathcal{E}}(t,c,r)}{\hat{P}(\mathcal{E})} \;=\; p^P_T(t)\cdot p^P_{R|TC}(r,t,c)\cdot p^{V'}_{C|T}(c,t),$$

which is equal to $P_{TCR}(t,c,r)$, the transcript distribution in the real protocol execution.

To prove the second claim, namely that the simulator is efficient, note that in each attempt, the probability of creating a valid transcript is $\hat{P}(\mathcal{E}) = \frac{1}{|\mathcal{C}|}$. Therefore the expected number of simulation attempts is $|\mathcal{C}|$, which is polynomial by assumption, and each attempt is itself polynomial-time, so that the overall running time is polynomial. □

Typically, the ZK definition requires that the simulator be *expected* polynomial-time, and hence the above argument suffices. However, one can argue that one should rather consider *worst-case* polynomial-time. In this case one has to stop the simulation after a fixed (polynomial) number of attempts and therefore there is an exponentially small probability of failure. Hence the distribution of the simulated transcript is only very close (but not equal) to the distribution of the actual transcript. Hence the protocol is not perfectly ZK, but only *statistically ZK* (which is good enough).

3.5 Sequentially Composed ZK Protocols are ZK

As the final step, we show that a protocol composed of several c-simulatable three-move rounds with small challenge spaces is perfect zero-knowledge. The following lemma holds for any type of ZK protocol, not only the specific three-move protocol discussed here.

Lemma 3.3. *A sequence of s black-box ZK protocols is black-box ZK.*

Proof. Consider the sequential composition of the s protocols as a single protocol, and consider an arbitrary verifier V' for this composed protocol. In the i-th sub-protocol, V' chooses its messages (in our example the challenge c_i) depending on all the information is has seen up to that point, including the transcripts of the first $i-1$ sub-protocols. This can alternatively be understood as follows: After each sub-protocol execution, say after the first $i-1$ sub-protocols, V' determines an arbitrary (efficient) verifier V'_i for the i-th sub-protocol, where the *choice* of V'_i depends on the first $i-1$ transcripts, but V'_i itself is of a type only suitable for the i-th sub-protocol.

The overall simulator proceeds sub-protocol by sub-protocol. For each sub-protocol, say the i-th, it uses the simulator for that sub-protocol and the verifier V'_i for that sub-protocol to simulate a transcript. The simulation of the overall transcript is the concatenation of all simulated sub-protocol transcripts. □

The following theorem captures that most protocols discussed in the course are zero-knowledge, provided the challenge spaces are kept small.

Theorem 3.4. *An interactive proof (P,V) consisting of s c-simulatable three-move rounds, each with polynomial-size challenge space and with uniformly chosen challenge, is perfect zero-knowledge.*

Proof. Lemma 3.1 implies that each sub-protocol is HVZK. Lemma 3.2 implies that each such sub-protocol is black-box ZK. Hence application of Lemma 3.3 concludes the proof. □

References

[Mau15] Ueli Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. In *Des. Codes Cryptogr.*, pages 663–676, 2015.