CS 880: Quantum Information Processing 11/2/2010

Lecture 24: Zero Knowledge

Instructor: Dieter van Melkebeek Scribe: Tyson Williams

Last lecture, we discussed cryptographic protocols. In particular, we gave a quantum protocol for secret key exchange that is secure in an information theoretical sense provided there is a secure public classical channel. We also discussed bit commitment and showed that no quantum protocol has information theoretic security. Today we will discuss zero knowledge systems and give an example of a classical zero knowledge protocol that remains zero knowledge even in the quantum setting.

1 Interactive Proof Systems

To introduce zero knowledge, we first need to introduce the notion of an interactive proof system.

Definition 1. *An* interactive proof system *(IPS) for a language L is a protocol between a computationally unrestricted prover P and a probabilistic polynomial-time verifier V such that on input x, which is available to both parties,*

$$(\forall x \in L) \ \ \Pr\left[(V \leftrightarrow P)(x) \ \textit{accepts}\right] = 1 \qquad \text{(completeness)}$$

$$(\forall x \notin L)(\forall P') \ \ \Pr\left[(V \leftrightarrow P')(x) \ \textit{accepts}\right] \leq \frac{1}{2} \qquad \text{(soundness)}$$

where $(V \leftrightarrow P)(x)$ means "the verifier's view while running the protocol with P on input x."

The view of the verifier contains his coin flips, communication received from the prover, and communication sent to the prover (although this last type of communication can be recreated by the verifier using the same random bits). The completeness does not have to be perfect (that is, equal to 1) but we will only discuss such IPSs. If soundness of 1/2 is too high, just repeat the protocol a polynomial number of times for exponentially small soundness. The soundness condition must hold for all provers P', even ones that deviate from the protocol and try to convince the verifier that x is in the language when it is not.

An IPS is a generalization of the proof system associated with the class NP. For NP, the prover provides the witness as the proof and the verifier checks it deterministically in polynomial time. The difference here is that the verifier is allowed randomness and may interact with the prover several times. Without the randomness, multiple interactions is not more powerful.

An example of an IPS is, of course, standard NP proofs. An interesting example is GraphNonIsomorphism. We do not know if this problem is in NP, but it has a very simple IPS. A *yes* instance is a pair of graphs G_0 and G_1 that are not isomorphic. If the number of vertices in the graphs differ, then the verifier does not need the help of the prover, so let both graphs have n vertices. The verifier picks a bit $b \in \{0,1\}$ and $\sigma \in S_n$ (both uniformly at random), sends $\sigma(G_b)$ to the prover, and asks the prover to state which b he used. If the prover responds correctly, then the verifier accepts; otherwise, he rejects.

If the graphs are not isomorphic, then the prover is always be able to correctly identify b because $\sigma(G_b)$ is only isomorphic with G_b and not with $G_{\overline{b}}$. Thus, this IPS has perfect completeness. If the graphs are isomorphic, then the prover has no way of knowing which graph G_b was selected: Given any graph he received from the verifier, the probability that $b = 0$ is 50%. Whatever the prover does, he will be correct with probability $1/2$, which matches our soundness bound.

In general, any language L has an IPS iff L can be decided in polynomial space. That L has an IPS implies that $L \in$ PSPACE is easy. The other direction is a nontrivial result of complexity theory.

2 Classical Zero Knowledge

2.1 Informal Definition

A *zero knowledge interactive proof system* (ZKIPS) is a special kind of IPS. There is an additional condition, namely, when $x \in L$, the verifier does not learn anything other than being convinced that the x is indeed in L. In an IPS, the soundness condition protects the verifier from accepting an incorrect claim. In a ZKIPS, the new condition protects the prover from having to reveal any information (other than the correctness of the claim). When the prover follows the protocol for an input $x \in L$, the verifier will learn nothing beyond the fact that $x \in L$.

Most standard NP proofs are not zero knowledge under standard complexity theory assumptions like P $\neq$ NP. Consider the standard NP proof that a graph is 3-colorable. The proof is a 3-coloring. Intuitively, this is not a zero knowledge proof system because the verifier has learned more than just the fact that the graph is 3-colorable. The verifier now knows a 3-coloring, which he is unable to compute under the assumptions. Now the verifier can act as the prover and convince a different verifier that this graph is 3-colorable, something that he could not have done previously.

2.2 Motivation

A ZKIPS can be used for authentication. The most popular form of authentication today is via a password that is given to the verifier. Anyone who watches the prover enter the password has broken the security. They can now successfully authenticate as the prover. If the authentication used a ZKIPS and the prover follows the protocol, then anyone can watch the prover's interaction with the verifier, but they will learn nothing besides the fact that prover is who he says he is. In particular, no one will be able to authenticate as the prover (unless they were able to previously). This holds even for the computer system that the prover was using to communicate with the verifier.

Cryptographic protocols typically require secret keys for various parties. We would like to know that all parties correctly follow the cryptographic protocol, but to know this for certain requires knowledge of their secret key. Instead, we can phrase it as an NP question by saying, does there exist a secret key that would have caused the behavior we observed in the other party. Now we can use a ZKIPS to be convinced of this fact without learning the value of the secret key.

2.3 Formal Definition for a ZKIPS

We formalize the property of zero knowledge for an IPS in a strong way – that whatever can be efficiently computed from some prior knowledge and interaction with the honest prover on any input $x \in L$, can be efficiently computed from the prior knowledge without interaction with the prover.

Definition 2. *A* zero knowledge interactive proof system *(ZKIPS) for a language L is an interactive proof system between a prover P and a verifier V where for all probabilistic polynomial time verifiers V', there exists a probabilistic polynomial time simulator $S_{V'}$ such that*

$$(\forall x \in L)(\forall a \in \Sigma^*)\ (V' \leftrightarrow P)(x,a) \sim S_{V'}(x,a)$$

where the relation $\sim$ between the two distributions can take one of three meanings:

1. *the distributions are perfectly identical, which is called* perfect zero knowledge*,*
2. *the distributions are close in the L_1 norm, which is called* statistical zero knowledge*, or*
3. *the distributions are computationally indistinguishable to a probabilistic polynomial time machine, which is called* computational zero knowledge*.*

In this definition, $S_{V'}$ simulates the interaction between P and V', and a represents the prior knowledge.

Let's discuss why this definition is what we want. The only source the (dishonest) verifier V' has to gain any information is his view of the interaction with the prover, which is denoted by $(V' \leftrightarrow P)(x,a)$. However, the definition says that V' can instead ignore the prover and gain the same information by running $S_{V'}(x,a)$, which does not require interaction with the prover. The verifier is able to do this since $S_{V'}$ is also a probabilistic polynomial-time algorithm.

2.4 Examples of a ZKIPS

With such strong definitions, there is the risk that no examples exist. However, our definition is not that strong. We give two examples of ZKIPSs, one for GraphIsomorphism (unconditionally) and one for 3-Colorability (assuming bit commitment).

We intuitively argued above that the standard NP proof that a graph is 3-colorable is not zero knowledge. The same reasoning applies for the standard NP proof that two graphs are isomorphic, which is the isomorphism. Note that formally proving those claims would imply separations lie P $\neq$ NP, and is therefore beyond the current techniques of complexity theory. In contrast, proving that a protocol is zero knowledge just requires a construction like the ones below.

2.4.1 Graph Isomorphism has a ZKIPS

The input is two graphs G_0 and G_1, both with n vertices.

1. The prover picks $b \in \{0,1\}$ and $\sigma \in S_n$ uniformly at random and sends $H = \sigma(G_b)$ to the verifier.
2. The verifier picks $c \in \{0,1\}$ uniformly at random and sends it to the prover.
3. The prover picks some $\rho \in S_n$ and sends it to the verifier.
4. The verifier *accepts* iff $H = \rho(G_c)$.

Suppose the graphs are isomorphic, say $G_0 = \pi(G_1)$. Then the completeness is perfect because the prover will pick ρ to be

- σ when $b = c$,

- $\sigma \circ \pi$ when $0 = b \neq c = 1$, and
- $\sigma \circ \pi^{-1}$ when $1 = b \neq c = 0$.

The soundness is exactly 1/2 because the only way for the prover to send a valid isomorphism when the graphs are not isomorphic is when $b = c$, which happens with probability 1/2. We will show that this protocol is perfectly zero knowledge by giving the simulator $S_{V'}$ on inputs $\langle G_0, G_1 \rangle$ and a.

The simulator $S_{V'}(\langle G_0, G_1 \rangle, a)$ begins by running the same actions as the prover in step 1. In step 2, it behaves like V' to get the bit c. If $b = c$, output (H, c, σ). If $b \neq c$, start over.

When $S_{V'}(\langle G_0, G_1 \rangle, a)$ succeeds and gets $b = c$, the output distributions are identical since $S_{V'}(\langle G_0, G_1 \rangle, a)$ followed the protocol. Conditioned on H and c, the probability that $b = c$ is 1/2, so the expected number of iterations until $S_{V'}(\langle G_0, G_1 \rangle, a)$ succeeds is 2. Thus we have a probabilistic, expected polynomial time simulator, which is good enough to achieve prefect zero knowledge. If a definite runtime is required (instead of an expected one), then we can modify $S_{V'}(\langle G_0, G_1 \rangle, a)$ to obtain statistical zero knowledge by iterating some large but constant number of times before outputting some fixed string if all iterations failed. This distribution will be very close to the actual distribution created by the protocol as required.

2.4.2 3-Colorability has a ZKIPS

A ZKIPS exists for 3-Colorability assuming bit commitment. Last lecture, we showed that no bit commitment protocol has information theoretic security, but such protocols do exists for the classical computational setting under computational assumptions, like the existence of one-way functions. Note that it is better to base a ZKIPS on hard problems because the zero knowledge property only guarantees that a computationally efficient party cannot do anything more after running the protocol than before. If the underlying computational problem is easy, then there is no need for interaction to break the security. For that reason, zero knowledge protocols based on 3-Colorability are safer than those based on GraphIsomorphism, as the former problem is NP-complete but the latter is believed not to be.

Suppose the prover has a 3-coloring $\gamma : V(G) \to \{R, Y, B\}$ of the input graph G. The protocol then proceeds as follows.

1. The prover selects a uniformly random permutation π of $\{R, Y, B\}$, commits to $\pi(\gamma(v))$ for all $v \in V(G)$, and sends those commitments to the verifier using the bit commitment scheme.
2. The verifier then selects $(u, v) \in E(G)$ uniformly at random and sends the edge to the prover.
3. The prover checks that (u, v) is indeed an edge in E. If $(u, v) \notin E$, the prover *aborts*. If $(u, v) \in E$, then the prover continues the protocol by revealing $a = \pi(\gamma(u))$ and $b = \pi(\gamma(v))$.
4. The verifier *accepts* iff $a, b \in \{R, Y, B\}$ and $a \neq b$.

If γ is a valid 3-coloring, then the verifier will always accept since the colors assigned to adjacent vertices are different choices of R, G, and B, so we have perfect completeness. If G is not 3-colorable, then there exists at least one edge where the incident vertices have the same color or one has an invalid color. Catching the prover in the case that all colors are valid but there is exactly one edge with incident vertices of the same color is the harder case to detect, which happens with probability $\frac{1}{|E|}$, so our soundness is at most $\frac{|E|-1}{|E|}$. This argument also relies on the provers bit commitments. After the verifier picks the edge (u, v), we cannot allow the prover to change to a coloring that is locally valid. In order to boost our confidence, we can repeat this protocol poly($|E|$) times to achieve another protocol with soundness of at most 1/2. Furthermore, this protocol is zero knowledge, which we show by constructing the simulator $S_{V'}$ on inputs G and a.

The simulator $S_{V'}(G, a)$ begins by running the same actions as the prover in step 1. In step 2, it behaves like V' to get the pair (u, v). If $(u, v) \notin E(G)$, *abort*. If $(u, v) \in E(G)$, then output two distinct colors from $\{R, Y, B\}$ uniformly at random.

When the verifier does not cheat and selects a pair of vertices that form an edge in G, two colors are revealed. Conditioned on the bit commitments and the edge (u, v), these two colors are fixed. However, these two colors are computationally indistinguishable from two distinct colors selected uniformly at random because the verifier does not have the computational ability to break the security of the bit commitments. Thus, this simulator proves that the protocol is computational zero knowledge.

Notice how simple this ZKIPS is. Every step only contains basic computations. This protocol could easily be implemented on a smart card. Also note that it is crucial that the prover check that the verifier's pair (u, v) is an edge. Without the check, this protocol is zero knowledge iff NP = RP.

3 Quantum ZKIPS

In a quantum IPS, the prover and verifier can perform quantum computations and their communication can be quantum. The prior knowledge will now be modeled by a quantum register $|\alpha\rangle$. We will now prove the following theorem.

Theorem 1. *The zero knowledge interactive proof system for* GraphIsomorphism *remains perfectly zero knowledge in the quantum setting. Furthermore, the simulator runs in worst-case polynomial time.*

A theorem like this is important because it says that the prover can continue to use a cheap, common classical computer and remain secure against a dishonest verifier who has the power of quantum.

Proof. Since the verifier can observe every message from the prover, the arguments for the completeness and soundness from the classical setting still hold. What remains is to show that this protocol is still zero knowledge, which is not obvious.

Why does our argument from the classical setting fail? It is because of the prior knowledge. The standard simulation procedure runs the basic simulator until the first success. For each trial we need a fresh copy of $|\alpha\rangle$, but the no cloning theorem forbids copying $|\alpha\rangle$. Another idea is to run the protocol backwards and try to recover $|\alpha\rangle$. However, checking for success involves a measurement, so we will not be able to recover $|\alpha\rangle$ exactly. We will show that the modified state of $|\alpha\rangle$ obtained by rewinding after a failed attempt nevertheless allows us to rerun the basic protocol with high probability of success, and keep doing so until the first success. The key property we need of the classical zero knowledge protocol is that the probability of success of the basic simulator is independent of $|\alpha\rangle$, namely $p = 1/2$ in the case of the protocol for graph isomorphism.

By assuming that $S_{V'}$ postpones all measurements until the end, we can represent $S_{V'}$ as a unitary matrix U applied to $|\alpha\rangle |0^m\rangle$ followed by a projective measurement (P_0, P_1), where P_1 corresponds to success. Of course U also acts on the input, but this will not affect the analysis.

For all $|\alpha\rangle$, we have

$$\| P_1 U |\alpha\rangle |0^m\rangle \|_2^2 = p. \qquad \text{(probability of success)}$$

and

$$\| P_0 U |\alpha\rangle |0^m\rangle \|_2^2 = 1 - p. \qquad \text{(probability of failure)}$$

We can rewrite the left-hand side of the latter equation as

$$\langle\alpha| \langle 0^m| U^\dagger P_0^\dagger P_0 U |\alpha\rangle |0^m\rangle = \langle\alpha| \langle 0^m| U^\dagger P_0^2 U |\alpha\rangle |0^m\rangle = \langle\alpha| \langle 0^m| U^\dagger P_0 U |\alpha\rangle |0^m\rangle ,$$

because a projective matrix is Hermitian and projecting twice is the same as once. Thus, we have that for all $|\alpha\rangle$,

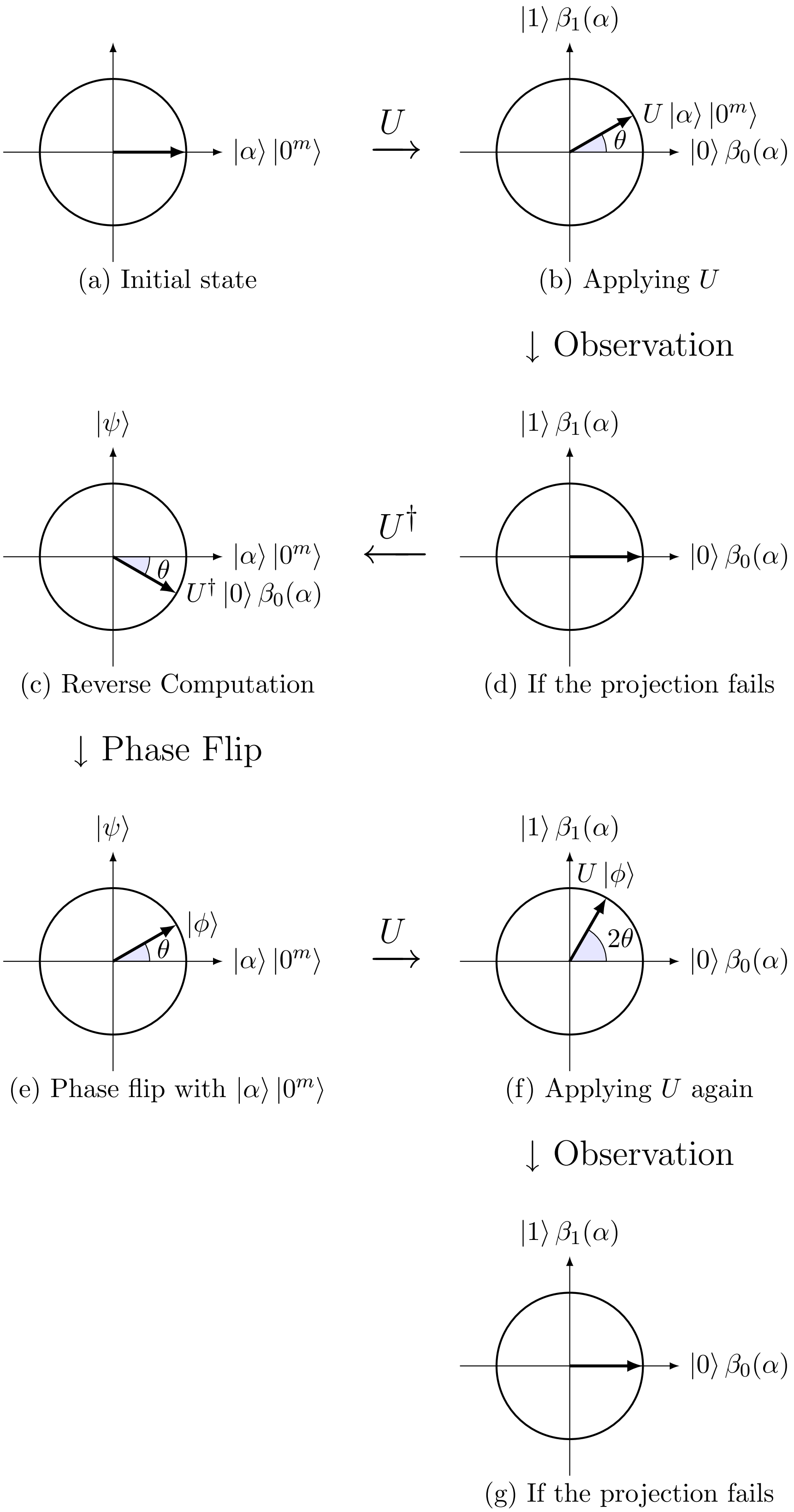
$$(I \otimes |0^m\rangle) U^\dagger P_0 U (I \otimes \langle 0^m|) |\alpha\rangle = (1-p) |\alpha\rangle .$$

The operator $(I \otimes |0^m\rangle) U^\dagger P_0 U (I \otimes \langle 0^m|)$ does the following to $|\alpha\rangle$: It takes $|\alpha\rangle$, extends it by m zeros, applies $U^\dagger P_0 U$, and extracts the components that end in m zeros. This operator is Hermitian and maps every $|\alpha\rangle$ to $(1-p)|\alpha\rangle$. The only way that can happen is if

$$(I \otimes |0^m\rangle) U^\dagger P_0 U (I \otimes \langle 0^m|) = (1-p)I.$$

This follows because this operator, being Hermitian, has a full basis of eigenvectors. Every eigenvalue must be $(1-p)$. Another way to view all of this is that the projection of $U^* P_0 U |\alpha\rangle |0^m\rangle$ onto components with 0^m at the end is $(1-p)|\alpha\rangle$, which is parallel to $|\alpha\rangle$. The latter is the property we will actually use. Note that the independence of the success probability on the state $|\alpha\rangle$ is what allowed us to argue it.

Now let's see, via a two dimensional diagram, what happens when we run our simulator $S_{V'}$. We start with the vector $|\alpha\rangle |0^m\rangle$ which we place on an axis (see Figure 1(a)). After applying U, we are in a state in which the observation can either lead to success denoted by $|1\rangle$ or failure denoted by $|0\rangle$ (see Figure 1(b)). If we project and measure a 1, then we are done, so assume that we measure a 0. This means we are now (after normalizing) in the state $|0\rangle \beta_0(\alpha)$ (see Figure 1(d)). Since we failed, we are going to try to return to the initial state by applying $U^\dagger$. There is a vector that we can pick for the vertical axis in Figure 1(a) so that $U^\dagger |0\rangle \beta_0(\alpha)$ lies in the plane of the figure. From the first plane in Figure 1(a) to the second plane in Figure 1(b), the unitary operator U caused a rotation by θ. Thus, going in the reverse direction will send us back by θ (see Figure 1(c)). Now look at the parts of $U^\dagger |0\rangle \beta_0(\alpha)$ that end in 0^m. We know that this part is parallel to $|\alpha\rangle$, so if we do a phase flip for all of the components which do not end in 0^m, then we reflect across the $|\alpha\rangle |0^m\rangle$ axis and get some state $|\phi\rangle$ (see Figure 1(e)). At this point, have a state that is different than the state $|\alpha\rangle |0^m\rangle$ we started from, but we can still use it in the simulation. If we apply U to $|\phi\rangle$, we return to the second diagram at an angle of 2θ (see Figure 1(f)). If we fail again, we return to the state in Figure 1(c) and repeat the above process (see Figure 1(g)).


$|\alpha\rangle\,|0^m\rangle$
$\xrightarrow{U}$
$|1\rangle\,\beta_1(\alpha)$
$U\,|\alpha\rangle\,|0^m\rangle$
θ
$|0\rangle\,\beta_0(\alpha)$
$\downarrow$ Observation
$|\psi\rangle$
$|\alpha\rangle\,|0^m\rangle$
θ
$U^\dagger\,|0\rangle\,\beta_0(\alpha)$
$\xleftarrow{U^\dagger}$
$|1\rangle\,\beta_1(\alpha)$
$|0\rangle\,\beta_0(\alpha)$
$\downarrow$ Phase Flip
$|\psi\rangle$
$|\phi\rangle$
θ
$|\alpha\rangle\,|0^m\rangle$
$\xrightarrow{U}$
$|1\rangle\,\beta_1(\alpha)$
$U\,|\phi\rangle$
2θ
$|0\rangle\,\beta_0(\alpha)$
$\downarrow$ Observation
$|1\rangle\,\beta_1(\alpha)$
$|0\rangle\,\beta_0(\alpha)$

(a) Initial state

(b) Applying U

(c) Reverse Computation

(d) If the projection fails

(e) Phase flip with $|\alpha\rangle\,|0^m\rangle$

(f) Applying U again

(g) If the projection fails

Figure 1: Two dimensional depiction of the simulator $S_{V'}$

Since the probability of success is the square of the projection on the vertical axis, the probability success in the first trial is

$$\Pr\left[\text{success in first trial}\right] = \sin^2\theta = p,$$

and the probability of success in any subsequent trials is

$$\Pr\left[\text{success in any subsequent trial}\right] = \sin^2(2\theta) = 4p(1-p).$$

In the case of graph isomorphism, the probability of success in the first trial is $p = 1/2$, and in the second trial is $4p(1-p) = 1$, so our simulator always halts and the running time is polynomial. Like in the classical setting, the output distribution on success is identical to the view of the verifier. Thus, the protocol is perfect zero knowledge. □

4 Next Time

In the next lecture, we will continue our discussion of quantum interactive proofs. After this, we will begin talking about error correction.