Quantum Computing 4/28/2022

Lecture 25: Post-Quantum Cryptography

Instructor: Dieter van Melkebeek Scribe: Jason Mohoney

Last lecture, we finished our discussion on Shor's algorithm for integer factorization and computing discrete logarithms. We covered the widely used RSA cryptosystem and the Diffie-Hellman key exchange protocol, and their vulnerabilities to quantum adversaries. We ended by discussing bit commitment and showed that no quantum protocol can achieve information theoretically secure perfect bit commitment. Today we cover *interactive proof systems* and a class of interactive proofs, known as *zero knowledge proofs*. We show that certain classical protocols based on these proof systems are robust against quantum adversaries. We end by discussing the computational power of classical and quantum interactive proof systems.

1 Interactive Proof Systems

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,

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

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

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 the common input x, the verifier's randomness (coin tosses), communication received from the prover, and communication sent to the prover. The completeness requirement above (with a right-hand side of 1) is known as *perfect completeness*. Sometimes the requirement is relaxed and the left-hand side is only required to be at least a some specified quantity $c > s$, but all the IPSs we consider have perfect completeness ($c = 1$). 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. In cryptographic settings we want the soundness parameter s to be negligible, i.e., $s = O(1/n^c)$ for every constant c so that an adversary running in polynomial time cannot break it. We can achieve better soundness, even starting from $s < 1 - \frac{1}{n^d}$ for any positive constant d by running poly(n) independent trials and accepting only if all accepted to get $s = O(\frac{1}{2^{n^c}})$ for any fixed constant c.

This system is a generalization of NP, the difference being that the verifier is allowed randomness and may interact with the prover several times. Without the randomness, multiple interactions is not more powerful. We will see later that allowing both randomness and multiple interactions makes a huge difference in power.

1.1 IPS for Graph Non-Isomorphism

Standard NP proofs are trivial examples of IPS as they use neither randomness nor multiple interactions: the prover sends a candidate witness to the verifier, who then checks the validity of

the witness in deterministic polynomial time. A more interesting example is the GRAPH NON-ISOMORPHISM problem. We do not know if this problem is in NP, but it has a very simple IPS. In this problem, a *yes* instance is a pair of graphs G_0 and G_1 that are not isomorphic, in other words the language is $L = \{(G_0, G_1) | G_0 \not\equiv G_1\}$. Let both graphs have n vertices, otherwise they will be trivially non-isomorphic. The protocol is then:

1. **Challenge**: V picks a bit $b \in_u \{0,1\}$, a permutation $\pi \in_u S_n$ and sends $H \doteq \pi(G_b)$. Here $\in_u$ denotes picking an element uniformly at random.
2. **Response**: P finds $a \in \{0,1\}$ such that $H \equiv G_a$ and then sends a.
3. **Decision**: V accepts iff $a = b$.

If the graphs are not isomorphic, then the prover P is always able to correctly identify b because $\pi(G_b)$ is only isomorphic with G_b and not with G_{1-b}. Thus, this IPS has perfect completeness. If the graphs are isomorphic, then P has no way of knowing which graph G_b was selected: Given any graph P received from the verifier, the probability that $b = 0$ is 50%. Whatever the prover does, they will be correct with probability 1/2. As we saw in section 1, repeated sampling can decrease this soundness to $O(\frac{1}{2^{\text{poly}(n)}})$.

2 Zero Knowledge Proofs

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 input 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 RP $\neq$ NP. Consider the standard NP proof that a graph is 3-colorable. The proof requires demonstrating a valid 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 they would be unable to compute in polynomial time. Now the verifier can act as the prover and convince a different verifier that this graph is 3-colorable, something that they could not have done previously.

2.2 Motivation

A ZKIPS can be used for authentication. Consider for example 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 the prover is who they say they are. 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.

Another motivation deals with adherence to cryptographic protocols. 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 *simulated* from 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

$$(\forall x \in L)(\forall h \in \{0,1\}^*)\ (V' \leftrightarrow P)(x,h) \sim S_{V'}(x,h)$$

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 statistical distance, which is called *statistical zero knowledge*,
3. the distributions are computationally indistinguishable in probabilistic polynomial time, which is called *computational zero knowledge*.

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

Let's discuss why this definition is what we want. The only source a (dishonest) verifier V' has to gain any information is their view of the interaction with the prover, which is denoted by $(V' \leftrightarrow P)(x,h)$. However, the definition says that V' can instead ignore the prover and gain the same information by running $S_{V'}(x,h)$, 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 ZKIPS for Graph Isomophism

Above we intuitively argued that the standard NP proof system for 3-COLORABILITY is not zero knowledge, as the 3-coloring is revealed to the verifier. More precisely, if the proof system were zero knowledge, then we would obtain an RP algorithm for 3-COLORABILITY, and thus for all of NP (by the NP-completeness of 3-COLORABILITY). That NP is included in RP is considered very unlikely. On the other hand, we cannot rule it out. If fact, one can also show that the hypothesis NP in RP implies that the standard NP proof system is zero knowledge, for formally proving that the standard NP proof system is not zero knowledge would show that NP is not in RP, which is beyond the scope of current techniques in complexity theory. In contrast, proving that a protocol *is* zero knowledge just requires demonstrating a construction like the ones below.

For a protocol for the GRAPH ISOMORPHISM problem, the input is two graphs G_0 and G_1, both with n vertices and is as follows:

1. **Commitment**: The prover picks $b \in_u \{0,1\}$ and a permutation $\pi \in_u S_n$ uniformly at random and sends $H = \pi(G_b)$ to the verifier.

2. **Challenge**: The verifier picks $a \in_u \{0,1\}$ and sends it to the prover.

3. **Response**: The prover picks $\sigma \in_u S_n$ such that $\sigma(G_a) = H$ and sends it to the verifier.

4. **Decision**: The verifier *accepts* iff $H = \sigma(G_a)$.

In contrast to the IPS for Graph Non-Isomorphism, this protocol gives more control to the prover by allowing them to chose what information they release to the verifier in the commitment stage. Protocols that follow this structure are known as Σ-protocols, although we will not formally define them here. To analyze this protocol, suppose the graphs are isomorphic, and say $G_0 = \pi(G_1)$. Then the completeness is perfect because both graphs are isomorphic to H. 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 $b = a$, which happens with probability $1/2$. We will show that this protocol is perfectly zero knowledge by giving the simulator $S_{V'}$ on inputs (G_0, G_1) and history h.

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

When $S_{V'}((G_0, G_1), h)$ succeeds and gets $a = b$, we know that the distribution of π is the same as the distribution of σ conditioned on H and a, so the output distributions are identical. Conditioned on H and a, the probability that $a = b$ is $1/2$, so the expected number of iterations until $S_{V'}((G_0, G_1), h)$ succeeds is 2. Thus we have a probabilistic, expected polynomial time simulator, which is good enough to achieve prefect zero knowledge.

2.5 Robustness against Quantum Adversaries

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 $|\psi\rangle$. We will now prove the following theorem.

Theorem 1. The zero knowledge interactive proof system based on the Graph Isomorphism problem remains perfectly zero knowledge in the quantum setting. Furthermore, the running time of the simulator is polynomially bounded in the worst case.

This theorem is important because it says that the prover can continue to use a cheap, common classical computer and remain secure against a dishonest verifier with quantum capabilities.

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 $|\psi\rangle$, but the no cloning theorem forbids copying $|\psi\rangle$. Another idea is to run the protocol backwards and try to recover $|\psi\rangle$. However, checking for success involves a measurement, so we will not be able to recover $|\psi\rangle$ exactly.

However, we know that the probability of success of the basic simulator is independent of $|\psi\rangle$, namely $p = 1/2$ in the case of the protocol for graph isomorphism. This means we can use oblivious

amplitude amplification, where we don't know $|\psi\rangle$ but do know that the sucess probability is the same for every $|\psi\rangle$. In addition, because we know p, we can guarantee success and do so in worst-case polynomial time (compared to average-case polynomial time in the classical setting). □

2.6 ZKIPS for 3-Colorability

From a cryptographic standpoint, 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 problems like 3-Colorability are safer than those based on Graph Isomorphism, as the former problem is NP-complete but the latter is believed not to be.

To address this we will now show that 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 exist for the classical computational setting under computational assumptions, like the existence of one-way functions.

The language is $L = \{G = (V, E) | G \text{ is 3-colorable}\}$, and the protocol is as follows:

1. **Commitment**: The prover knows a 3-coloring $\gamma : V \to [3]$, and then picks a permutation $\pi \in_u S_3$ and sends a commitment to the coloring $\kappa = \pi \circ \gamma$.
2. **Challenge**: The verifier then selects $e = (v, w) \in_u E$ and sends e to the prover.
3. **Response**: If $(v, w) \in E$, then the prover reveals $\kappa(v)$ and $\kappa(w)$.
4. **Decision**: The verifier *accepts* iff $\kappa(v) \neq \kappa(w)$.

If γ is a valid 3-coloring, then the verifier will always accept, 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. Our soundness parameter is then at most $s = 1 - \frac{1}{|E|}$, which we can boost to $O(1/2^{n^c})$. This argument also relies on the prover's bit commitments. After the verifier picks the edge (v, w), we cannot allow the prover to change to a coloring that is locally valid. Furthermore, this protocol is zero knowledge, which we show by constructing the simulator $S_{V'}$ on inputs G and h.

The simulator $S_{V'}(G, h)$ begins by running the same actions as the prover in step 1. In step 2, it behaves like V' to get the pair (v, w). If $(v, w) \in E$, then output two distinct colors 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 (v, w), these two colors are fixed. However, these two colors are computationally indistinguishable from two distinct colors selected uniformly at random because they were permuted by a randomly chosen π, which the verifier cannot predict because it 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.

3 Power of Interactive Proofs

We end with some discussion of the computational power of interactive proof systems.

The first result is that the set of all languages that have a classical IPS, IP, is equal to the set of all languages that can be decided using a polynomial amount of space, PSPACE. This is a well-known result in complexity theory. In fact, this still holds in the quantum setting: the set of languages with quantum IPS, QIP, equals PSPACE as well. An additional property in the quantum setting is a canonical form for all quantum interactive proofs, which is a quantum version of the Σ-protocol that we've seen in the examples above:

1. **Commitment**: P sends a register X of qubits.
2. **Challenge**: V picks $b \in_u \{0, 1\}$ and sends b.
3. **Response**: P sends register Y.
4. **Decision**: V decides whether to accept.

3.1 Power of Multi-Prover Interactive Proofs

A multi-prover interactive proof (MIP) is one where the verifier can interact with multiple provers that cannot communicate with each other. In fact, two provers is enough to get the full power. The addition of a second prover makes a significant change in the complexity. The intuition behind this is similar to what the police use when they interrogate suspects, the verifier can play the provers off each other and harness the fact that they cannot coordinate once the interrogation has started. The complexity class of such problems MIP is actually equal to the set of problems that can be solved in nondeterministic exponential time, NEXP. Note that, in contrast to PSPACE, NEXP is provably larger than NP. The same holds if we allow the verifiers and provers to have quantum capabilities, so QMIP = NEXP. A caveat with this is that the provers may not have prior entanglement. If we do allow this, we get the complexity class QMIP*, which is equal to the class of classical MIPs where the provers can have prior entanglement, MIP*. This result was known for a while, but remarkably in 2020 [JNV$^+$20] it was shown that MIP* = RE, the class of recursively enumerable languages. This class is enormous, it is equivalent to the halting problem which is infeasible to solve, even if we had all the computational power in the world.

References

[JNV$^+$20] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*=re, 2020. doi: 10.48550/ARXIV.2001.04383.