**Quantum Algorithms** 4/25/2023

# Lecture 25: Post-Quantum Cryptography

Instructor: Dieter van Melkebeek

Post-quantum cryptography refers to showing that certain classical cryptographic remain secure in the quantum setting. Such results are pertinent in the current era, where quantum computers are not practically available but some entities may have access to them now or in the near future. In this lecture we show that classical zero-knowledge systems like the one for graph isomorphism are robust against quantum adversaries. The argument hinges on oblivious amplitude amplification, for which this setting represents a natural application. We start by introducing interactive proofs and their zero-knowledge variant, and end by discussing the computational power of classical and quantum interactive proof systems.

## 1 Interactive proof systems

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 $(V \leftrightarrow P)(x)$ denotes 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), and the communication received from 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 some positive constant $d$ by running $\text{poly}(n)$ independent trials and accepting only if all accepted to get $s = O(\frac{1}{2^{n^c}})$ for any fixed constant $c$.

Standard NP proofs are trivial examples of an 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. In contrast, in an IPS the verifier is allowed randomness and may interact with the prover several times. Without the randomness, multiple interactions is not more powerful. However, the combination of randomness and interaction turns out to be very powerful, as we will see later.

**IPS for graph non-isomorphism.** An example of a problem that is not known to be in NP but has a simple IPS is graph non-isomorphism (GNI). In this decision problem, a *yes* instance is a pair

of graphs $G_0$ and $G_1$ that are not isomorphic, in other words the language is $L = \{(G_0, G_1) \mid G_0 \not\equiv G_1\}$. 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 $H \doteq \pi(G_b)$.
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, a polynomial number of repetitions of the protocol improve the soundness to $O(\frac{1}{2^{\text{poly}(n)}})$.

## 2 Zero knowledge proofs

The goal of zero-knowledge is intuitive and can be formalized adequately in complexity-theoretic terms. We start with an informal discussion.

**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 beyond the fact 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 revealing any information other than the correctness of the claim that $x \in L$. 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 NP not having polynomial time randomized algorithms with bounded error. 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 themselves. 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.

**Motivation.** A ZKIPS can be used for authentication and provides a mechanism that is superior traditional passwords. In the latter mechanism the prover provides a password 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 uses 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 ensure that all parties correctly

follow the cryptographic protocol but do so without anyone revealing their secret keys. We can phrase adherence as an NP question by saying, does there exist a secret key that would have caused the behavior we observed in the other party. If we have a ZKIPS for NP (say, for 3-coloring), we can use it to be convinced of adherence without learning the value of the secret key.

**Formal definition.** 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 1.** 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. Among other things, the history can include transcripts from prior runs of the protocol on the same input.

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. Thus, on an input $x \in L$, the verifier $V'$ has learned nothing beyond the fact that $x \in L$.

Above we informally argued that the standard NP proof system for 3-colorability is not zero knowledge, as the 3-coloring is revealed to the verifier. Given our formal definition, one can show that, if the proof system were zero knowledge, then we would obtain a randomized algorithm polynomial-time algorithm with bounded error for 3-colorability, and thus for all of NP (by the NP-completeness of 3-colorability). This is considered unlikely; it would allow us to solve NP-complete problems efficiently (using randomness). On the other hand, we cannot rule it out. In fact, one can also show that the hypothesis that NP has polynomial-time randomized algorithms with bounded error implies that the standard NP proof system is zero knowledge. Thus, we cannot hope to show whether or not the standard NP-proof for 3-colorability is zero-knowledge without resolving a major open problem in complexity theory. In general, showing that protocols are not zero-knowledge is difficult. In contrast, proving that a well-designed protocol *is* zero-knowledge if often feasible and just requires demonstrating a construction like the ones below. We start with one for the complement of graph non-isomorphism.

**ZKIPS for graph isomorphism.** Consider the following protocol for graph isomorphism (GI) on an instance consisting of two graphs $G_0$ and $G_1$, both with $n$ vertices. In the case where $G_0 \equiv G_1$, we assume that $P$ has figured out permutations $\rho_{a\to b} \in S_n$ for $a, b \in \{0,1\}$ with $a \neq b$ such that $G_b = \rho_{a\to b}(G_a)$. In the case where $G_0 \not\equiv G_1$, these permutations are arbitrary.

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 sets $\sigma = \pi$ if $a = b$, and $\sigma = \pi \circ \rho_{a\to b}$, otherwise, and sends $\sigma$ to the verifier.
4. **Decision**: The verifier *accepts* iff $H = \sigma(G_a)$.

In contrast to the IPS for GNI, this protocol for GI gives more control to the prover on what information they release to the verifier. In particular, there is an additional initial phase, the commitment phase, in which the prover picks the graph $H$, which was previously picked by the verifier in the challenge phase. Correspondingly, in the challenge phase of the new protocol, the verifier only picks a bit $a$. Protocols that follow this structure are common and referred to as $\Sigma$-protocols, although we will not formally define them here.

Let us first argue that the protocol under consideration is a valid IPS for GI. Completeness is perfect because if $G_0 \equiv G_1$, then $H$ is isomorphic to both $G_0$ and $G_1$, so no matter what value $V$ picks for $a$, $P$ is always able to send an isomorphism from $G_a$ to $H$. The soundness is exactly $1/2$ because, in case $G_0 \not\equiv G_1$, the only way for the prover to send a valid isomorphism is when $a = b$, which happens with probability $1/2$.

We now show that the protocol is perfectly zero knowledge by developing, for a given verifier $V'$, a simulator $S_{V'}$ that, on inputs $(G_0, G_1)$ with $G_0 \equiv G_1$ and history $h$, produces a random variable with exactly the same distribution as the view $(V' \leftrightarrow P)(G_0, G_1; h)$. What is the view of the verifier in this protocol? It consists of the graph $H$ that is sent by the prover in step 1, the bit $a$ that is picked by the verifier in step 2, and the permutation $\sigma$ that is sent by the prover in step 3. Note that conditioned on $H$ and $a$, the distribution of $(b, \sigma)$ is uniform over $\{0,1\} \times \{\rho \in S_n : H = \rho(G_a)\}$. This implies that $\Pr[a = b] = 1/2$, and that the distribution of the view conditioned on $a = b$ is the same as without the conditioning.

Note that when $a = b$, the view is simply $(H, a, \pi)$. This allows us to build the following simulator. $S_{V'}((G_0, G_1), h)$ begins by running the same actions as the prover in step 1, namely picking $b \in_u \{0,1\}$ and $\pi \in_u S_n$, and setting $H = \pi(G_b)$. In step 2, it runs $V'$ to obtain the bit $a$. If $a$ happens to equal $b$, it is able to execute step 3 and complete the view by outputting $(H, a, \pi)$. If $a \neq b$, the simulator starts over.

We already argued that the distribution of the view conditioned on $a = b$ is the same as without the conditioning. If follows that the output of the simulator is correct. Moreover, as $\Pr[a = b] = 1/2$, the expected number of iterations until the simulator succeeds is 2. Thus, assuming $S'$ runs in probabilistic polynomial time, we the simulator $S_{V'}$ runs in expected polynomial time.

**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 is now modeled by a quantum register $|\psi\rangle$. We show the following result.

**Theorem 1.** *The classical zero knowledge interactive proof system for graph isomorphism 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 measure every message from the prover before using it, 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 success 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). □

**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 3-colorability are safer than those based on graph isomorphism, as the former problem is NP-hard but the latter is believed not to be.

To address this we now show that a ZKIPS exists for 3-colorability assuming bit commitment, which is known to be possible classically if one-way functions exist. Due to the bit commitments, the protocol is only computationally zero knowledge.

The language is $L = \{G = (V, E) : G \text{ is 3-colorable}\}$, and the protocol is as follows, where we assume that the prover has figured out a valid 3-coloring $\gamma : V \to [3]$ in case the graph is 3-colorable; otherwise $\gamma$ denotes an arbitrary function $\gamma : V \to [3]$.

1. **Commitment**: The prover picks a permutation $\pi \in_u S_3$ of the three colors and sends a commitment to the coloring $\kappa = \pi \circ \gamma$.

2. **Challenge**: The verifier 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 $\kappa(w)$. Otherwise, abort the protocol.

4. **Decision**: The verifier *accepts* iff $\kappa(v) \neq \kappa(w)$.

If $\gamma$ is a valid 3-coloring, then the verifier always accepts, 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 the coloring of $u$ and $v$.

Furthermore, the 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 $V'$ sends an edge $(u, v) \in E$, without the random permutation $\pi$, the honest prover would reveal two fixed distinct colors. Due the random permutation $pi$, these two colors are mapped to two distinct colors chosen uniformly at random, which his the same as the simulator. By the properties of the bit commitment protocol, the actual distribution and the simulator's distribution are computationally indistinguishable.

# 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. Note that in classical interactive proofs, the power of the prover can even include quantum. This is because by measuring the qubits received from the prover, the verifier can effectively make the prover behave classically. When we allow also the verifier to use quantum operations, the power of interactive proofs could go up, but it turns out it does not: 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 $\Sigma$-protocol that mentioned 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.

**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 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: 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 coincides with 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. *CoRR*, abs/2001.04383, 2020.