Frontiers of Quantum Complexity and Cryptography Spring 2022

Lecture 10

Lecturer: Henry Yuen *Scribes: Yunhao Xing*

1 Overview

In the last lecture we talked about the idea of *pseudorandom quantum states* (PRS) and presented a construction of PRS, assuming the existence of pseudorandom functions (PRF) from classical cryptography.

In this lecture we talk about how the landscape of different cryptographic primitives are organized, and then discuss one cryptographic application of PRS.

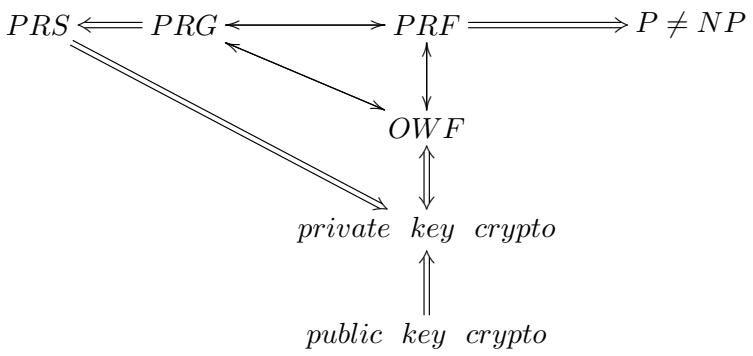
2 Landscape of Cryptography

Over the last four decades, computer scientists have developed a rich understanding of how different *complexity assumptions*, *cryptographic primitives*, and *cryptographic tasks* relate to each other. A *complexity assumption* is an assumption about computational complexity classes. The most important complexity assumption is the famous $P \neq NP$ conjecture. This is a holy grail of computer science and mathematics, and it appears that we are very far from being able to settle it one way or another. However the prevailing belief is that P is indeed different from NP.

A *cryptographic primitive* is a fundamental object that turns out to be extremely useful in a variety of cryptographic settings. In this context, we think of objects such as pseudorandom functions (PRF), pseudorandom generators (PRG), and one-way functions (OWF) as being cryptographic primitives. Pseudorandom states (PRS) are an example of a new kind of cryptographic primitive.

A *cryptographic task* is a higher-level task that you would naturally want to perform, such as encryption, or digitally signing a message, or performing a distributed computation in a private manner. There is a dazzling array of different cryptographic tasks one might want to perform. There are two broad categories of tasks that people often consider: *private-key cryptography* and *public-key cryptography*. Private-key cryptography involves things like encryption where both sender and receiver share a secret key that is unknown to anyone else. Public-key cryptography is about harder tasks like performing encryption using a public key, that can only be decrypted using a private key (think RSA).

The following diagram illustrates the known relationships between them. An arrow from A to B indicates that "A implies B". If there is no arrow from B to A, that indicates that it is unknown (and indeed, unlikely) that B implies A.



PRF, PRG, and OWF are equivalent primitives, meaning if you can build one then you can build the others. Private-key cryptography tasks are all equivalent to the existence of PRG/PRF/OWFs. In other words, one can construct private-key cryptosystems using OWF. Conversely, if you construct a (classical) private-key cryptosystem, it *must've* used a PRG/PRF/OWF somewhere inside (even if you didn't use one explicitly!).

The same is true for public-key cryptosystems – they must all use OWFs – but it is not known how to construct public-key cryptosystems from OWFs only. In fact, it is believed that public-key cryptosystems necessarily require (in a sense) harder assumptions than the existence of OWFs [5].

All of the aforementioned things – primitives and private/public-key cryptosystems – imply $P \neq NP$. It is not known whether $P \neq NP$ implies OWFs, though – this is a big open question in complexity theory and cryptography.

Where does PRS fit in? From PRF/PRG/OWF, it is possible to build PRS as we saw in the previous class. However, it is not known how to build a OWF/PRG/PRF from PRS, and in fact it was recently shown that it potentially could be the case that $P = NP$ (and thus OWFs do not exist, and thus virtually all of classical cryptography is not possible), yet PRS still exist. What is meant by "potentially could be the case" is that there is no *black-box* construction of OWFs from PRS; this is a result of Kretschmer [4].

The concept of pseudorandom states was introduced by Ji, Liu and Song in 2018 [1]; they also showed that PRGs could be used to construct PRS. However, the usefulness of PRS to achieve cryptographic tasks was left as an open question. They presented an application to private-key quantum money but that was it.

Recently, Ananth, Qian and Yuen [2] and Morimae and Yamakawa [3] showed that one can in fact use PRS to build interesting private-key crypto primitives, *without* using any OWFs. To reconcile this with the previous claim that private-key crypto necessitates the use of OWFs, in these PRS-based cryptosystems, the honest parties must be able to manipulate quantum information and transmit it to each other (whereas in the classical case, we assume that the honest parties can only process classical information). Thus there is no contradiction.

What this shows is that, even in a hypothetical world where $P = NP$, not all hope is lost for cryptography – provided that one has the ability to generate pseudorandom quantum states.

3 Candidate Construction of PRS generator that doesn't (obviously) imply PRG/OWF

To illustrate how PRS could plausibly not involve any PRG/OWF at all, consider the following candidate construction of a PRS. Consider the generator $G : \{0,1\}^n \to (\mathbb{C}^2)^{\otimes m}$ where for every $k \in \{0,1\}^n$, the generator G interprets the input k as the description of a m-qubit circuit, and then evaluates the circuit k on the all zeroes input $|0 \cdots 0\rangle$ to obtain an m-qubit pure state $|\psi_k\rangle$.

If the depth of circuits k is not too small, then it is plausible that the output states $|\psi_k\rangle$ are pseudorandom. Proving this outright would beyond the reach of current techniques in cryptography/complexity theory (in particular, this would imply that $BQP \neq PSPACE$, which would be a monumental result). But the point is that, in this construction, there is no obvious PRG/OWF lurking inside (especially since PRG/OWFs are defined to be efficiently computable by *classical* algorithms).

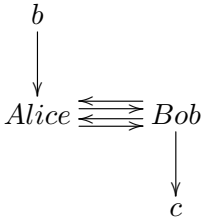
4 Application of PRS: Bit commitment scheme

We now present an example of a cryptographic task, called *bit commitment*, that can be accomplished using pseudorandom states as a primitive. A bit commitment scheme (or just commitment scheme, for short), is the cryptographic analogue of putting a secret in a sealed envelope, and then opening it up later to verify the secret. For example, suppose Alice wants to make a prediction (e.g. stock market will go up tomorrow); today she puts the prediction in the sealed envelope and next day Bob can confirm the prediction. Intuitively, we want that Alice can't change her mind after she puts her prediction into the sealed envelope, but also that Bob can't read the prediction until Alice reveals the contents of the envelope.

In classical cryptography, bit commitment schemes can be constructed from pseudorandom generators (and thus one-way functions), but we want to show how pseudorandom states can also be used to construct them.

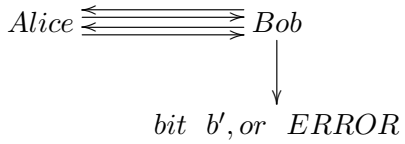
Definition of a commitment scheme. Commitment scheme has two phases, called a *commit phase* and a *reveal phase*. In each phase, Alice (who wants to commit to a bit) communicates interactively with Bob (the verifier of the bit).

- Commit phase: Alice give a bit b (e.g. her prediction of whether the stock market will go up or down) and, after some rounds of communication, Bob outputs the commitment c (i.e. the sealed envelope).



- Reveal phase: After some rounds of communication, Bob either outputs a bit b' (may not

necessarily be the same as Alice's input b) or ERROR.



Correctness Property of a Commitment Scheme. If Alice and Bob are *honest* and follow the commitment scheme according to how they're supposed to, then the scheme should satisfy the *correctness property* in that, after the commit phase and reveal phase, Bob should obtain the bit b that Alice received with high probability. In other words, Bob sees the same bit that was put into the sealed envelope.

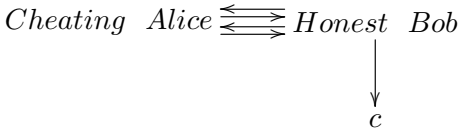
Security Properties of Commitment Schemes. Now imagine that one of the parties are dishonest, meaning that they're trying to cheat the scheme. If Bob is being dishonest, then that means he wants to be able to acquire the bit b before the reveal phase (i.e. open the sealed envelope before the proper time). We say that a commitment scheme satisfies the *hiding property* if, given the commitment c, Bob cannot tell whether c was a commitment to $b = 0$ or to $b = 1$.

If Alice is being dishonest, then that means she wants to be able to trick Bob to output an arbitrary bit b that's chosen *after* the commit phase, but *before* the reveal phase. In the stock market example, this is as if Alice, after learning the stock market crashed, tries to convince Bob that she had predicted it crashed beforehand (even though she might've predicted otherwise). We want the commitment c after the commit phase to be *binding*, meaning that Alice can't change her mind without Bob realizing that she's cheating.

We say that a commitment scheme is *secure* if it satisfies both hiding and binding properties. You might ask, what about the situation when both Alice and Bob are being dishonest? In that case there's no honest user to "protect", so we just ignore that situation.

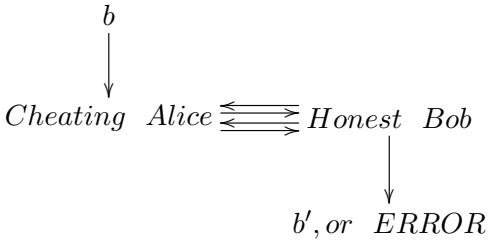
In more detail, the security properties are defined as follows:

- **(Computational) Hiding**: For Honest Alice, Cheating Bob.
  Let c_0 be commitment if Alice commits to $b = 0$. Let c_1 be commitment if Alice commits to $b = 1$. Then the (computational) hiding property is satisfied if for all polynomial-time distinguishers D,
  $$\Pr[D(c_0) = 1] \approx \Pr[D(c_1) = 1] \ .$$
- **(Statistical) Binding**: For Cheating Alice, Honest Bob. In the commit phase, Cheating Alice may not necessarily commit to a bit, but she communicates with Honest Bob anyways and at the end of the commit phase he outputs a supposed "commitment" c.

  In the reveal phase, we think of Cheating Alice as getting a bit b as input and her goal is to

try to convince Honest Bob to believe that c is a valid commitment to b.



Then we say that Let E_b denote the event that Cheating Alice convinces Honest Bob that she committed to bit b. Then the (statistical) binding property is satisfied if no matter what strategy Cheating Alice uses in the commit and reveal phases – even if it takes more than polynomial time – the following inequality is satisfied:

$$\Pr[E_0] + \Pr[E_1] \leq 1 + \text{negl}$$

where negl denotes a negligible quantity (think of it as an arbitrarily small constant).

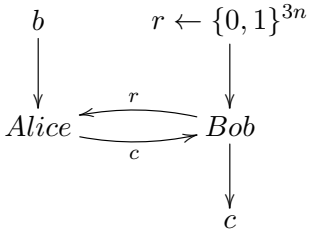
What this inequality is ruling out the scenario that Cheating Alice has a strategy where she gets Honest Bob to create a "fake commitment" c, and then *for the same commitment* has the option of convincing Honest Bob that it was a commitment to both $b = 0$ or $b = 1$ with high probability! That would correspond to $\Pr[E_0] + \Pr[E_1] \approx 2$.

This inequality is saying that, whatever success probability Cheating Alice has to convince Honest Bob that the commitment was to $b = 0$ comes directly at the expense of her success probability of convincing him that it is to $b = 1$.

Commitments using PRGs. Before explaining how to construct a commitment scheme using pseudorandom states, we first describe a famous classical solution to commitments that use pseudorandom generators. This is called the *Naor commitment scheme*. There is no quantum involved in this scheme.

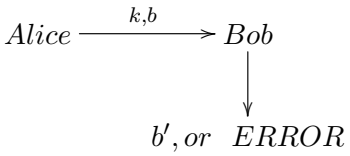
The commitment scheme works as follows. Let $G : \{0,1\}^n \to \{0,1\}^m$ for $m = 3n$ denote a PRG. Then Honest Alice and Honest Bob will follow this protocol.

- Commit Phase



1. Bob samples a uniformly random string $r \leftarrow \{0,1\}^{3n}$ and sends it to Alice.
2. Alice pick a random key $k \leftarrow \{0,1\}^n$ and send commitment $c = G(k) \oplus br$, where br denotes the all zeros string if $b = 0$ and r if $b = 1$.

- Reveal Phase



1. Alice sends k, b to Bob.
2. If $b = 0$, check if $c = G(k)$. If $b = 1$, check if $c = G(k) \oplus r$. If the check passed, then Bob outputs b. Otherwise, output ERROR.

References

[1] Ji, Z., Liu, Y. & Song, F. Pseudorandom quantum states. *Annual International Cryptology Conference*. pp. 126-152 (2018)

[2] Ananth, P., Qian, L. & Yuen, H. Cryptography from Pseudorandom Quantum States. *ArXiv Preprint ArXiv:2112.10020*. (2021)

[3] Morimae, T. and Yamakawa, T., 2021. Quantum commitments and signatures without one-way functions. *Cryptology ePrint Archive.* (2021)

[4] Kretschmer, W. Quantum pseudorandomness and classical complexity. *ArXiv Preprint ArXiv:2103.09320*. (2021)

[5] Impagliazzo, R. and Rudich, S. Limits on the provable consequences of one-way permutations. *In Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (STOC)* pp. 44-61 (1989).