Quantum and Post-Quantum Cryptography

Venkata Koppula[*]

January 2nd, 2023

Contents

Many thanks to Ramprasad Saptarishi for this LaTeX template.

[*] kvenkata@cse.iitd.ac.in

Introduction

Lecture 1: January 3rd, 2023

1 COL759 Recap

Let us start with a quick recap of a few basic principles of provable security from COL759. For any cryptographic primitive, recall the four-step recipe for a provably secure construction:

1. (Syntax and security definition) The first step is to define the syntax. This involves defining the various algorithms involved, as well as the correctness condition. For example, in the case of public key encryption, there is a setup algorithm that outputs a public key and a secret key, an encryption algorithm that uses the public key to encrypt a message (producing a ciphertext), and a decryption algorithm that uses the secret key to decrypt the ciphertext. Correctness requires that if a message m is encrypted using a public key, then decryption of the ciphertext using the corresponding secret key must produce m.

   Once the syntax is defined, we define security for the cryptographic primitive. This is usually done via a security game. For public key encryption, we defined the semantic security game between a challenger and an adversary. The challenger runs the setup algorithm to sample a secret key and a public key. It sends the public key to the adversary. Next, the adversary picks two distinct messages m_0 and m_1 (adversarially), and sends them to the challenger. The challenger encrypts one of them (using the public key) and sends the resulting ciphertext to the adversary. In order to win the game, the adversary must guess whether m_0 was encrypted, or m_1. An encryption scheme is secure if no polynomial time adversary can win this game with noticeable advantage.

2. (Choosing an appropriate cryptographic assumption) Most cryptography is based on cryptographic assumptions. These are computational problems which are believed to be 'hard'. For example, we saw computational problems such as RSA, DDH. The second step, therefore, is to choose an appropriate cryptographic assumption.

3. (Proposing a construction) Once we have chosen a cryptographic assumption, we propose a construction. Depending on the structure present in our cryptographic assumption, the constructions can be very different. For instance, compare the RSA based PKE scheme and the El-Gamal encryption scheme.

4. (Proof of security) The final step ties together the first three steps. We show that if there exists a polynomial time adversary that wins the security game (defined in Step 1) against our construction (given in Step 3), then there exists a polynomial time algorithm that solves the 'hard' computation problem (chosen in Step 2).

This recipe has worked very well so far (at least in theory). Decades of research in algorithms and complexity theory have given us many hard compu-

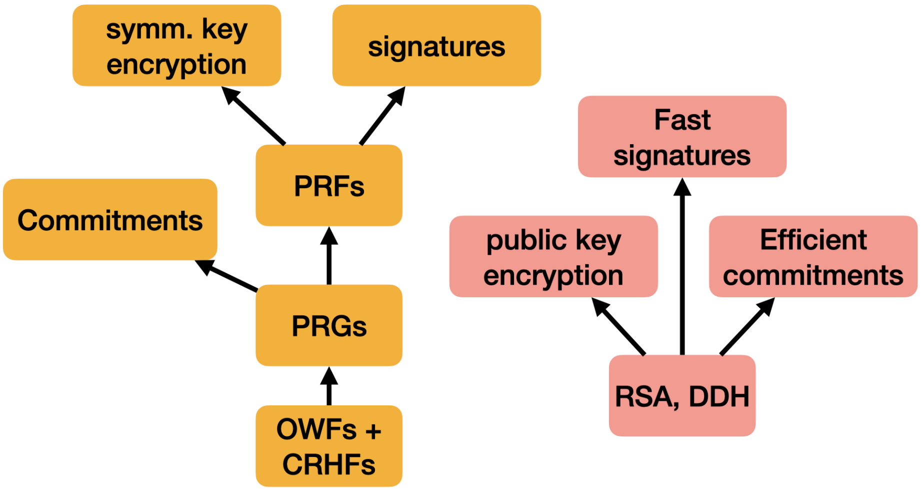

Figure 1: COL759: summarized in one figure. Some cryptographic primitives that can be built from OWFs and CRHFs. In practice, we have very efficient OWFs, PRGs, PRFs and CRHFs. Others such as public key encryption and key agreement rely on computational number-theoretic problems such as RSA or DDH.

tational problems, and some of them are well suited for building cryptographic primitives. Thanks to these hard computational problems, we have good confidence in our security systems.

What happens if someone finds an efficient algorithm for one of these hard problems? The natural idea then is to use a different hard problem, and hope that (a) it is structurally very different so that it doesn't succumb to the same attack (b) it still has enough structure to be useful for cryptography. This is the reason why we have different constructions for the same cryptographic primitive under a diverse set of assumptions.

2 Quantum computers are coming

In the recent years, there has been tremendous progress in the area of quantum computing. A couple of years ago, Google announced a 53 qubit quantum computer. Even though this quantum device was extremely noisy, Google claimed that it could solve certain computational problems within seconds, which would require much more time on a classical computer. While this demonstration is exciting news (and quantum computing has received much media attention due to this), this is bad news for cybersecurity. A lot of our modern protocols rely on number-theoretic problems such as RSA and DDH, and there exist efficient quantum algorithms for solving RSA and DDH. As a result, if we have a crypto primtive whose security is based on RSA/DDH, the security proof is useless in the presence of quantum adversaries.

Since then, other research groups have built quantum processors with 100+ qubits.

Fortunately, we do have computational problems which (a) are potentially quantum-resilient, (b) can be used for building cryptography (see Part I for more details). In Section 4, we discuss how to build quantum resilient one-way functions and quantum-resilient collision resistant hash functions. In Section 5, we discuss how to build quantum-resilient public key encryption schemes. As a result, have we successfully dealt with all quantum adversaries (assuming the computational problems used in Section 4 and 5)? For instance, we saw (in the last two lectures of COL759) that OWFs + CRHFs suffice for building (classical) zero knowledge proofs. Does that mean that quantum-resilient OWFs + quantum-resilient CRHFs imply quantum-resilient zero knowledge proofs? Similarly, it is easy to show that the existence of secure public key encryption implies the existence of secure commitment schemes. Does this mean that quantum-resilient PKE implies quantum-resilient commitment schemes? The next section explores these questions in a bit more detail, and most of this course will revolve around such questions.

3 Challenges in the presence of quantum adversaries

This section will briefly discuss two scenarios where a classical proof/definition is not very useful in the presence of quantum adversaries.

3.1 *Rewinding issue in quantum zero knowledge proofs*

Towards the end of COL759, we saw the notion of zero knowledge proofs, and also saw that public key encryption can be used to build a secure zero knowledge protocol for the 3COL problem, and therefore we get a zero knowledge protocol for all of NP.

We will discuss these protocols in detail next week.

Suppose now we want a zero knowledge protocol in the presence of quantum adversaries. Let us focus on the zero-knowledge property. This involves cheating verifiers. Suppose we want to deal with cheating quantum verifiers. The security definition stays the same, except for a minor modification — we will need to allow the simulator to be a quantum polynomial time algorithm.

Definition Attempt (ZK property wrt quantum poly. time adversaries, informal)**.** *For every quantum polynomial time verifier V^*, there exists a quantum polynomial time simulator S such that the verifier's view in the real protocol is indistinguishable from the simulator's output.*

Given this reasonable looking definition, can we replicate the classical proof for zero-knowledgeness? Unfortunately, no. The proof crucially relied on rewinding the verifier, and in the quantum setting, one needs to be very careful with regard to such arguments (and in particular, many of the classical proofs simply don't have an analogue in the quantum setting).

3.2 *The curious case of commitments*

In the last section, we saw that proofs of security may not translate to the quantum setting. For some cryptographic primitives, we might need new security definitions too. We will illustrate this using cryptographic commitments.

Recall the following toy motivation for commitments: there is a professor who gives very difficult assignments. The students want some 'proof' that the professor can solve the assignment. Commitments can be used in this scenario. The professor, when giving out the assignment, also produces a 'commitment' to the solutions. This commitment must not reveal the committed solutions. Additionally, we also want the commitment to be binding. After the assignment deadline, the professor must produce an 'opening' which proves that he/she indeed committed to the solutions. Informally, we don't want the following scenario: the students submit solutions, and the professor uses one of those (correct) solutions to produce a matching opening. Let us formally define the syntax for commitments, and focus on the binding security game.

Last semester, we defined commitments as a one-round interactive protocol. Here, we are looking at a slightly weaker notion which is easier to work with. There is a honestly chosen public key, and the sender/receiver must use this public key. This allows us a 'non-interactive' commitment scheme.

Non-interactive commitments with setup : A non-interactive commitment scheme with setup consists of the following two algorithms:

- $\mathsf{Setup}(1^n)$: The setup algorithm takes as input the security parameter, and outputs a public key pk.
- $\mathsf{Commit}(\mathsf{pk}, m; r)$: The commitment algorithm takes as input a public key pk, message m and randomness r. It outputs a commitment com. The randomness r is used as the opening.

Last semester, we defined the following (strong) security game for binding:

Binding-Game

- Challenger chooses a public key pk using Setup and sends it to the adversary.
- Adversary produces a commitment com, and two messages m_0, m_1 and randomness r_0, r_1. The adversary wins if $\mathsf{Commit}(\mathsf{pk}, m_0; r_0) = \mathsf{Commit}(\mathsf{pk}, m_1; r_1) = \mathsf{com}$.

Figure 2: The Binding Security Game

Note, for our toy scenario, we only require a weaker security requirement for binding. I'll call this 'prof-binding':

In our informal lingo, weaker security requirement implies the adversary's job is harder, and therefore our 'expectations' from the commitment scheme are 'weaker'.

Prof-binding is not a standard notion; please don't use this outside our classroom :) We will give this a proper name when we discuss this in more detail.

Prof-Binding-Game

- Challenger chooses a public key pk using Setup and sends it to the adversary.
- Adversary produces a commitment com.
- Challenger picks a random message m and sends it to the adversary.
- Adversary produces randomness r and wins if $\mathsf{Commit}(\mathsf{pk}, m; r) = \mathsf{com}$.

Figure 3: The Prof-Binding Security Game

Prof-binding a weaker security requirement than (standard) binding, because if a commitment scheme satisfies binding, then it also satisfies prof-binding.

Claim 3.1. *Let $\mathcal{C} = (\mathsf{Setup}, \mathsf{Commit})$ be a commitment scheme. If there exists a p.p.t. adversary that wins the* Prof-Binding-Game *w.r.t. $\mathcal{C}$, then there exists a p.p.t. adversary that wins the* Binding-Game *w.r.t. $\mathcal{C}$.*

Sketch of Proof. Suppose there exists an adversary that wins the Prof-Binding-Game w.r.t. $\mathcal{C}$. The reduction algorithm works as follows:

- The reduction algorithm receives public key pk from the challenger, which it forwards to the adversary.
- Next, the adversary sends a commitment com. The reduction algorithm picks a uniformly random message m and sends to the adversary. The adversary sends opening r.
- The reduction then 'rewinds' the adversary to the point immediately after it sends com. The reduction chooses a new random message m' and sends it to this rewinded adversary. It receives r' as the new opening. Note that if the adversary successfully wins the Prof-Binding-Game, then $\mathsf{Commit}(\mathsf{pk}, m; r) = \mathsf{Commit}(\mathsf{pk}, m'; r') = \mathsf{com}$.
- The reduction algorithm sends (m, r) and (m', r') to the challenger, and wins the Binding-Game.

□

Note: It is important to rewind the adversary in this proof, and not restart the adversary's execution. This is because we want two (message, randomness) pairs *for the same commitment* com.

This claim, together with the claim you proved in Assignment 1 of COL759, give us the following theorem:

Theorem 3.2. *Assuming the existence of secure PRGs, there exists a commitment scheme that satisfies binding security. Using Claim 3.1, there also exists a commitment scheme that satisfies prof-binding security (assuming the existence of secure PRGs).*

Let us now focus our attention on the quantum analogues of these security definitions. The definition is quite natural — replace the 'p.p.t. adversary' with a 'quantum polynomial time adversary'. Your COL759 Assignment 1 proof can be adapted to get a commitment scheme that satisfies binding security w.r.t. quantum adversaries, assuming quantum-secure PRGs. However, the existence of quantum-secure PRGs **does not** immediately imply the existence of commitment schemes that satisfies prof-binding security w.r.t. quantum adversaries. Again, the issue is that we cannot rewind a quantum adversary. This brings us to the following questions:

> *What is the 'correct' definition for binding security in the presence of quantum adversaries? And if we care about prof-binding, then how do we construct such commitment schemes?*

3.3 *Going beyond classical messages*

So far, we have considered classical crypto primitives in the presence of quantum adveraries. What happens if the messages themselves are *qubits* instead of classical bit strings? Can we commit to a quantum state, or encrypt a quantum state? Can we commit to a quantum state using only classical communication? These are some of the questions that we will consider in the final segment of our course.

Part I: Lattice-based Cryptography

Notations We will use the following notations for this chapter (and most other chapters, unless specified otherwise).

- $\mathbb{Z}_q = \{-q/2, \ldots, -1, 0, 1, \ldots, q/2 - 1\}$. The $(\bmod\ q)$ operation is a mapping from $\mathbb{Z}$ to $\mathbb{Z}_q$.
- For a finite set $\mathcal{S}$, $x \leftarrow \mathcal{S}$ denotes a uniformly random element drawn from $\mathcal{S}$. Similarly, for a distribution $\mathcal{D}$ over $\mathcal{S}$, $x \leftarrow \mathcal{D}$ denotes an element drawn from distribution $\mathcal{D}$.
- For any two distributions $\mathcal{D}_1, \mathcal{D}_2$ over a finite set $\mathcal{X}$, the statistical distance between $\mathcal{D}_1$ and $\mathcal{D}_2$, denoted by $\mathsf{SD}(\mathcal{D}_1, \mathcal{D}_2)$ is defined as follows:

  $$\mathsf{SD}(\mathcal{D}_1, \mathcal{D}_2) = \frac{1}{2}\left(\sum_{a \in \mathcal{X}} \left| \Pr_{x \leftarrow \mathcal{D}_1}[x = a] - \Pr_{x \leftarrow \mathcal{D}_2}[x = a] \right|\right)$$

  We say that two distributions are statistically indistinguishable, denoted by $\mathcal{D}_1 \approx_s \mathcal{D}_2$ if $\mathsf{SD}(\mathcal{D}_1, \mathcal{D}_2)$ is negligible in $n = \log |\mathcal{X}|$.

  If two distributions are statistically indistinguishable, then no algorithm can distinguish between these two distributions with non-negligible advantage.
- Let $\{\mathcal{X}_n\}_n$ be a family of finite sets, where $|\mathcal{X}_n| \leq 2^{\mathrm{poly}(n)}$ for some fixed polynomial $\mathrm{poly}(\cdot)$. For any two efficiently sampleable distributions $\mathcal{D}_1, \mathcal{D}_2$ over $\mathcal{X}_n$, the shorthand notation $\mathcal{D}_1 \approx_c \mathcal{D}_2$ indicates that the two distributions are computationally indistinguishable. That is, for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

  $$\Pr[\mathcal{A}(x) = 1 : x \leftarrow \mathcal{D}_1] - \Pr[\mathcal{A}(x) = 1 : x \leftarrow \mathcal{D}_2] \leq \mu(n).$$

4 The Short-Integer-Solutions Problem

The first (potentially) quantum-resilient assumption of this course is the Short Integer Solutions problem.

Computational Problem 1 (Short Integer Solution Problem ($\mathsf{SIS}_{n,m,q}$))**.** *Given a uniformly random matrix* $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$*, output a nonzero vector* $\mathbf{x} \in \{-1, 0, 1\}^m$ *such that* $\mathbf{A} \cdot \mathbf{x} \pmod{q} = \mathbf{0}$.

The 'interesting' regime for this problem is when $n^{1.5} \leq q \leq \exp(n)$ and $m \geq \Omega(n \log q)$. If we did not have the constraint that $\mathbf{x} \in \{-1, 0, 1\}$, then this problem can be solved easily using Gaussian elimination. However, once we add this restriction for $\mathbf{x}$, this problem is no longer easy. If m is large enough, then such a 'short integer solution' is guaranteed to exist (see exercise below).

Exercise 4.1. *Show that if $3^m > q^n$, then for every matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, there exists a vector $\mathbf{x} \in \{-1,0,1\}^m$ such that $\mathbf{A}\cdot\mathbf{x} \pmod q = \mathbf{0}$.*

However, finding such a short integer solution is believed to be hard (even given a quantum computer). Ajtai [Ajt96] showed that, in the interesting regime, if there exists an efficient algorithm for the SIS problem, then there exists an efficient algorithm for certain lattice problems in the *worst-case*.

4.1 *One-Way Functions and Collision Resistance from SIS*

Recall, a collision resistant hash function family is a set of keyed functions such that, given a random function from this family, it is hard to find two inputs that map to the same output. Let q be a power of 2, and $m > 2n\log q$. Consider the following function family $\mathcal{H} = \left\{H_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_q^n\right\}_{\mathbf{A}\in\mathbb{Z}_q^{n\times m}}$ where

$$H_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\cdot\mathbf{x} \pmod q$$

This is a compressing function since $m > 2n\log q$, and if there exist distinct bit-vectors $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{A}\cdot\mathbf{x} = \mathbf{A}\cdot\mathbf{y}$, then $\mathbf{A}\cdot(\mathbf{x}-\mathbf{y}) = \mathbf{0}$, and $\mathbf{x}-\mathbf{y} \in \{-1,0,1\}^m$.[1]

Since this function family is sufficiently compressing, collision resistance implies one-wayness.

Exercise 4.2. *Show that if $\mathcal{H} = \left\{H_k : \{0,1\}^{2n} \to \{0,1\}^n\right\}_{k\in\mathcal{K}}$ is a collision-resistant hash function family, then $\mathcal{H}$ is also a one-way function family.*

However, this does not hold true if $\mathcal{H}$ is only mildly-compressing. That is, suppose $\mathcal{H} = \left\{H_k : \{0,1\}^{n+1} \to \{0,1\}^n\right\}_{k\in\mathcal{K}}$ is a collision-resistant hash function family. Construct a new hash function family $\mathcal{H}'$ that is also collision-resistant, but $\mathcal{H}'$ is not one-way.

4.2 *Commitment Scheme from SIS*

In this section, we will present a non-interactive commitment scheme with setup, based on SIS. For this construction, we will consider a variant of SIS, called 'inhomogeneous SIS' (iSIS).

Computational Problem 2 (Inhomogeneous Short Integer Solution Problem ($\mathsf{iSIS}_{n,m,q}$))**.** *Given a uniformly random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and a uniformly random vector $\mathbf{b} \leftarrow \mathbb{Z}_q^n$, output a nonzero vector $\mathbf{x} \in \{-1,0,1\}^m$ such that $\mathbf{A}\cdot\mathbf{x} \pmod q = \mathbf{b}$.*

It is easy to show that this variant is as hard as SIS.

[1] All these equalities are modulo q, I will ignore $\pmod q$ when it is clear from the context.

Exercise 4.3. *Show that there exists a p.p.t. adversary that solves* $\mathsf{iSIS}_{n,m,q}$ *with non-negligible probability, if and only if there exists a p.p.t. adversary that solves* $\mathsf{SIS}_{n,m,q}$ *with non-negligible probability.*

The iSIS gives a direct proof of one-wayness of $\{H_{\mathbf{A}} \equiv \mathbf{A} \cdot \mathbf{x}\}_{\mathbf{A}}$. This proof goes via the *leftover hash lemma*, a useful lemma that we'll encounter multiple times in this course. Essentially, this lemma says that if m is sufficiently larger than $n \log q$, then given a random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, the vector $\mathbf{A} \cdot \mathbf{x}$, where $\mathbf{x}$ is drawn from a distribution with enough randomness, looks like a uniformly random vector in $\mathbb{Z}_q^n$. We will not prove this lemma here (you can refer to [AB09] for a proof of the lemma).

Fact 4.4. *Let* $S \subset \mathbb{Z}_q$, $|S| = B$. *If* $m \geq n \log_B q + n$, *then the statistical distance between the following distributions is a negligible funtion of* n*:*

$$\mathcal{D}_1 = \left\{ (\mathbf{A}, \mathbf{A}\cdot\mathbf{x}) : \begin{array}{c} \mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \\ \mathbf{x} \leftarrow \{0,1\}^m \end{array} \right\} \qquad \mathcal{D}_2 = \left\{ (\mathbf{A}, \mathbf{u}) : \begin{array}{c} \mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \\ \mathbf{u} \leftarrow \mathbb{Z}_q^n \end{array} \right\}$$

This is a simplified version of the leftover hash lemma. The 'full version' states that for any distribution $\mathcal{D}$ *over* $\mathbb{Z}_q^m$ *with sufficient entropy, if* $\mathbf{x} \leftarrow \mathcal{D}$, *then* $(\mathbf{A}, \mathbf{A}\cdot\mathbf{x}) \approx_s (\mathbf{A}, \mathbf{U})$.

Exercise 4.5. *Let* $m = n$. *Compute a lower bound on the statistical distance between the following distributions:*

$$\mathcal{D}_1 = \left\{ (\mathbf{A}, \mathbf{A}\cdot\mathbf{x}) : \begin{array}{c} \mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \\ \mathbf{x} \leftarrow \{0,1\}^m \end{array} \right\} \mathcal{D}_2 = \left\{ (\mathbf{A}, \mathbf{u}) : \begin{array}{c} \mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \\ \mathbf{u} \leftarrow \mathbb{Z}_q^n \end{array} \right\}$$

Let us put the leftover hash lemma to good use. Recall, last semester we had discussed that OWFs can be used to build PRGs, and in one of your assignments, you had constructed a commitment scheme using PRGs. Therefore, following this approach, we have a commitment scheme whose security is based on SIS. There is a simpler direct construction based on SIS, which is described below.

... to be continued in next lecture.
Please think about this construction before next lecture.

Lecture 2:
January 6th, 2023

Construction 4.6 (Commitment Scheme from SIS)**.** *The message space for our commitment scheme is* $\{0,1\}^m$, *and the sender's commitment algorithm uses* m *bits of randomness.*

- $\mathsf{Setup}(1^n)$:
- $\mathsf{Commit}(\mathsf{pk}, \mathsf{msg})$:

$\Diamond$

5 The Learning-with-Errors Problem

Part II: Interactive Proofs

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009.

[Ajt96] M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 99–108, New York, NY, USA, 1996. Association for Computing Machinery.

Appendix: Cryptographic Primitives

Definition 1 (Non-interactive commitment scheme with setup)**.** *A non-interactive commitment scheme with setup consists of two algorithms:* Setup *and* Commit *with the following syntax:*

- $\mathsf{Setup}(1^n)$*: takes as input the security parameter, and outputs a public key* pk*.*
- $\mathsf{Commit}(\mathsf{pk}, \mathsf{msg}; r)$ *: takes as input the public key* pk*, the message* msg *to be committed, randomness r, and outputs a commitment* com*.*

A non-interactive commitment scheme with setup must satisfy the following two security properties:

- ***Binding property****: A commitment scheme satisfies the binding property if for any prob. poly. time adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,* $\Pr[\mathcal{A}$ *wins the binding security game*$] \leq \mu(n)$ *where the binding security game is defined below:*

Binding-Game

- Challenger chooses $\mathsf{pk} \leftarrow \mathsf{Setup}(1^n)$ and sends pk to $\mathcal{A}$.
- Adversary sends a commitment com, together with two (message, opening) pairs (msg_0, r_0) and (msg_1, r_1). The adversary wins if $\mathsf{Commit}(\mathsf{pk}, \mathsf{msg}_0; r_0) = \mathsf{Commit}(\mathsf{pk}, \mathsf{msg}_1; r_1) = \mathsf{com}$.

Figure 4: The Binding Security Game

- ***Hiding property****: A commitment scheme satisfies the hiding property if for any prob. poly. time adversary $\mathcal{A}$, there exists a neglibile function $\mu(\cdot)$ such that for all n,* $\Pr[\mathcal{A}$ *wins the hiding security game*$] \leq 1/2 + \mu(n)$ *where the binding security game is defined below:*

Hiding-Game

- Challenger chooses $\mathsf{pk} \leftarrow \mathsf{Setup}(1^n)$ and sends it to the adversary.
- The adversary sends two messages $\mathsf{msg}_0, \mathsf{msg}_1$.
- Challenger chooses $b \leftarrow \{0,1\}$, computes $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{pk}, \mathsf{msg}_b)$ and sends com to $\mathcal{A}$.
- Adversary sends guess b' and wins if $b = b'$.

Figure 5: The Hiding Security Game

$\Diamond$