# Week 6

Nadav Kohen

July 30, 2025

Last week we looked at some advanced topics: Sigma protocols and the OMDL and AMODL assumptions. In our final week of prework, we will review and reinforce our understanding of the Schnorr security proof, as well as practice writing security definitions for new protocols. We will also briefly introduce Schnorr multi-signatures and Schnorr threshold signatures.

Before we begin, let us take a moment to take another pass at the range proofs that were discussed in our last call together.

**Exercise 1.** *Recall that for a group $\mathbb{G}$ of worder $p$ with generators $g$ and $h$, a Pedersen commitment to $x \in \mathbb{Z}_p$ with randomness $y \in \mathbb{Z}_p$ is of the form $C = g^x h^y$.*

*(a) Construct a special HVZK Sigma protocol with special soundness for the statement that a commitment value, $C$, is a Pedersen commitment for the value $x = 1$ (without revealing $y$, of course). Create this protocol by starting with the Schnorr ID protocol for $h^y$ and modifying the verification step at the end.*

*(b) Next, use the OR construction for Sigma protocols from last week to construct a special HVZK Sigma protocol with special soundness for the statement that a commitment value, $C$, is a Pedersen commitment for either the value $x = 0$, or the value $x = 1$ (without revealing which, of course).*

(c) *Instead of computing $C = Commit(x;y)$ by generating a random value of $y$ and returning $g^x h^y$, we will instead generate $n$ random values, $y0, \ldots, y_{n-1}$ and let $y$ be equal to $y_0 + 2y_1 + \cdots + 2^{n-1}y_{n-1} \bmod p$. Then, if $x$ is equal to $x_0 + 2x_1 + \cdots + 2^{n-1}x_{n-1}$ (where the $x_i$ are $x$'s bits in binary) we get that $C = \prod_i Commit(x_i; y_i)(2^i)$ and each of the $x_i$ is either equal to 0 or 1! Because our commitments are perfectly hiding, it is safe to share the values $Commit(x_i; y_i)$.*

*Using your solution to the previous part, write down an explicit Sigma protocol for the statement that all of the commitments, $C_i = Commit(x_i; y_i)$, are committed to a bit (0 or 1). Alongside the verification that $C = \prod_i C_i^{2^i}$, this constitutes a range proof!*

Moving on, as we have now seen more than a couple times, reducing the security of protocols such as Schnorr to underlying computational assumptions usually (outside of the AGM) relies on the use of "rewinding"/"forking" arguments, where an adversary is run multiple times with related inputs and a fixed source of randomness. Because of the prevalence of these kinds of arguments, security proofs can oftentimes be simplified by refactoring so that there is a single independent forking lemma beforehand, which is invoked within a security proof.

I actually chose our first Schnorr security proof to be out of the Katz and Lindell book, in part, because they do not do this. But now that we have the proper motivation,

**Reading 1.** *Read about a simple and general forking lemma in:*
*Mihir Bellare and Gregory Neven - New Multi-Signature Schemes and a General Forking Lemma - Section 3. DO NOT (YET) READ SECTION 4!*

I recommend that, on a first reading, you should not worry too much about the details of the proof of the lemma, and instead focus primarily on understanding what the lemma is claiming.

**Exercise 2.** *Using the lemma from Reading 1, write up a proof in your own words that*

*Schnorr digital signatures are secure in the Random Oracle Model if the Discrete Log problem is hard. You may refer back to the proofs in Katz and Lindell, but I would like a direct proof that does not explicitly invoke the Fiat-Shamir transform. I encourage you to use this opportunity to be very pedantic and to ensure that you are comfortable with all of the details!*

Once you have completed, or at least nearly completed, this exercise,

**Reading 2.** *Read the general forking lemma's author's proof of the security of the Schnorr digital signature that they used to demonstrate the use of their lemma in:*
*Mihir Bellare and Gregory Neven - New Multi-Signature Schemes and a General Forking Lemma - Section 4.*

The Bellare and Neven paper of Readings 1 and 2 introduced a Schnorr-based multi-signature that was a precursor to MuSig (you can go back and read more of this paper if you wish, but it is probably not the best use of our time together).

**Exercise 3.** *In the following reading, pause on page 9 when you reach the SECURITY section, read the first paragraph of that section, and then attempt to define multi-signature scheme security against concurrent attacks using an attack game to capture the notion described in that first paragraph.*

*Compare your solution to the definition given in the reading.*

**Reading 3.** *Read about MuSig in:*
*Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille - Simple Schnorr Multi-Signatures with Applications to Bitcoin.*

As before, on a first reading, I recommend that you don't get too caught up in proof details, but rather return to the proofs once you have gotten a full high-level picture of the protocol and approach.