Week 5

Nadav Kohen

July 23, 2025

Last week we studied two proofs of the security for the Schnorr digital signature assuming that DL is hard (one in the AGM+ROM, the other just in the ROM). We then introduced the notion of a Sigma protocol that generalizes and abstracts much of the ideas behind the Schnorr ID protocol and its Fiat-Shamir transform (the Schnorr digital signature), which we will be expanding on this week. Let's begin with some examples of Sigma protocols:

Reading 1. *Read about Okamoto's protocol, the Chaum-Pedersen protocol, and a Sigma protocols for arbitrary linear relations (optionally, also read about a Sigma protocol for the pre-image of a homomorphism) in:*
Dan Boneh and Victor Shoup - A Graduate Course in Applied Cryptography - Section 19.5.

Before moving on, we will quickly apply some of our new knowledge to a Sigma protocol relevant to Bitcoin,

Exercise 1. *Proofs of Discrete Log Equality, also referred to as PoDLEs (pronounced poodles) or DLEQ proofs, are a Sigma protocol where the verifier has is given four group elements, g, h, A, and B, and the prover shows that they know a single value a such that $A = g^a$ and $B = h^a$. In other words, they prove that the elements A and B have the same discrete log subject to different bases/generators (g and h). Propose a Sigma protocol for this relation.*

Time to replace our verifiers with hash functions modeled as random oracles.

Reading 2. *Using the Schnorr signature as a mental model, read about the Fiat-Shamir transform for Sigma protocols in:*
Dan Boneh and Victor Shoup - A Graduate Course in Applied Cryptography - Section 19.6.

Exercise 2. *Use the Fiat-Shamir transform applied to your solution for Exercise 1 to create a non-interactive DLEQ proof algorithm. Compare your proposal to BIP 374.*

Exercise 3. *Suppose you have a Sigma protocol, (P, V), that is special HVZK. Show that the signature scheme resulting from the variant of the Fiat-Shamir transform where we set the challenge to be $c = H(m)$ instead of $c = H(R, m)$ is insecure.*
(Hint: You have access to a HVZK simulator).

Reading 3. *Read about combining Sigma protocols in:*
Dan Boneh and Victor Shoup - A Graduate Course in Applied Cryptography - Section 19.7.

Exercise 4. *Generalize the AND-proof and OR-proof constructions in the previous reading from two Sigma protocols to n Sigma protocols (where $n \geq 2$ is a constant number). State the relations for your new Sigma protocols, and argue that they provide special soundness and are special HVZK under appropriate assumptions. The computational and communication complexity (but not number of rounds) of your protocols should scale linearly in n.*

You have now read (or read the equivalent of) the vast majority of Chapter 19 of a graduate cryptography textbook, amazing!

To finish this week off, I will now introduce some variants of the Discrete Log problem (DL), namely the One More Discrete Log problem (OMDL) and the Algebraic One More Discrete Log problem (AOMDL).

As Chelsea Komlo writes in her note, “Notes on Proving the Security of Single-Party Schnorr” (which I will link on Discourse), “The OMDL problem is useful for proving variants of Schnorr signature schemes when it becomes difficult for the prover to correctly program the random oracle. Such situations can occur when the adversary has more influence over the inputs that are provided to the random oracle. For example, in threshold signature schemes or blind signatures, the adversary is allowed to participate in the signing protocol, and so has some amount of influence over the value of the challenge. In this setting, proving security becomes harder because the simulator does not know the value of the challenge c when it selects its commitment R. While OMDL allows for a useful proving mechanism in this setting, OMDL is a stronger assumption than plain dlog, which has tradeoffs when considering how realistic this assumption is in a practical setting."

In that same note, a summary of how OMDL is used to prove the Schnorr ID scheme from last week is secure under concurrent attacks is given, but I would like us to actually go through the source paper for our next reading:

Reading 4. *Read about how we can use OMDL to prove Schnorr ID secure against impersonation under concurrent attacks (IMP-CA) in the following paper, skipping the sections about GQ: Mihir Bellare and Adriana Palacio - GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks*

Note that OMDL is not required for showing Schnorr *signatures* are secure against concurrent attacks because the Fiat-Shamir transform eliminates the interactive challenge that a concurrent attacker might take advantage of during the Schnorr *identification protocol*, which is the focus of the above reading.

A related assumption proposed by Jonas Nick, Tim Ruffing, and Yannick Seurin in the MuSig2 paper is the AOMDL assumption. As they put it, "A cryptographic assumption is algorithmically falsifiable if it can be decided in p.p.t. whether a given algorithm breaks it. While this is true for most standard assumptions such as the RSA assumption or the DL assumption, it is notably not true for the OMDL assumption, where the OMDL challenger needs to provide the adversary with a DL oracle that cannot be implemented in p.p.t. (unless the DL problem is easy, but then the OMDL assumption does not hold anyway).

While we believe that the OMDL has withstood the test of time, it is still desirable to avoid nonfalsifiable assumptions whenever possible. We observe that the DL oracle can be in fact implemented in p.p.t. when the solving algorithm is required to be algebraic. In the context of OMDL, this translates to the requirement that whenever the adversary queries the discrete logarithm of a group element via the DL oracle, it outputs a representation of this group element in the basis formed by the generator and all DL challenges it has received thus far (which together constitute all group elements it has received thus far). As a result we obtain a falsifiable variant of the OMDL assumption that we call the algebraic OMDL (AOMDL) assumption. Since every algebraic algorithm is also a normal algorithm, the AOMDL assumption is immediately implied by the well-established OMDL assumption.

Since our reductions in both the ROM and in the AGM+ROM are algebraic in this sense, we can rely on the falsifiable AOMDL assumption. We would like to stress that being algebraic here refers to a property of the reduction, which acts as the algorithm solving (A)OMDL, and our reductions are algebraic independent of whether the unforgeability adversary, to which the reduction has access internally, is algebraic. As such, the use of the AOMDL assumption is independent and orthogonal of our use of the AGM as described in the previous subsection. In particular we can rely on the AOMDL assumption even in our ROM-only proof.

We believe that the AOMDL problem is helpful beyond the scope of this paper, as it turns out that essentially all security proofs in the literature use the OMDL problem in an algebraic and thus falsifiable fashion [e.g., BP02; NKD+03; BS07; FPS20]. We do not claim that our observation about algebraic algorithms is a deep insight—in fact implementing the DL oracle is straight-forward given an algebraic solving algorithm—we simply believe it is useful for the evaluation of security results."

Exercise 5. *Justify the statement "that the DL oracle can be in fact implemented in p.p.t. when the solving algorithm is required to be algebraic." Specifically, describe how the DL oracle should be implemented.*

Exercise 6. *The first paper cited, BP02, in the claim "as it turns out that essentially all security proofs in the literature use the OMDL problem in an algebraic and thus falsifiable fashion" is actually the paper from the previous reading! Rewrite the reduction in Figure 5 to be an AOMDL adversary instead of an OMDL adversary by providing the necessary additional coefficient data to all DL oracle invocations.*

Next week we will be finishing up our prework by reviewing and by studying some recent papers relevant to Bitcoin!