ASCrypto 2025: Introduction to Proving Systems

September 29 of 2025

Author Emanuel Nicolás Herrador
Speaker Arantxa Zapico

1 Interactive proofs

The idea is to prove something to another entity efficiently without proving it at all (complete). It is made interactive. It must satisfy:

- Completeness: If something is indeed true and both (prover and verifier), follow the procedure, verifier accept
- Soundness: If something is false, then verifier rejects with overwhelming probability
- Zero-Knowledge: The verifier does not learn anything but the truth of something

More formally, we define something as follows. Let be $R = \{(x, y) : \dots\}$ a PT relationship, and “something is true” if $x \in \mathcal{L}_R$ where L is the language of R.

Prover sends messages to verifier and verifier sends challenges to our prover to prove that it’s true something. The challenge are random (as possible) to avoid deshonest provers.

There’re some points to take in account like efficiency. Things like public parameter size (pp), proof size ($|m_1| + \dots + |m_k|$), prover time and verifier time.

Let’s see some properties that interactive proofs can have. For that, we consider $\mathcal{P}$ as honest prover and $\mathcal{P}^*$ as malicious prover.

Succinctness This is the first point about a SNARK. We need a proving system to be succinct in communication and in verification. I.e., $\sum |m_i| << |w|$ and $\text{time}(\mathcal{V}) << \text{time}_R(x, w)$ respectively.

(Perfect) Completeness If $x \in \mathcal{L}_R$ and $\mathcal{P}$ follow the procedure, $\mathcal{V}$ accepts. The probability must be 1 if we want perfect completeness. Formally, $Pr[\langle \mathcal{P}(pp, (x, w), \mathcal{V}(pp, x))] = 1$.

(Computational) Soundness If $x \notin \mathcal{L}_R$ then $\mathcal{V}$ rejects with overwhelming probability. I.e., if $\nexists w : (xm, w) \in R$ then $\mathcal{V}$ rejects it with overwhelming probability. Formally, $Pr[\langle \mathcal{P}^*(pp, x), \mathcal{V}(pp, x)\rangle \leq \text{negl}(\lambda)$

Knowledge-soundness There exists a PT algorithm $\mathcal{E}$, the extractor, such that for every malicious prover $\mathcal{P}^*$, then $Pr[(x, w) \in R : w \leftarrow \mathcal{E}^{\mathcal{P}^*}(x)] - Pr[\langle \mathcal{P}^*(x), \mathcal{V}(x)\rangle = 1] \leq \text{negl}(\lambda)$

An argument that satisfy Knowledge-soundness is an argument of knowledge. I.e., Knowledge-soundness is more powerful that soundness.

1.1 SNARK(G)s

It’s one way to construct it.

Tool 1: Interactive Oracle The prover sends message to an oracle and verifier ask questions to this oracle. Then, verifier doesn’t learns anything about m_i because its questions are asked by the oracle. The oracle works as an intermediate between them and we assume it as powerful and trusted.

Tool 2: Functional Commitment Scheme We can’t trust to an oracle. Therefore, we’ve to use a commitment scheme to replace this oracle when $\mathcal{P}$ sends commitments and also make computations of f; $\mathcal{P}$ also sends the proof. The scheme is as follows: $\mathcal{P}$ sends a commitment of m_1, then $\mathcal{V}$ ask for $f(m_1, \alpha_1)$ and $\mathcal{P}$ reply with $y \leftarrow f(m_1, \alpha_1)$ and $\pi \leftarrow \text{Open}(f, y)$ as the proof. Finally, $\mathcal{V}$ can do the verification with $\text{Verify}(y, \text{com}_1, \alpha_1, \pi)$.

Notice that the succinctness of this scheme is based in the commitments (as for example, proof size is the size of commits and $pp \leftarrow CS \cdot \mathcal{K}$).

As Oracle is an idealized model, here we don’t have cryptographic assumptions. However, we’ve it in the commitment scheme.

From interactive to non-interactive proofs With that we construct an interactive succinct argument and we want to construct a non-interactive succinct argument. For that we'll use a hash function $H : \{0,1\}^* \to \{0,1\}^{256}$. And we want that for this function the following problems are harder to solve (computationally):

- Collision resistant: Find $x, y : H(x) = H(y)$
- Pre-image resistant: Given z, find $x : H(x) = z$
- Second pre-image resistant : Given x, find $y : H(x) = H(y)$

Here, all the messages (commitments) and challenge answers are sent by $\mathcal{P}$ once. Therefore, we want a way for $\mathcal{P}$ to generate random challenges to solve and prove that they are random.

Here we'll use **Fiat-Shamir Heuristic**. The challenge is created with the hash of all the public information gotten before. I.e., for example, $a_i = H(x, \text{com}_1, \ldots, \text{com}_i)$. Let's say that π is all the proof sent by $\mathcal{P}$. Then, it's secure under the random oracle model.

We've for knowledge soundness:

$$\begin{aligned}
&pp \leftarrow \mathcal{K}\\
&(x, \pi) \leftarrow \mathcal{P}^*(pp)\\
&w \leftarrow \mathcal{E}(pp, x, \pi)\\
&Pr\left[(x, w) \notin R \wedge \mathcal{V}(pp, x, \pi) = 1\right] \leq \text{negl}(\lambda)
\end{aligned}$$

1.2 Short summary

We saw:

- SNARK: Succinct Non-interactive Argument of Knowledge
- SNARG: Succinct Non-interactive Argument
- Efficiency: prover/verifier time, proof/pp size
- Security: Setup (trusted/transparent), model (ROM) and assumptions (discrete log)

And most of it depends on the commitment scheme.