Commitment schemes

Distributed Lab

August 20, 2024

Plan

Commitments Overview

Commitment Definition

Definition

A cryptographic commitment scheme allows one party to commit to a chosen statement without revealing the statement itself. The commitment can be revealed in full or in part at a later time, ensuring the integrity and secrecy of the original statement until the moment of disclosure.

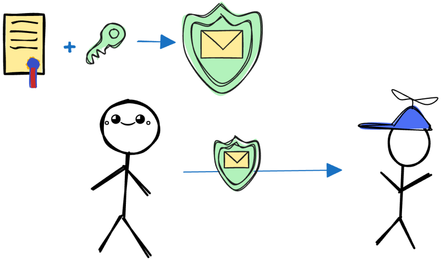

Figure: Overview of a commitment scheme

Commitment Definition

Definition

Commitment Scheme $\Pi_{\text{commitment}}$ is a tuple of three algorithms: $\Pi_{\text{commitment}} = (\text{Setup}, \text{Commit}, \text{Verify})$.

1. Setup (1^λ): returns public parameter *pp* for both comitter and verifier;
2. Commit (pp, m, r): returns a commitment c to the message m using public parameters *pp* and, optionally, a secret opening hit r;
3. Open (pp, c, m, r): verifies the opening of the commitment to the message m with an opening hit r.

Commitment Scheme Properties

Definition

1. *Hiding:* verifier should not learn any additional information about the message given only the commitment $\mathcal{C}$.
   1. *Perfect hiding*: adversary with any computation capability tries even forever cannot understand what you have hidden.
   2. *Computationally hiding*: we assume that the adversary have limited computational resources and cannot try forever to recover hidden value.
2. *Binding:* prover could not find another message m_1 and open the commitment $\mathcal{C}$ without revealing the commited message m.
   1. *Perfect binding*: adversary with any computation capability tries even forever cannot find another m_1 that would result to the same $\mathcal{C}$.
   2. *Computationally binding*: we assume that the adversary have limited computational resources and cannot try forever.

Note

Perfect hiding and perfect binding cannot be achived at the same time

Hash-based Commitments

Hash-based commitments

As the name implies, we are using a cryptographic hash function H in such scheme.

Definition

1. Prover selects a message m from a message space M which he wants to commit to: $m \leftarrow \mathbb{M}$
2. Prover samples random value r from a challange space C (usually called blinding factor) from $\mathbb{Z}$: $r \xleftarrow{R} \mathbb{C}$
3. Both values will be concatenated and hashed with the hash function H to produce the commitment: $\mathcal{C} = H(m \parallel r)$

Vector Commitments

Merkle Tree commitments

A naive approach for a vector commitment would be hash the whole vector. More sophisticated scheme uses divide-and-conquer approach by building a binary tree out of vector elements.

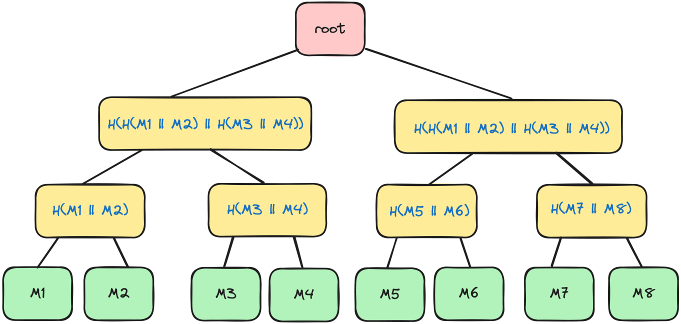

Figure: Merkle Tree structure

Merkle Tree Proof (MTP)

To prove the inclusion of element into the tree, a corresponding Merkle Branch is used. It allows to perform selective disclosure of the elements without revealing all of them at once.

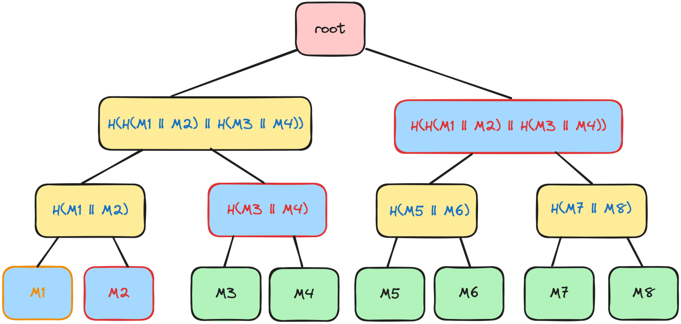

Figure: Merkle Tree inclusion proof branch

Thanks for your attention!