BIT COMMITMENT AND ZERO-KNOWLEDGE

BURTON ROSENBERG
UNIVERSITY OF MIAMI

Contents

1. Introduction

Let's look carefully at a real world protocol used by two players who seek to decide fairly upon one bit. Often the players are the Flipper and the Chooser, and the Flipper flips and coin and hides the result in plain sight. The Chooser then calls heads or tails and the Flipper reveals the coin.

The elements of this protocol. are,

(1) The protocol does not favor the Flipper or the Chooser.
(2) The Flipper might or might not know the coin.
(3) The Chooser does not know the coin until it is revealed.
(4) The Flipper cannot change the coin outcome, especially once the Chooser choses.

A *bit commitment protocol* is a algorithm between communicating processes that achieves with this real-world coin flipping protocol achieves.

Date: November 14, 2025.

2. Definitions

As does the real world coin flipping protocol, a commitment protocol has two phases: *commit* and *reveal*.

Commit: In the commit phase the committing algorithm takes a value and a random value and calculates out a value, called the *commitment*. It sends the commitment to the chooser.

Reveal: In the reveal phase, the chooser is already in possession of the commitment. The chooser can do what it wishes, but at a certain point it asks the committer for information in order to open the commitment and reveal the value.

Given a space of values V, a space of random values R and a space of commitments C, denote the commitment by $c = Con(v, r)$.

In the commit phase, c is sent to the chooser. In a *canonical reveal protocol*, in the reveal phase the committer sends v and r. The chooser recomputes and checks if

$$c = Con(v, r).$$

If so, the chooser accepts that v is the committed value.

The two properties required for a commitment protocol are *hiding* and *binding*.

Definition 2.1 (Hiding)**.** In a commitment protocol is *perfectly hiding* or *computationally hiding* if given a commitment c the chooser has no or effectively no knowledge of v.

In the case of perfectly hiding, even unbounded computation cannot reveal v.

In the case of computationally hiding, an probabilistic polynomial time bound adversary playing the adversarial indistinguishability game wins with advantage not negligibly different from 1/2.

Definition 2.2 (Binding)**.** In a commitment protocol is *perfectly binding* or *computationally binding* if given a commitment c the chooser cannot reveal any value other than v in the reveal phase.

In the case of perfectly binding, even unbounded computation cannot reveal any value other than v in the reveal phase.

In the case of computationally binding, an probabilistic polynomial time bound adversary cannot find a value and a random value that reveals and different v in the reveal phase.

These properties are counters to the two ways of cheating on the commitment.

(1) Hiding means the chooser cannot cheat by learning v before the reveal. Perfect hiding means no chooser can cheat; hiding means no polynomially bound chooser can cheat with non-negligible probability.
(2) Binding means the committer cannot cheat by changing its commitment based on the choosers decision. Perfect binding means no committer can cheat; binding means that no polynomially bounded committer can cheat with non-negligible probability.

Lemma 2.1. No commitment scheme can be both perfectly hiding and perfectly binding.

3. Implementations

Construction 3.1 (Blum-Micali-Naor[1])**.** Let $g : V \to C$ be a pseudo-random permutation, and $h : V \to \{\,0, 1\,\}$ be predicate for which is hard to invert. Then a perfectly binding commit scheme is given by,

$$Com(b, r) = \langle\, g(r), b \oplus h(r)\,\rangle.$$

Proof: For any $g(r)$ this is only one r. This determines $h(r)$ and therefore b. Given $\langle\, c, \hat{b}\,\rangle$, to know b from $\hat{b} = b \oplus h(r)$ requires r. To find this from $g(r)$ is computationally infeasible as g is a pseudorandom permutation. □

Construction 3.2 (El Gamal)**.** Let p be a prime and q a large prime diving $p-1$. Let g and h be generators of a q sized subgroup of $\mathbb{Z}_p^\times$, for which $\log_g(h)$ is unknown. Then a perfectly binding commit scheme is given by,

$$Com(b, r) = \langle\, g^r, g^b\, h^r\,\rangle \pmod{p}.$$

Proof: This is an implementation of the Blum-Micali-Naor scheme, were g^r is the pseudorandom permutation and h^r is the predicate. □

[1]Moni Naor, *Bit Commitment Using Pseudorandomness*, STOC 1989.

Construction 3.3 (Pedersen[2])**.** Let p be a prime and q a large prime diving $p-1$. Let g and h be generators of a q sized subgroup of $\mathbb{Z}_p^\times$, for which $\log_g(h)$ is unknown. Then a perfectly hiding commit scheme is given by,

$$Com(b,r) = g^b\, h^r \pmod{p}.$$

Proof: Let $g = h^i$. Then all $c = ib + r \bmod q$ give the same commitment. Hence b is ambiguous and the scheme is perfectly hiding. However, and for the same reason, it is not perfectly binding. But finding a second commitment requires knowing or finding i. Hence it is computationally binding. □

4. Every NP set has a ZK Proof

Definition 4.1. Let $A, B \subseteq \Sigma^*$ be sets. A polynomial reduction from A to B, written $A \leq_P B$ is a polynomial time computable map $f : \Sigma^* \to \Sigma^*$ such that $a \in A \iff f(a) \in B$.

Theorem 4.1. If a set $A \subseteq \Sigma^*$ is NP complete, and can be recognized in zero-knowledge, then every NP set can be recognized in zero-knowledge.

Proof: Let B be in NP. Since A is NP complete, then $B \leq_P A$. For a $b \in B$, consider the $a = f(b) \in A$ where f is the polynomial reduction from A to B. Since B is recognizable in zero-knowledge, run the protocol for $a \in A$ and answer the same for $b \in B$. □

Theorem 4.2. Every NP set is recognizable in computational zero-knowledge.

Proof: We will give the construction for recognizing the NP-complete set of graph 3-colorabilitiy.

Given a graph $G = (V, E)$, it is 3-colorable if there exists a coloring $\chi : V \to \{ R, G, B \}$ such that for all edges $(v, v') \in E$, $\chi(v) \neq \chi(v')$.

An IP protocol for 3-colorability proceeds as follows. Let the graph be colorable with 3-coloring χ. The prover picks a random $\pi \in \mathcal{S}_3$. Then $\pi \circ \chi$ is also a 3-coloring.

[2]Torben Pryds Pedersen, *Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing,* CRYPTO '91

Using a perfectly binding commitment protocol commit to all the colors on all the vertices and send the commitments to the verifier. The verifier chooses an edge and asks that the vertex colors of the vertices incident on the edge be revealed.

The prover reveals the colors, and the verifier accepts only if the colors are different.

Completeness: If the graph is 3-colorable, then the protocol completes successfully and the verifier accepts.

Soundness: If the graph is not 3-colorable no matter what prover, some two vertices will be committed that have the same color and are connected by an edge. Given that the commitment is perfectly binding, the prover cannot avoid that if the verifier that picks this edge will reject. Since at least one edge can reject, the rejection is at least $1/|E|$.

This protocol will have to be repeated about $|E|$ to boost the soundness probability to a constant. And this can be done in polynomial time.

Zero-knowledge: The simulator colors the vertices at random and commits to the colors. It then runs the (arbitrary) verifier $\tilde{V}$ for the selection of the edge. Assuming the verifier could not break the commitments, the probability that the verifier asks for an edge that will pass verification is 2/3. Else the simulator outputs $\perp$.

We here use the computational hiding property, else the verifier $\tilde{V}$ would infer from the commitments a bad edge and ask for that edge. Is the success rate of this was high enough the verifier can force the simulator to abort too often.

Now we need to show that these ensembles are computationally indistinguishable,

$$\{\, S(G) \,|\, S(G) \neq \perp \,\} = \{\, View([P, \tilde{V}](G)) \,\}.$$

What the verifier sees is a set of commitments to colors, one for each vertices. For the right hand distribution, for no two commitments which share an edge will the commitments be to the say value. For the left hand distribution, there is not such a restriction.

Therefore (in a general notion of how this can be proved) to distinguish these distributions is to distinguish whether commitments commit to the same value or to different value. If this were done it would contradict the computationally hiding property of the commitment scheme.