Statistically hiding commitments based on one-way permutations

Scribe: Sina Jafarpour

1 Introduction

In cryptography, a commitment scheme or a bit commitment scheme is a method that allows a user to commit to a value while keeping it hidden, and while preserving the user's ability to reveal the committed value later. A useful way to visualize a commitment scheme is to think of the sender as putting the value in a locked box, and giving the box to the receiver. The value in the box is hidden from the receiver, who cannot open the lock (without the help of the sender), but since the receiver has the box, the value inside cannot be changed. Commitment schemes are important to a variety of cryptographic protocols, especially zero-knowledge proofs and secure computation. We will consider the naive case of commitment to either 0 or 1.

In order to come up with a commitment scheme C, the scheme should have the following two properties:

- Hiding: It should be hard to distinguish between commitment to 0 and commitment to 1 :

$$C(0) \approx C(1)$$

- Binding: There should be no way for a person who commits to one bit, to claim that he has committed to another value later:

$$\forall C \nexists \text{ (or just hard to find) } r_0, r_1 \text{ such that } C = C_{r_0}(0) = C_{r_1}(1)$$

We can define both statistical and computational definition for both of the hiding and biding properties. The commitment scheme is statistically hiding if the statistical distance of $C(0)$ and $C(1)$ is small and is computationally hiding if $C(0)$ is computationally indistinguishable from $C(1)$.

For binding, on the other hand, the scheme is statistically binding if $\nexists\, r_0, r_1$ such that $C_{r_0}(0) = C_{r_1}(1)$ and is computationally binding if a polynomial adversary cannot find r_0, r_1 such that $C_{r_0}(0) = C_{r_1}(1)$. As $C(0, U_n), C(1, U_n)$ are disjoint, there exists no commitment scheme with both statistical hiding and statistical binding properties.

It is easy to come up with a scheme that is statistically binding and computationally hiding from pseudorandom generators, and as we know using one way functions we can build pseudorandom generators. So suppose we have the following PRG: $G : \{0,1\}^n \to \{0,1\}^{3n}$. Then the following protocol is the desired commitment scheme:

1. Alice chooses $S \leftarrow U_n$, Bob chooses $R \leftarrow U_n$ and sends R to Alice
2. Alice: If wants to commits to 0 sends $G(S)$ to Bob, otherwise sends $G(S) \oplus R$ to Bob.

We have $G(S) \approx U_{3n} \approx U3n \oplus R \approx G(S) \oplus R$. So the scheme is computationally hiding. Also by the union bound:

$$\Pr[\exists S, S' : G(S) = G(S') \oplus R] \leq 2^{2n}.2^{-3n} = 2^{-n}$$

So the scheme is statistically binding. In fact, in many cases such as zero knowledge proofs, we want out scheme to be statistically binding and computationally hiding.

2 Coin tossing to the well protocol

In order to come up with the desired protocol, with shall define the notion of *collision resistant hash functions*. A family of hash functions $\mathcal{H}$ is a set of collision resistant hash functions if $\forall$ computational adversary A and negligible ϵ we have:

$$\Pr_{h \approx H}[A(h) = (x, x') \text{ such that } h(x) = h(x')] < \epsilon$$

Now suppose Alice wants to commit to a bit b. The protocol will be:

1. Bob chooses h from $\mathcal{H}$ uniformly at random and sends h to Alice.
2. Alice chooses x , r independently at random from U_n and sends $r, h(x), \langle x, r\rangle \oplus b$ as the commitment to b.

Claim 0.1. *x has min entropy $\geq \frac{n}{2}$*

The binding property is immediate. If Alice wants to pretend that she had committed to some other value, she has to come up with a $x' \neq x$ such that $h(x') = h(x)$, which is impossible by the CR property of the hash function.

For Hiding suppose $BAD = \{y' : |h^{-1}(y')| \leq 2^{\frac{n}{4}}\}$. Then $\Pr[y \in BAD] \leq 2^{\frac{n}{2}}.2^{\frac{n}{4}} = 2^{0.75n}$ which is a negligible fraction of 2^n, so we can assume $x \notin BAD$ which in this case x will have min entropy at least $\geq \frac{n}{4}$ and so we can use any strong extractor rather $\langle r, x\rangle$. Hence any scheme of the form $r, h(x), Ext_r(x) \oplus r$ can be used.

This protocol is called coin tossing to the well! However, we are interested to have a protocol that is not based on the collision resistance property of the hash functions. In next sections we will try to construct such a protocol.

3 Commitment from PRP

Suppose $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way permutation and $h : \{0,1\}^n \to \{0,1\}^{n-1}$ is CRH. First of all, we will show by a counter example that $h(f(x))$ is not CRH in general. Consider the case that h is universal hash function (a, b are chosen uniformly at random).

Counter example: $\exists f :\ h(f(x))$ is not CRH:

$$h(x) = ax + b \text{ over GF}(2^n)$$

$$\forall a, \exists A : h(x) = Ax + b \text{ over GF}(2)$$

Assume that if the first $\frac{n}{10}$ coordinates of y or the last $\frac{n}{10}$ coordinates of y are 0 then it is easy to find $f^{-1}(y)$. Then it is possible to attack $Af(x)+b :$ (A maps n bits to n-1 bits) We should find y, y' such that.

$$y \neq y', Ay + b = Ay' + b \tag{1}$$

$$y_1 = ... = y_{\frac{n}{10}} = 0 \tag{2}$$

$$y'_{n-\frac{n}{10}+1} = ... = y'_n = 0 \tag{3}$$

So we can pick a $z \in Kern(A)$ and partition it to y, y' with the above conditions. Then we can simply compute x, x' from y, y' and so we have a collision! However there exits functions such that $h(f(x))$ is CRH.

Now we will see how to design a coin to the well protocol from functions above. Suppose $h_1, ...h_n$ are linearly independent and

$$h_1 = 1\$\$...\$ \tag{4}$$
$$h_2 = 01\$\$...\$ \tag{5}$$
$$... \tag{6}$$
$$h_{n-1} = 0...1\$ \tag{7}$$

Then the following interactive protocol is a protocol to commit to b:

1. Alice samples x uniformly at random and computes $y = f(x)$
2. Bob sends h_1 to Alice, Alice computes $b_1 \oplus \langle h_1, y\rangle$ and sends it to Bob
3. Bob sends h_2 to Alice, Alice computes $b_2 \oplus \langle h_2, y\rangle$ and sends it to Bob
4. ...
5. Bob sends h_{n-1} to Alice, Alice computes $b_{n-1} \oplus \langle h_{n-1}, y\rangle$ and sends it to Bob

And in order to reveal, Alice simply sends $x, y = f(x)$ to Bob. Bob checks the validity and computes b. The hiding property is obtained since each round gives Bob information about just one bit of b, and by PRP property. Now we will prove the binding property:

Theorem 1. *If $\exists S^*$ such that with probability ϵ inverts both y_0, y_1 then $\exists$ I that can invert a one way permutation with probability $(\frac{\epsilon}{n})^{100}$*

Remark This is false if we give H to S^*. The blackbox construction of the protocol needs $\frac{n}{\log n}$ rounds.

In order to invert y, we need to choose $h_1, ..., h_{n-k}$ iteratively such that y is consistent with $h_1, ...h_{n-1}$.[1]

So we choose $h_1, ...h_{n-k}$ iteratively, and at each iteration we resample until we get right sample with respect to what the sender gives. In other words, we choose $h_1, ...h_{n-k}$ and then we repeatedly resample y until it is consistent, then we choose $h_{n-k}, ..., h_{n-1}$ at random and we hope that $S^*(h_1, ..., h_{n-1})$ inverts y. The heart of the analysis is as follow:

S^* runs the protocol, with probability ϵ it gives $f(x_0) = y, f(x_1) = y$. So sometimes the output is what we desire. Suppose we run the protocol until we are successful (in expectation 2(n-1) rounds), which means:[2]

$$D_I : y \approx U_n$$
$$h_1, ...h_{n-k} \approx \mathcal{C}(y)$$

The following two lemmas together imply the theorem:

Lemma 2. $\Pr_{y,h_1,...h_{n-k} \approx D_S}[S^*(h_1, ...h_n)$ *inverts* $y] \geq \frac{\epsilon}{1002^k}$ *and moreover* $\forall Z : \Pr_{(h,y)\approx D_{S^*}}[(h, y) \in Z] \leq 2^{-0.6k}$.

[1] $\forall h \in \{h_1, ..., h_{n-k}\} : S^*(h) = y$ where $h(y) = z$.

[2] We will denote the consistency by $\mathcal{C}$.

Proof. First we have the following:

$$\Pr[S^*(h,h') \text{ inverts } (y_0,y_1) \in \mathcal{C}(h_0,h_1)] \geq \epsilon$$
$$\Pr[S^*(h,h') \text{ inverts } (y_0,y_1) \in \mathcal{C}h] \geq \epsilon$$
$$\Pr_{h,h'\approx H, y\approx(h)}[S^*(h,h') \text{ inverts } y] \geq \frac{\epsilon}{2^k}$$

Now, let $Z_h = \{y \mid (h,y) \in Z\}$. Then with probability at least $1 - \frac{\epsilon}{100}$ we have: $\Pr_{y\approx\mathcal{C}h}[Z_h] \leq \frac{100}{\epsilon}2^{-0.6k} \leq 2^{-0.55k}$. So we get $|Z_h| \leq 2^k 2^{-0.55k} \leq 2^{0.45k}$ And therefore $\Pr[\exists y_0, y_1 \in Z_h : h(y_0) = h(y_1)] \leq |Z_h|^2 2^{-k} \leq 2^{-0.1k} \leq \frac{\epsilon}{2}$. Hence with probability ϵ at least one of the y_0 or y_1 is not in Z_h. □

Lemma 3. *D_I is at most $(2^{0.6k}, 10)-$skewed with respect to D_S meaning that except with probability $2^{-0.6k}$ if an event happens over D_S with probability ϵ, then the corresponding event happens over D_I with probability $\frac{\epsilon}{10}$.*

Proof. Let $D_0 = D_{S^*}, D_1, ..., D_{n-k} = D_I$. We will prove this lemma by a hybrid argument with skews :

$$P_i(h,y) \leq (1 + \frac{1}{n})P_{i+1}(h,y)$$

So at the end P_1 is incremented by a factor e.
At each step i:

- Let D_i : sample $h_1, ..., h_{n-k-1}$ at random from H.
- take $y \approx \mathcal{C}(h_1, ..., h_{n-k-1})$
- sample h_{n-k} from H consistent with y.

So how much skew are we introducing? If we consider the consistency graph, in D_{i-1} we choose y from left degrees at random and then we sample h to be consistent with that y. In D_i however, we first choose h and then we choose a y to be consistent with that h. This is equivalent to first choosing y according to the distribution of the weights of right nodes, and then choosing h uniformly from the neighbors of y. Then we have the following claim:

Claim 3.1. $\Pr_y[DEG(y) \geq (1 + \frac{1}{n^2})|H|] \leq 2^{-0.99k}$.

Proof: Define: $X_h = 1(\langle y, h\rangle = b_h)$, then:

$$\Pr[|\sum X_h - \frac{|H|}{2}| \geq \frac{1}{n}\frac{|H|}{2}] \leq \frac{4n}{|H|} \leq \frac{10n}{2^k} \leq 2^{-0.99k}$$

This is the chebyshev's inequality and we can use it because of the pairwise independence, which completes the proof. □

So far we saw the interactive hashing protocol using one-way permutation. Now we only sketch very roughly how it generalizes to getting the same thing using one-way function.

For the case that f is a 2^k to 1 regular one-way function, if we look at $f(x), h, h(x)$ where $|h(x)| = k$ and h is pairwise independent, what we get is "almost" one to one and behaves like a one-way permutation. The first try for k for the general case could be $\frac{\log sec(f(x))}{2}$ where $sec(f(x)) = |f^{-1}(x)|$. But there are two possibilities:

1. $k \gg \log |f^{-1}(x)|$. In this case $[f(x), h, h(x)]$ is not close to the uniform distribution. In this case we get binding property, but not hiding.

2. $k \ll \log |f^{-1}(x)|$. In this case it is easy to invert the interactive hashing procedure. So we get hiding property, but not the binding anymore.

One observation is that in the second case we still have some entropy left in x. So what we do is running another interactive hashing procedure with $h(x)$ of length $n - k$. They together determine x and so at least one of them is binding. In addition one of them is hiding. Now if we choose k at random with probability at least $1/n$ we guess correctly and we get:

- The first phase is binding.
- Both phases are hiding.

Moreover, we always have:

- One phase is binding.
- One phase is hiding.

This protocol is called $\frac{1}{n}$-weak 2-phase commitment. Then we shall amplify $1/n$ to ≈ 1. Assuming we have done this we still only have one phase which is binding, but we have fully hiding property. The main tool used to handle this issue is universal one-way hash function which we saw in previous session and we refer the reader for details to the references below.

References

[1] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil Vadhan . *Statistically Hiding Commitments and Statistical Zero-Knowledge Arguments from Any One-Way Function*, http://www.eecs.harvard.edu/ salil/papers/SHcommit-abs.html.

[2] Rafail Ostrovsky, Ramarathnam Venkatesan, Moti Yung . *Interactive Hashing Simplifies Zero-Knowledge Protocol Design*, Appeared In Proceedings of EUROCRYPT-93) Springer Verlag.

[3] Rafail Ostrovsky, Ramarathnam Venkatesan, Moti Yung . *Perfect Zero-Knowledge Arguments for NP Can Be Based on General Complexity Assumptions.* , Preliminary version appeared in Proceedings of advances in cryptology (CRYPTO-92) Springer-Verlag Lecture Notes in Computer Science.