Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing

Shai Halevi * Silvio Micali

MIT – Laboratory for Computer Science,
545 Technology Square, Cambridge, MA 02139

Abstract. We present a *very practical* string-commitment scheme which is *provably secure* based solely on collision-free hashing. Our scheme enables a computationally *bounded* party to commit strings to an *unbounded* one, and is optimal (within a small constant factor) in terms of interaction, communication, and computation.
Our result also proves that constant round statistical zero-knowledge arguments and constant-round computational zero-knowledge proofs for NP exist based on the existence of collision-free hash functions.

1 Introduction

String commitment is a fundamental primitive for cryptographic protocols. A commitment scheme is an electronic way to temporarily hide a value that cannot be changed. Such a scheme emulates by means of a protocol the following two-stage process. In Stage 1 (the Commit stage), a party called the Sender locks a message in a box, and sends the locked box to another party called the receiver. In Stage 2 (the De-commit stage), the Sender provides the Receiver with the key to the box, thus enabling him to learn the original message.

Commitment-schemes are very useful building blocks in the design of larger cryptographic protocols. They are typically used as a mean of flipping fair coins between two players, and also play a crucial part in some zero-knowledge proofs and in various types of signature schemes. Commitment schemes can also be used in scenarios like bidding for a contract, where committing to a bid rather than sending it in the clear can eliminate the risk of it being "leaked" to the competitors.

It is easily seen that if both parties have unlimited computational power, they cannot emulate the above process by just exchanging messages back and forth. Thus, at least one of the two parties must be *computationally bounded,* so that cryptographic technology can be applied. Indeed, many cryptographic implementations of commitment schemes have been suggested in the literature. A particularly important case of string commitment is when the Sender is computationally bounded, but the Receiver may have unlimited computational resources. This is so for at least three good reasons:

* E-mail: `shaih@theory.lcs.mit.edu`

1. Bounded-to-unbounded commitment schemes allow one to use suitable short security parameters even if the Receiver has a lot of computing power.
2. Bounded-to-unbounded commitment schemes protect the Sender even if the underlying cryptographic assumption happens to be wrong (say, if the computational difficulty of factoring is assumed, and the Receiver has a revolutionary algorithm for factoring).[2]
3. There are theoretical applications in which one must use bounded-to-unbounded commitment schemes to yield the desired result; for instance, to obtain constant-round computational zero-knowledge proofs for NP (as shown in [11]), or to obtain statistical zero-knowledge arguments for NP (as shown by [13, 16]).

1.1 Previous Work

Many commitment schemes in the unbounded-receiver model are known based on number-theoretic constructions. The first such scheme was suggested by Blum [3] in the context of flipping coins over the phone. Blum described a commitment scheme for one bit, which is based on the hardness of factoring. Blum's scheme calls for one or two modular multiplications and a k-bit commitment string for every bit which is being committed to (where k is the size of the composite modulus). A similar construction with the same efficiency parameters was later described by Brassard and Crépeau [4].

A more efficient construction, which is also based on the hardness of factoring, was introduced by Goldwasser, Micali and Rivest [12]. Their collision-free permutation-pairs enables one to commit to long messages using about the same amount of local computation as in Blum's scheme, but to send only a k-bit commitment string, regardless of the length of the message being committed to. Since then, this construction was used in many other works (e.g. [2, 8, 9, 10, 14]). One common problem of all these constructions is that they all rely on composite numbers of a special form (i.e., product of two primes which are both 3 mod 4). Thus they require a special initialization procedure in which these special-form numbers are established. Recently, Halevi [14] described a method which uses the GMR construction but avoids the need for this initialization step.

Several other constructions in the literature are based on the difficulty of extracting discrete-logarithms. In particular, Pedersen [18] and Chaum, van-Heijst and Pfitzmann [8], described a scheme in which the Sender can commit to a string of length k (where k is the size of the prime modulus) by performing two modular exponentiations, and sending a k-bit commitment string.

There were also a few implementations of commitment-schemes using more generic complexity assumptions. Naor [15] presented a commitment scheme in the bounded receiver (and unbounded sender) model, which can be implemented

[2] Moreover, such schemes still protect the Receiver in case the underlying cryptographic assumption is "semi-wrong" and the De-commit stage occurs soon thereafter the Commit one (e.g., although the Sender knows how to factor, he can not do it in just one hour).

using any pseudorandom-generator. As opposed to the previous schemes, however, Naor's scheme is interactive, and it requires 2 rounds of communication to commit to a string. The Sender in this scheme generates an $O(n)$-bit pseudorandom string and sends an $O(n)$-bits commitment string in order to commit to an n-bit message. In the unbounded receiver model - Naor, Ostrovki, Venkatesan and Yung [16] described a construction which is based on any one-way permutation. Their scheme is particularly inefficient, however, in that it calls for $2k$ rounds of communication and one application of the one-way permutation for each bit which is being committed to.

In addition to the above work, Several researchers showed that a commitment scheme for a single bit can be implemented using "quantum computing devices". The first such scheme was the (flawed) scheme by Bennet and Brassard [1]. Better schemes were later suggested by Brassard and Crépeau [5] and Brassard, Crépeau, Jozsa and Langlois [6].

1.2 Our result

We present a commitment scheme which is provably secure under a standard assumption in the model in which the Sender is computationally bounded and the Receiver is all-powerful. Moreover, this scheme is more efficient than many other schemes discussed in the literature (even ones where both parties are computationally bounded).

The assumption under which we prove the scheme secure is the existence of collision-free hash functions. These are functions that map strings of arbitrary length to fixed-length ones, so that it is computationally infeasible to find two different pre-images of a common output string. Collision-free hash functions (often referred to as message-digest functions) are widely believed to exist, and are used extensively in cryptography, including in digital signatures schemes, authentication schemes, etc.

Efficiency. Let us now elaborate on the efficiency of our scheme. As for any other protocol, there are three important resources to consider: interaction, communication, and computation.

Interaction. Protocols are typically interactive because their parties communicate by exchanging messages back and forth. Interaction is, however, very expensive; because the number of rounds of communication heavily weigh on the overall running time of a protocol. Notably, our scheme is *non-interactive*. That is, in each stages the Sender sends a single message to the Receiver, who needs not to reply at all.

Communication. Another important resource in a protocol is the number of bits sent by its parties. In a commitment scheme, this is measured against the length of the message being committed to (denoted by n), and the security parameter (denoted by k).[3]

[3] The security parameter may control the success probability of the Sender in changing her message after having committed to it, as well as the probabilistic advantage the

It is easy to see that, in any commitment scheme, (1) the number of bits exchanged during the Commit Stage must be at least k, and (2) that the number of bits exchanged during the entire protocol must, on the average, be at least $n+k$.

Our scheme requires that the Sender transmits $O(k)$ bits in the Commit Stage and $n+O(k)$ bits in the De-commit Stage (where the constant hidden in the $O(\cdot)$ notation is at most 9). Thus the overall communication complexity of our scheme is optimal within a constant factor.

Computation. A third crucial resource is the amount of (local) computation for the parties. Our scheme calls for (1) a single collision-free hashing of the message; (2) one collision-free hashing of a random $O(k)$-bit string (typically $k=128$); and (3) one evaluation of a universal hash function on an $O(k)$-bit string (typically by multiplying this string by a binary matrix). [4]

The efficiency of our scheme is comparable to that of schemes which achieve much weaker notion of security in weaker models. Indeed, it seems that even in the bounded-to-bounded model, the most efficient (reasonable) strategy for committing to a string σ consists of having the Sender transmit to the Receiver the value $F(\sigma)$, where F is a "good hash function". However, such a strategy is *not* secure enough. It is clear, for example, that upon receiving $F(\sigma)$, even a bounded Receiver may dismiss possible candidate strings σ' from consideration by checking that $F(\sigma') \neq F(\sigma)$. It is therefore perhaps surprising that our scheme succeeds in being almost as efficient as the above "minimal" one, while offering strong security in a more adversarial model.

We wish, however, to point out that our commitment scheme offers slightly different security assurances than those offered by prior schemes in the unbounded-receiver model. In those works, the Receiver had absolutely zero advantage in guessing what the Sender's message may be from its commitment. In our case, instead, the Receiver may obtain some advantage, but this advantage is provably *exponentially small in the security parameter*. Overall a small price, and one worth paying in order to have an efficient commitment scheme with a reasonable assumption. In Figure 1 we sketch the parameters of some of the schems in the literature, as compared to the scheme which we suggest in this paper.

Complexity-theoretic implications. Since our scheme works in the unbounded-receiver model, it also has complexity-theoretic implications. Namely, using our protocol in the constructions of [13, 16] yields *constant round* statistical zero-knowledge arguments for NP, and using it in the construction of [11] yields constant-round computational zero-knowledge proofs for NP.

Thus our result implies that both of these exist if collision-free hashing exists. Note that constant-round protocols of both kinds were previously only known to exist based on number-theoretic assumptions (since the bit-commitment in

Receiver may get about the message from its commitment.

[4] See Section 2 For a definition and implementation of universal-hashing. We note that evaluating a universal hash function is typically cheaper than evaluating a collision-free hash function.

Commiting to an n-bit message with security-parameter k

the scheme	works in model	complexity assumption	# rounds for commitment
GMR-based [14]	unbounded-receiver	factoring Blum-integers	1-round
Pedersen [18]	unbounded-receiver	discrete-log	1-round
Naor [15]	bounded-receiver	pseudorandom-generator	2-rounds
NOVY [16]	unbounded-receiver	one-way permutation	$2k$-rounds
This paper	unbounded-receiver	collision-free hashing	1-round

the scheme	length of commit-string	local computations	typical $k =$
GMR-based	$O(k)$	n modular multiplications	1024
Pedersen	$O(\max(k, n))$	$O(\max(k, n))$ modular multiplications	1024
Naor	$O(\max(k, n))$	generating $O(n)$ pseudorandom bits. error-correction encoding of message	64 (?)
NOVY	$O(n \cdot k)$	n applications of one-way permutation $n \cdot k^2$ XOR operations	≥ 64 ?
This paper	$O(k)$	1 collision-free hashing of n-bit message. 1 collision-free hashing of $O(k)$-bit string. 1 universal-hashing of $O(k)$-bit string.	128

Fig. 1. Comparison between commitment-schemes

[16] uses many rounds). Hence this work proves that these protocols too can be shown to exist based on generic complexity assumption.

1.3 A False Solution

Before presenting our scheme, it is useful to point out why simpler constructions based on collision-free hashing do NOT work. For the purpose of the discussion below we still rely on an intuitive understanding of what a commitment scheme is and when it does or does not work. The reader is referred to Section 2 for a more formal description.

Let MD (for Message-Digest) be a collision-free hash function. One example of a false solution is provided by the "minimal" strategy discussed above (i.e., having the Sender commit to a message M simply by sending $C = MD(M)$ to the Receiver, and de-commit by simply sending M.)

In an effort to fix the flaw in this simple scheme, one may try to have the Sender first pad the message M with a sufficiently-long random string R, and then sends $C = MD(M \circ R)$ (where $M \circ R$ is the concatenation of M and R). Unfortunately, this construction may not work either (even when the Receiver is bounded). Indeed, it may be that MD, though collision-free, leaks some of the bits of M. In addition, in our more difficult model, the unbounded Receiver may get a good probabilistic advantage in guessing which of two messages M and M' is more likely to be the committed one. For instance, he can compute the size of the pre-image of C when the message is M, compare it to the size of the pre-image of C when the message is M', and guess accordingly.

Of course, the latter attack can be prevented if MD has additional properties besides being collision-free (e.g., if MD is "regular"). However, we do NOT want to assume these additional properties in our construction, since the more assumptions we make, the less likely it is that these assumptions are true. Yet we wish to have an *efficient* commitment scheme whose security against an unbounded Receiver is PROVABLE.

1.4 Our Construction in a Nutshell

Our solution is similar in spirit to the second construction above, but "adds random bits" to the message in a more sophisticated way, thus enabling a proof of correctness against an unbounded Receiver without ANY additional assumptions.

For clarity of presentation we present the construction in two steps. At first we present a simple scheme in which the commitment string is of length $O(n+k)$, and then we show how to modify it so as to get an $O(k)$-bit commitment.

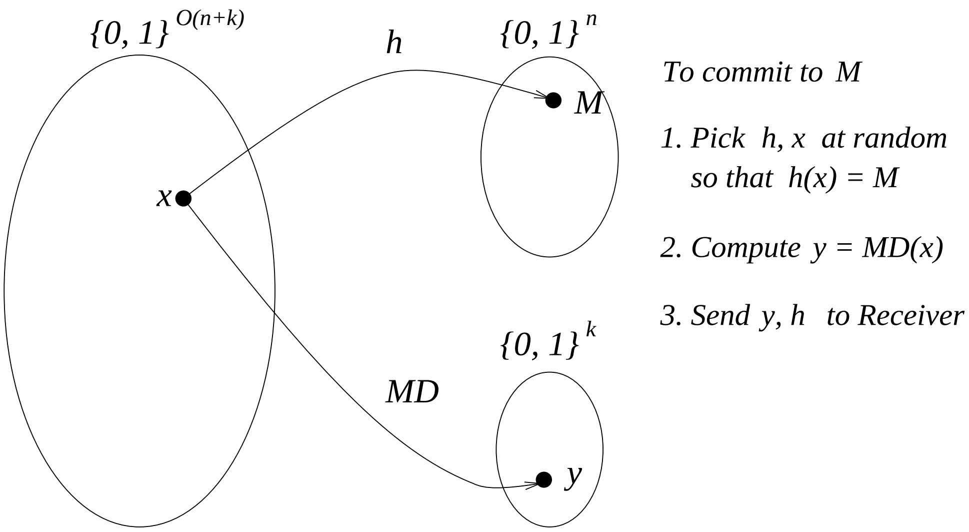

Fig. 2. A simple commitment-scheme. h is a universal hash-function and MD is a message-digest function.

THE FIRST SCHEME. The first scheme uses universal hashing as a tool for "adding randomness" to the message. Universal hashing was introduced by Carter and Wegman [7] and it plays a very important role in many areas of computer-science. Intuitively, a family of hash functions $H = \{h : A \rightarrow B\}$ is universal if picking a function at random from H "has the same effect" as picking a totaly random function from A to B. See Section 2 for a formal definition and a construction of universal hash functions.

To commit to an n-bit message M, the Sender picks at random a string x of length $O(n+k)$ and a universal hash function $h : \{0,1\}^{O(n+k)} \rightarrow \{0,1\}^n$ so that $h(x) = M$. Then she applies the collision-free hash function MD to the random string x to get $y = MD(x)$ and sends $C = \langle y, h\rangle$ to the Receiver. See Figure 2 for an illustration of this scheme. Since there are known constructions of universal hash functions in which it only takes $O(n+k)$ bits to describe any function in the family, then the length of the commitment string is $O(n+k)$.

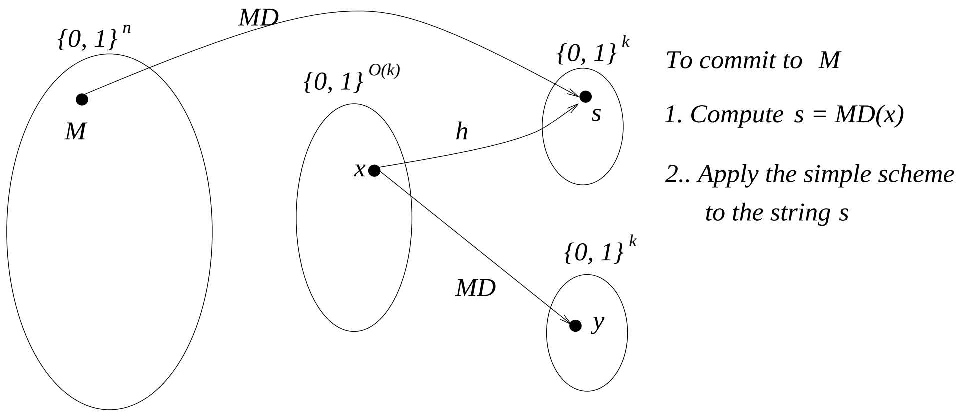

Fig. 3. The modified commitment-scheme.

GETTING AN $O(k)$ COMMITMENT STRING. To reduce the size of the commitment string we observe that instead of applying the above scheme to the message M itself, we can first apply the collision-free hash function to the message, thus obtaining a k-bit string $s = MD(M)$, and then have the sender commit to s. In terms of the commitment-scheme above, this means that we have $n = k$ and therefore the commitment is of length $O(k+k) = O(k)$ bits. See Figure 3 for an illustration of the modified scheme.

2 Preliminaries

2.1 Universal Hashing

Universal hashing was introduced by Carter and Wegman [7] and it plays a very important role in many areas of computer-science. Let S and T be two sets, and let H be a family of functions from S to T. We say that H is a universal family of hash functions if for any two different elements $s_1 \neq s_2$ in S and for any two elements t_1, t_2 in T we have

$$\Pr_{h \in H}[h(s_1) = t_1 \text{ and } h(s_2) = t_2] = \frac{1}{|T|^2}$$

For an easy example, when $S = \{0,1\}^l$ and $T = \{0,1\}^n$ we can have $H = \{h_{A,b} : A \in \{0,1\}^{n \times l}, b \in \{0,1\}^n\}$ where we define $h_{A,b}(r) \stackrel{\text{def}}{=} Ar + b$ (all the operations take place in a linear space over $GF(2)$). To specify a function from this family we need $n(l+1)$ bits. A more efficient construction is to restrict A to be a Toeplitz matrix. That is, A should be fixed on the diagonals, $A_{i,j} = A_{1+1,j+1}$. This way we can describe any function in H using only $2n + l - 1$ bits.

2.2 Statistical Difference

Let $\mathcal{D}_1, \mathcal{D}_2$ be two probability distributions over the same base set S. The *statistical difference* between $\mathcal{D}_1$ and $\mathcal{D}_2$, denoted $||\mathcal{D}_1 - \mathcal{D}_2||$, is defined as

$$||\mathcal{D}_1 - \mathcal{D}_2|| \stackrel{\text{def}}{=} \sum_{s \in S} \left| \Pr_{\mathcal{D}_1}[s] - \Pr_{\mathcal{D}_2}[s] \right|$$

Notice that for any two distributions $\mathcal{D}_1, \mathcal{D}_2$, we always have $0 \leq ||\mathcal{D}_1 - \mathcal{D}_2|| \leq 2$.

2.3 Negligible Functions

We say that a non-negative function $f(n)$ is *negligible* if as n gets larger, $f(n)$ goes to zero faster than any fixed polynomial in $1/n$. That is, for any constant $c > 0$ there is an integer n_c so that for all $n > n_c$, $f(n) < 1/n^c$.

2.4 Feasible Algorithms

We say that a (possibly randomized) algorithm A is *feasible*, if the running-time of A on inputs of length n is bounded by some polynomial in n.

2.5 Collision-Free Hashing

In this extended abstract we only provide an informal description of what a collision-free hash function is. Intuitively, a collision-free hash function is a function $MD : \{0,1\}^* \to \{0,1\}^k$ (for some integer k) so that it is infeasible to find two different strings $x \neq y$ so that $MD(x) = MD(y)$. That is, any feasible algorithm can only succeed in finding two such strings with negligible probability (where the probability is measured against k). For practical purposes, the SHA algorithm [19] is often considered to be such a function (for $k = 160$).

From the formal point of view, however, we must have a family of functions from $\{0,1\}^*$ to $\{0,1\}^k$, and the infeasibility requirement is formulated with respect to a function which is chosen at random from that family. Moreover, to get a meaningful definition we must have infinitely many such families, each is indexed by a different k.

2.6 Commitment Schemes

In this paper we do not try to give the most general definition possible for a commitment scheme. Instead, we restrict ourself to only talk about non-interactive schemes, which are the ones that we discuss.

THE SYNTACTIC STRUCTURE OF A COMMITMENT SCHEME. A commitment scheme is a protocol of two phases (the *Commit* and *De-commit* phases) between two parties (the Sender and the Receiver). Both parties share a common input, which is the security parameter of the system encoded in unary (we denote this by 1^k). Besides 1^k, the Sender also has another input, m, which is the message string to which she wants to commit herself. When used inside some other protocol, the parties may also have other inputs which represent their history at the point where the commitment scheme is being invoked.

The parties execute the Commit phase first and the De-commit phase at some later time. Typically, when used in another protocol, there will be some other parts of that protocol between the Commit and the De-commit phases.

During the Commit phase the Sender sends to the Receiver a commit-string c and during the De-commit phase the Sender sends to the Receiver a de-commit string d. From c and d the Receiver computes the message m and then checks that m is consistent with c and d.

In a non-interactive commitment scheme we can view the Sender as a probabilistic algorithm SEND which on input $(1^k, m)$ outputs a pair (c, d), and the Receiver as another algorithm RECEIVE which on input $(1^k, c, d)$ outputs either a string m or the special symbol $\perp$ (meaning that the strings c, d are not the commit/de-commit strings for any message).

THE SEMANTICS OF A COMMITMENT SCHEME. The semantics of a commitment scheme should ensure that after the Commit phase the Receiver does not know anything about the message yet, but the Sender can not change it anymore, and that after the De-commit phase the Receiver is able to learn the message.

The definition of what it means for the Receiver "not to know anything about m", and for the Sender "not to be able to alter m" depends on the computational power of the parties. In the context of this paper, the Sender is bounded to probabilistic polynomial-time and the Receiver has unbounded computational power. Thus, we require the following properties

Meaningfulness: If both the Sender and the Receiver follow their parts in the protocol, then the message m which the Receiver computes from (c, d) after the De-commit phase is equal to the Sender's input message. That is,

$$\forall k \in \mathcal{N}, m \in \{0,1\}^*,\ \textsc{Receive}(1^k, \textsc{Send}(1^k, m)) = m$$

Secrecy: For any string $m \in \{0,1\}^*$, let $C_k(m)$ denote the distribution over the commit-strings for m. That is, $C_k(m)$ is the distribution on the first coordinate of the pair which is obtained by running the algorithm SEND$(1^k, m)$. We require that

$$\forall m_1, m_2 \in \{0,1\}^*,\ \|C_k(m_1) - C_k(m_2)\| = O(2^{-k})$$

Non-Ambiguity: It is computationally infeasible to generate a commit-string c and two de-commit strings d, d' such that the Receiver would compute one message m from (c,d) and a different message from (c,d').
This means that for any feasible algorithm SEND', we have that

$$\Pr\left[(c,d,d') \leftarrow \text{SEND}'(1^k);\ \begin{array}{l}\text{RECEIVE}(1^k,c,d) \neq \perp,\\ \text{RECEIVE}(1^k,c,d') \neq \perp,\\ \text{RECEIVE}(1^k,c,d) \neq \text{RECEIVE}(1^k,c,d')\end{array}\right] = \text{negligible}(k)$$

where the probability is taken over the random coin-tosses of SEND' (and of RECEIVE if it happens to be probabilistic).

REMARK 1. In the above definition we chose to control both the statistical advantage that the Receiver gets from the Commit phase and the probability that the Sender can cheat in the De-commit phase by a single security parameter k. It is possible to have two different parameters controlling these two aspects. The generalization of the scheme we suggest below for that case is trivial.

REMARK 2. In the first scheme we present, the Secrecy property only holds for messages of the same length. That is, the Receiver does learn the length of the message from the commitment string. However, in the final construction this does not matter, since we only use the first scheme to commit to messages of some fixed length.

3 The First Scheme

In this section we present a commitment scheme in which the length of the commitment string is $O(n+k)$, where n is the length of the message being committed to and k is the security parameter. Later, in Section 4 we show how this can be improved to get an $O(k)$ commitment string.

For the rest of this section, fix the message length n and the security parameter k and set $L = 4k + 2n + 4$. Let $MD : \{0,1\}^L \rightarrow \{0,1\}^k$ be a collision-free hash function. That is, we assume that the Sender can not find $x \neq y \in \{0,1\}^L$ so that $MD(x) = MD(y)$. Also, let H be a universal family of hash functions from $\{0,1\}^L$ to $\{0,1\}^n$.

THE COMMITMENT SCHEME To commit to a message $m \in \{0,1\}^n$, the Sender first picks a random $r \in \{0,1\}^L$ and computes $y = MD(r)$ and then picks a random function[5] $h \in H$ for which $h(r) = m$.

The commit-string is $c = \langle h, y\rangle$, and the de-commit string is $d = r$. To de-commit m the Sender sends r to the Receiver, who verifies that $y = MD(r)$ and computes $m = h(r)$. See Figure 2 for an illustration of that scheme.

This scheme is indeed non-interactive and requires very little local computation. If we use the construction of universal hashing which we present in Section 2

[5] In the construction of universal hashing which we describe in Section 2, this can be done by picking A at random and computing $b = m - Ax$.

then the size of the commitment-string is $|h| + |y| = (L + 2k) + k = 7k + 2n = O(k + n)$ as promised. The only thing left to do is to prove that this is indeed a commitment scheme.

3.1 Analysis of the Scheme

The analysis if the scheme is fairly straightforward (though a little technical): The non-ambiguity part is obvious, as it is clear that being able to open the commitment in two different ways implies that the Sender can find a collision in MD.

The less obvious part is to prove that the Receiver gets almost no statistical advantage about m from the commit string. To show this, we need to show that for any two messages m_1, m_2, the distributions $C_k(m_1), C_k(m_2)$ are statistically close (up to 2^{-k}).

Theorem 1. *For all $k \in \mathcal{N}$ and $m_1, m_2 \in \{0,1\}^n$, $\|C_k(m_1) - C_k(m_2)\| < 2^{-k}$.*

Proof. Before starting the proof, let us first set some notations: In the scheme above, we denote by "$C_k(m) = \langle h, y\rangle$" the event that on input $(1^k, m)$, the Sender sends $\langle h, y\rangle$ as the commitment string. For any $y \in \{0,1\}^k$ we denote by $S(y)$ the size of the pre-image of y under MD. That is,

$$S(y) \stackrel{\text{def}}{=} |MD^{-1}(y)| = \left|\{r \in \{0,1\}^L : \ MD(r) = y\}\right|$$

Also, for any $y \in \{0,1\}^k, m \in \{0,1\}^n, h \in H$ we let $T(y, h, m)$ denote the size of the intersection between $MD^{-1}(y)$ and $h^{-1}(m)$. That is

$$T(y, h, m) \stackrel{\text{def}}{=} |\ MD^{-1}(y) \cap h^{-1}(m)\ | = \left|\{r \in \{0,1\}^L : \ MD(r) = y \ \&\ h(r) = m\}\right|$$

The following proof is somewhat technical, but still rather straightforward. For the sake of readability we divide it into four steps: In Step 1 we give an explicit expression for the probability of the event $C_k(m) = \langle h, y\rangle$ it terms of $T(y, h, m)$. In Step 2 we use it to develop an explicit expression for $\|C_k(m_1) - C_k(m_2)\|$. In Step 3 we give an upper-bound on a key term of the last expression, and in Step 4 we plug this upper bound back in the expression to get the final bound on $\|C_k(m_1) - C_k(m_2)\|$.

Step 1. We start the proof by looking at any $y_0 \in \{0,1\}^k, m_0 \in \{0,1\}^n, h_0 \in H$ and evaluating the probability of the event $C_k(m_0) = \langle h_0, y_0\rangle$. To do that, we first consider some string $r_0 \in \{0,1\}^L$ and evaluate the probability of the event $C_k(m_0) = \langle h_0, y_0\rangle$ given that r_0 was chosen by the Sender during the Commit phase. We denote this probability by $\Pr[C_k(m_0) = \langle h_0, y_0\rangle \mid r_0]$.

Clearly, if $r_0 \notin MD^{-1}(y_0)$ or $r_0 \notin h_0^{-1}(m_0)$ then picking r_0 rules out the possibility of outputting $\langle h_0, y_0\rangle$ as the commitment string. So it is left to consider only those r's that are in $MD^{-1}(y_0) \cap h_0^{-1}(m_0)$.

For $r_0 \in MD^{-1}(y_0) \cap h_0^{-1}(m_0)$, after picking r_0 it is guaranteed that y_0 is part of the commitment string. As for h_0, in order for it to be in the commitment string we need to "hit it" when we pick a function at random from the set $\{h \in H : h(r_0) = m_0\}$. Since H is a uniform hash-family, we know that for all r_0, m_0 the size of that set is exactly $|H|/2^n$, so the probability of picking h_0 from it is exactly $2^n/|H|$. Thus we get for all m_0, r_0, h_0, y_0

$$\Pr[C_k(m_0) = \langle h_0, y_0\rangle \mid r_0] = \begin{cases} \frac{2^n}{|H|} \text{ if } r_0 \in MD^{-1}(y_0) \cap h_0^{-1}(m_0) \\ 0 \quad \text{otherwise} \end{cases} \tag{1}$$

Now we can compute the probability that $C_k(m_0) = \langle h_0, y_0\rangle$ as

$$\begin{aligned} &\Pr[C_k(m_0) = \langle h_0, y_0\rangle] \\ &= \sum_{r\in\{0,1\}^L} \Pr[r] \cdot \Pr[C_k(m_0) = \langle h_0, y_0\rangle \mid r] \\ &= \sum_{r\in MD^{-1}(y_0)\cap h_0^{-1}(m_0)} 2^{-L} \cdot \frac{2^n}{|H|} \\ &= \frac{2^n}{2^L|H|} \cdot \left|MD^{-1}(y_0) \cap h_0^{-1}(m_0)\right| \;=\; \frac{2^n T(y_0, h_0, m_0)}{2^L|H|} \end{aligned} \tag{2}$$

Step 2. For the rest of the prove, fix any $m_1, m_2 \in \{0,1\}^n$, and we try to give an upper-bound on the statistical difference $||C_k(m_1) - C_k(m_2)||$. By definition of statistical difference, and using Equation 2, we have

$$\begin{aligned} &||C_k(m_1) - C_k(m_2)|| \\ &= \sum_{y,h} \mid \Pr[C_k(m_1) = \langle h, y\rangle] - \Pr[C_k(m_2) = \langle h, y\rangle] \mid \\ &= \sum_{y,h} \frac{2^n}{2^L|H|} \mid T(y,h,m_1) - T(y,h,m_2) \mid \\ &= 2^{n-L} \sum_y \frac{1}{|H|} \sum_h \mid T(y,h,m_1) - T(y,h,m_2) \mid \end{aligned} \tag{3}$$

Step 3. In this step we prove an upper-bound on the expression

$$E_y \stackrel{\text{def}}{=} \frac{1}{|H|} \sum_h \mid T(y,h,m_1) - T(y,h,m_2) \mid$$

Notice that for any $y \in \{0,1\}^k$, E_y is the expected value of the quantity $|T(y,h,m_1) - T(y,h,m_2)|$ when h is chosen at random from H.

So fix any $y_0 \in \{0,1\}^k$ and consider its pre-image $MD^{-1}(y_0)$. For any vector $r \in MD^{-1}(y_0)$ we define a random variable (over the random choice of h)

$$\rho_r = \begin{cases} 1 \; h(r) = m_1 \\ -1 \; h(r) = m_2 \\ 0 \text{ otherwise} \end{cases}$$

Then for any given h we have by definition $T(y,h,m_1) - T(y,h,m_2) = \sum_r \rho_r$.

Now notice that for any r we have $E_h[\rho_r] = 0$ and $E_h[\rho_r^2] = \frac{2}{2^n}$. Furthermore, since H is a universal-hash family then the ρ_r's are pairwise independent. Applying Chebyshev's inequality we get for any $\delta > 0$

$$\begin{aligned}&\Pr_h\left[|T(y_0,h,m_1) - T(y_0,h,m_2)| > \delta\right] \\ &= \Pr_h\left[\left|\sum_{r\in MD^{-1}(y_0)} \rho_r\right| > \delta\right] < \frac{2|MD^{-1}(y_0)|}{\delta^2\cdot 2^n} = \frac{S(y_0)}{\delta^2\cdot 2^{n-1}}\end{aligned} \tag{4}$$

In particular, if we substitute $\delta = (S(y_0)^2/2^{n-1})^{1/3}$ in Equation 4 we get

$$\begin{aligned}&\Pr_h\left[\Big|T(y_0,h,m_1) - T(y_0,h,m_2)\Big| > \left(\frac{S(y_0)^2}{2^{n-1}}\right)^{1/3}\right] \\ &< \frac{S(y_0)}{2^{n-1}}\cdot\left(\frac{2^{n-1}}{S(y_0)^2}\right)^{2/3} = \left(S(y_0)\cdot 2^{n-1}\right)^{-1/3}\end{aligned} \tag{5}$$

Using Equation 5 and the fact that $|T(y_0,h,m_1) - T(y_0,h,m_2)| \le S(y_0)$ for all h, we can bound E_{y_0} by

$$\begin{aligned}E_{y_0} &= E_h[|T(y_0,h,m_1) - T(y_0,h,m_2)|] \\ &\le S(y_0)\cdot\Pr_h\left[\Big|T(y_0,h,m_1) - T(y_0,h,m_2)\Big| > \left(\frac{S(y_0)^2}{2^{n-1}}\right)^{1/3}\right] \\ &\quad + \left(\frac{S(y_0)^2}{2^{n-1}}\right)^{1/3}\cdot\Pr_h\left[\Big|T(y_0,h,m_1) - T(y_0,h,m_2)\Big| \le \left(\frac{S(y_0)^2}{2^{n-1}}\right)^{1/3}\right] \\ &\le S(y_0)\cdot\left(S(y_0)\cdot 2^{n-1}\right)^{-1/3} + \left(\frac{S(y_0)^2}{2^{n-1}}\right)^{1/3} = 2\left(\frac{S(y_0)^2}{2^{n-1}}\right)^{1/3}\end{aligned} \tag{6}$$

STEP 4. We are now ready to give an upper bound on the statistical difference $\|C_k(m_1) - C_k(m_2)\|$. We substitute the bound from Equation 6 into the expression of Equation 3 to get

$$\|C_k(m_1) - C_k(m_2)\| \le 2^{n-L}\sum_y 2\left(\frac{S(y)^2}{2^{n-1}}\right)^{1/3} = 2^{(2n+4)/3-L}\sum_y S(y)^{2/3} \tag{7}$$

Recall that $\sum_y S(y) = 2^L$ (because every $r \in \{0,1\}^k$ is in the pre-image of some y). Since the function $f(x) = x^{2/3}$ is concave then the expression $\sum_y S(y)^{2/3}$ is maximized when all the $S(y)$'s are equal (i.e., when $S(y) = 2^{L-k}$ for all y). Hence

$$\sum_y S(y)^{2/3} \le 2^k\cdot\left(2^{L-k}\right)^{2/3} = 2^{(k+2L)/3}$$

Substituting this last bound in Equation 7, and using the fact that $L = 2n+4k+4$ we get

$$\|C_k(m_1) - C_k(m_2)\| \le 2^{(2n+4)/3-L}\cdot 2^{(k+2L)/3} = 2^{(2n+k+4-L)/3} = 2^{-k} \tag{8}$$

4 Getting an $O(k)$-Bit Commitment String

In this section we describe briefly how to modify the above scheme so as to get an $O(k)$-bit commitment scheme. On a message m, the Sender first computes the k-bit string $s = MD(m)$, and then apply the above commitment string to the string s. To de-commit m the Sender sends both the message m and the de-commit message of the first scheme. The Receiver checks that s is the string being committed to in the first message and that $MD(m) = s$.

Since we execute the first scheme on a message of length k, then the commitment-string is of length $7k + 2k = 9k$, regardless of the message length.

It is immediate to prove that if MD is a collision-free hash function then this scheme too is a commitment scheme. We omit this proof from this extended abstract.

5 Open Problems

An interesting open problem is to reduce the assumptions needed for a commitment scheme. In particular, it is not known whether universal one-way hash functions (in the sense of Naor and Yung [17]) are sufficient for commitment schemes in the unbounded receiver model. [6]

Another open problem is to design efficient commitment schemes which have nice homomorphism properties. In particular, in some scenarios it is desirable to be able to compute a commitment for $a + b$ (or $a \cdot b$) from the commitments to a and to b.

Acknowledgments. The authors thank Oded Goldreich and the Crypto committee members for their useful comments.

References

[1] C.H. Bennett and G. Brassard Quantum Cryptography: Public Key Distribution and Coin Tossing. In *Proc. of IEEE International Conf. on Computers, Systems, and Signal Processing*, IEEE, 1984, pages 175-179.

[2] G. Bleumer, B. Pfitzmann and M. Waidner. A Remark on a Signature Scheme where Forgery can be Proved. In I.B. Damgård, editor, *Proc. of Eurocrypt'90*, Lecture Notes in Computer Science, volume 473, Springer-Verlag, 1990. pages 441–445.

[3] M. Blum. Coin flipping by telephone. In *Proc. IEEE Spring COMPCOM*, pages 133–137. IEEE, 1982.

[4] G. Brassard and C. Crèpeau. Nontransitive Transfer of Confidence: A Perfect Zero-Knowledge Interactive Protocol for SAT and Beyond. In *Proc. 27th IEEE Symp. on Foundations of Comp. Science*, IEEE, 1986. pages 188–195.

[6] It is easy to show, however, that the collision-freeness assumption is necessary for *non-interactive* commitment schemes in the unbounded receiver model, in which the commitment string is shorter than the message itself.

[5] G. Brassard and C. Crèpeau. Quantum bit commitment and coin tossing protocols. In A.J. Menezes and S.A. Vanstone, editors, *Proc. Crypto '90*, Lecture Notes in Computer Science, volume 537. Springer-Verlag, 1991. pages 49–61.
[6] G. Brassard, C. Crèpeau, R. Jozsa and D. Langlois. A Quantum Bit Commitment Scheme Provably Unbreakable by Both Parties. In *Proc. 34th IEEE Symp. on Foundations of Comp. Science*, IEEE, 1993.
[7] L. Carter and M. Wegman. Universal Hash Functions. J. of Computer and System Science 18, 143-154 (1979).
[8] D. Chaum, E. van Heijst and B. Pfitzmann. Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer. In J. Feigenbaum, editor, *Proc. Crypto '91*, Lecture Notes in Computer Science, volume 576, Springer-Verlag, 1992. pages 470–484.
[9] I.B. Damgård, Practical and Provably Secure Release of a Secret and Exchange of Signatures. T. Helleseth, editor, *Proc. EuroCrypt '93*, Lecture Notes in Computer Science, volume 765, Springer-Verlag, 1994. pages 200–217.
[10] I.B. Damgård, T.P. Pedersen, and B. Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In D.R. Stinson, editor, *Proc. Crypto '93*, Lecture Notes in Computer Science, volume 773. Springer, 1994. pages 250–265.
[11] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proofs Systems for NP. *Journal of Cryptology*, Vol. 9, No. 2, 1996.
[12] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, April 1988.
[13] Moti Yung and Russell Impagliazzo. Direct minimum-knowledge computations. In C. Pomerance, editor, *Proc. Crypto '87*, Lecture Notes in Computer Science, volume 293, Springer-Verlag, 1988. Pages 40–51.
[14] S. Halevi, Efficient commitment with bounded sender and unbounded receiver. In D. Coppersmith, editor, *Proc. Crypto '95*. Lecture Notes in Computer Science, volume 963, Springer-Verlag, 1995. pages 84–96.
[15] M. Naor. Bit commitment using pseudo-randomness. In G. Brassard, editor, *Proc. Crypto '89*, Lecture Notes in Computer Science, volume 435. Springer-Verlag, 1990. pages 128–137.
[16] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for *NP* can be based on general complexity assumptions. In Ernest F. Brickell, editor, *Proc. Crypto '92*, Lecture Notes in Computer Science, volume 740, Springer-Verlag, 1993. pages 196–214.
[17] M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. In *Proc. 21st ACM Symp. on Theory of Computing*, ACM, 1989. pages 33–43.
[18] T.P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In J. Feigenbaum, editor, *Proc. Crypto '91*, Lecture Notes in Computer Science, volume 576, Springer-Verlag, 1992. pages 129–140.
[19] Federal Information Processing Standards, Publication 180. Specifications for a Secure Hash Standard (SHS).

This article was processed using the LaTeX macro package with LLNCS style