On the Existence of Bit Commitment Schemes and Zero-Knowledge Proofs

Ivan Bjerre Damgård[1]

Abstract

It has been proved earlier that the existence of bit commitment schemes (blobs) implies the existence of zero-knowledge proofs of information possession, which are MA-protocols (i.e. the verifier sends only independent random bits) [BrChCr], [GoMiWi].

In this paper we prove the converse result in a slightly modified form: We define a concept called *weakly zero-knowledge*, which is like ordinary zero-knowledge, except that we only require that an *honest* verifier learns nothing from the protocol. We then show that if, using an MA-protocol, P can prove to V in weakly zero-knowledge that he possesses a solution to some hard problem, then this implies the existence of a bit commitment scheme. If the original protocol is (almost) perfect zero-knowledge, then the resulting commitments are secure against an infinitely powerful receiver.

Finally, we also show a similar result for a restricted class of non-MA protocols.

1 Introduction and Related Work.

A bit commitment scheme (blob) is a method that allows protocol participant A to choose a bit b, some random input r and compute from this a *commitment* to b, $BC(b,r)$. To be useful, the bit commitment scheme must satisfy:

- It is hard to predict b from $BC(b,r)$ essentially better than at random.
- A can later *open* the commitment, to convince anybody else about her original choice of b. This is usually done just by revealing r.
- A cannot change her mind about her choice, i.e. she cannot find r, r' such that $BC(1,r) = BC(0,r')$.

This very loose and informal description should be enough to understand the basic ideas in this paper. More formal treatments can be found in [Da] or [BrChCr]. Note also that establishing a commitment may sometimes involve interaction between sender and receiver.

One easy example of a commitment scheme is the case where A is given a large integer n which is the product of 2 prime factors congruent to 3 modulo 4. A can

[1] The author is with Mathematical Institute, Aarhus University, Ny Munkegade, DK 8000 Aarhus C, Denmark.

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 17-27, 1990.

now compute a commitment as $BC(b,r) = r^2 \; mod \; n$, where r is a randomly chosen residue with Jacobi symbol -1^b. In this case, commitments to 1 have exactly the same distribution as commitments to 0, but knowledge of two square roots of a number modulo n with different Jacobi symbols clearly suffices to factor n. Hence, even with infinite computing power, b cannot be found from $BC(b,r)$, but A could cheat if he had computing power enough to factor n before it was "too late", i.e. before the whole protocol is completed.

Commitment schemes are extremely useful in the construction of cryptographic protocols. The general zero-knowledge proof of [GoMiWi] and [BrChCr] as well as the multiparty computation protocols of [ChDaGr] and [GoMiWi2] are based entirely on commitment schemes.

The existence of bit commitments is implied by the existence of one-way functions, as shown by Naor [Na]. The converse, however, is not necessarily true, even if we exclude from the discussion commmitments that are not binary encodable, like quantum blobs [Br] for example: In the present paper, we base commitments on problems for which one can select hard instances at random. This, however, does not necessarily imply that the problem is in any sense "hard on the average". This adds to the interest of studying bit commitments in general and their connection to zero-knowledge protocols.

Another difference to the work of Naor is that our construction has the potential of producing commitments secure against an infinitely powerful receiver (which implies that the sender can cheat if he has enough computing power). The commitments from [Na] has the dual property: the sender cannot cheat at all, while the receiver must be restricted.

The protocols we will be concerned with here are protocols for proving possession of information (rather than proving language membership as in the original zero-knowledge paper [GoMiRa]). In this setup, a prover P (Peggy) possesses a solution to some problem, and tries to convince a verifier V (Vic) that indeed she knows this solution, while giving Vic absolutely no clue as to what the solution is. A little more formally: suppose a relation R on sets U and V, and an element $y \in V$ are given. Then Peggy is trying to convince Vic that she knows how to compute $x \in U$ such that $(x,y) \in R$ holds.

Following [FiFiSh] and [ToWo], we will let both the prover and the verifier be probabilistic polynomial time Turing machines.

In this model, it was proved in [BrChCr] that the existence of a bit commitment scheme implies the existence of a zero-knowledge proof of information possession (as defined in [ToWo]) for any problem in MA, i.e. for any problem, Peggy can prove to Vic in zero-knowledge that she knows a solution, as long as this solution can be verified by a BPP-algorithm. Furthermore, Vic only has to send independently chosen random bits, i.e. the protocol is an MA-protocol.

Our result can be seen a sort of converse to the above: if a (weakly) zero- knowledge proof of knowledge exists for some hard problem, and the protocol has a structure similar to that of [BrChCr], then a bit commitment scheme exists.

In independent work, Fiege and Shamir [FiSh] found a result similar to ours for the special case where the verifier sends only 1 bit per round. This was used to design two-round zero-knowledge proofs of knowledge for any NP-problem. The key observation there was that commitments constructed by our method are always "chameleon", i.e. the bit contained in a commitment can always be changed if extra information is known. One can then simulate parallelized protocols by making this information available to the simulator.

We would like to point out two interesting facts about our result:

- It adds a theoretical basis to the intuitive belief shared by many researchers, that bit commitment schemes are very fundamental objects ideed, and if they do not exist, zero-knowledge proofs of knowledge - at least interesting ones - are probably, to use an expression of Brassard, a fancy way of talking about the empty set [2].

  Put another way: it shows that zero-knowledge proofs of knowledge based on bit commitments are "as invulnerable as possible" against collapse of cryptographic assumptions: if the assumption falls, no interesting objects exist of the kind we are trying to construct.

- It shows that existence of an MA zero-knowledge proof of knowledge for one hard problem is a sufficient condition for the existence of such proofs for anything in MA (by [BrChCr]), even if the problem we start with is not NP-complete.

The restriction to MA-protocols does not seem to be a severe limitation of the result: the most powerful zero-knowledge proof known ([BrChCr], [GoMiWi]) are MA. Furthermore the result is not limited to MA-protocols, as shown by Section 3.

2 Main Result

In this extended abstract, we will only give informal definitions and proofs. For a rigorous definition of zero-knowledge proofs of information possession, and a detailed description of communicating Turing machines, the reader is referred to [ToWo].

We now describe a proof system of information possession for a relation R on sets X and Y: It consists of a pair of communicating probabilistic polynomial time Turing machines (P, V). Common input to P and V is $y \in Y$. P also gets as input $x \in X$. At the end of the conversation, V outputs "accept" or "reject", and the proof system (P, V) is said to accept or reject accordingly. Let m denote the length in bits of y.

We will restrict our attention in this section to the case of MA-protocols, i.e. the prover speaks first, and the verifier sends only uniformly and independently chosen bits.

Thus, we assume that the conversation between P and V has the following form:

- P sends a message m_1.

[2] Zero-knowledge proofs of knowledge do exist for some problems independently of computational assumptions, examples are discrete log, graph isomorphism, and factoring. However if bit commitments did not exist, all these problems would be easy, and their interactive proofs therefore uninteresting.

- V sends bits $b_1, ..., b_k$ to P.
- P sends a message m_2.
- V decides, based on m_1, m_2, $b_1, ..., b_k$ whether to accept or not.

We assume for simplicity that k is constant as a function of m. For generalizations of this, see the remarks after Theorem 1.

We require about (P, V) that:

- For the verifier's protocol V, there exists a *simulator* (a probabilistic polynomial time Turing machine), M_V, which (with the help of V) simulates the conversation between P and V, such that the output of M_V is polynomially (in m) indistinguishable from the real conversation between P and V. Loosely speaking, this means that a polynomial time algorithm cannot guess essentially better than at random whether a given conversation was produced by M_V or by (P, V). Thus we only require that an *honest* verifier learns nothing from the protocol.
- If $(x, y) \in R$, then (P, V) always accepts.
- For any prover's protocol P^*, there exists an *interrogator* (a probabilistic polynomial time Turing machine), M_{P^*}, which (with the help of P^*) tries to compute $x \in X$ such that $(x, y) \in R$. We require that there is a constant $\epsilon < 1$ such that
$$Prob((P^*, V) \ accepts \ AND \ M_{P^*} \ fails) \leq \epsilon,$$
for all sufficiently large m. This is the definition proposed in [ToWo]. A very similar - but technically slightly different - definition appeared in [FiFiSh].

If these 3 conditions are satisfied, (P, V) is called a *weakly zero-knowledge proof system of information possession.*

ϵ is called the *error probability* of (P, V).

We can now state the main result:

Theorem 1

Suppose there exists a binary relation R with the following properties:

- It is easy to select y such that an x with $(x, y) \in R$ exists, but such that computing one is a hard problem. Further, given x and y, it is easy to check, whether (x, y) is in R.
- R admits a weakly zero-knowledge proof system of knowledge with error probability ϵ, where the conversation is of the form described above.

Then there exists a bit commitment scheme with the properties described in Section 1. If the proof system is perfect zero-knowledge, then the commitment scheme constructed is secure against an infinitely powerful receiver, i.e. commitments to 0 have the same distribution as commitments to 1.

Remarks

Note that the proof system would be uninteresting, if the first condition above was not satisfied: it does not make much sense for P to try to keep her solution to the problem secret, if V could just compute a solution himself! Also in this case, the existence of an interrogator becomes trivial: M_{P^*} can just compute the solution by itself, without ever talking to P^*.

The assumption about selectability of y is closely related to the notion of *invulnerable generators*. [AABFH] contains the first study of the theory behind this notion, and the results are improved in [FLM].

As mentioned, we assume that k, the number of challenge bits pr. round, is constant as a function of m. The Theorem could be proved in much the same way if $k = O(log(m))$ and we assume that $\epsilon = O(m^{-1})$. With this extension, the result covers all known MA zero-knowledge proofs of information possession.

Proof

We first describe loosely the basic idea of the proof: We will have two players, A who will create commitments, and B who receives them. The idea is that A will create and send to B the start of a conversation between P and V. To open a commitment, A sends the rest of the conversation. B accepts this, if V would have accepted, based on the conversation given.

If A does not know a solution to the problem instance in question, this implies to some extend a commitment: by the properties of (P, V), A is unable to complete the conversation with respect to *all* possible values of the challenge bits $b_1, ..., b_k$. He is therefore "committed" to the set of values for $b_1, ..., b_k$ for which he *can* complete the conversation. The rest of the proof consists of technical lemmas and tricks that amplify this into a regular commitment scheme.

Let t be the smallest integer, such that $\epsilon^t < 1/2$. By Lemmas 2 and 3 in [ToWo], the proof system that consists of t iterations of (P, V) is weakly zero-knowledge and has error probability ϵ^t. Let (P', V') denote this proof system. We let

$$m_1^i, b_1^i, ..., b_k^i, m_2^i$$

denote the conversation produced by the i'th iteration of (P, V). Let $\mathbf{b} = (b_j^i)_{j=1...k, i=1...t}$, and let $\mathcal{B}$ be the set of all possible values of $\mathbf{b}$.

We can now describe how A can create and open what we will call a *quasi bit commitment* QBC:

1. The receiver of the commitments, B, chooses some $y \in Y$ according to the first condition in Theorem 1, and gives it to A.

2. A commits to a bit b by running the simulator $M_{V'}$ to produce a conversation of the form descibed above. He sends $(m_1^1, ..., m_1^t)$ to B.

3. B partitions $\mathcal{B}$ randomly in two subsets C_0, C_1 such that $|C_0| = |C_1|$, and sends them to A.

4. Let v be chosen such that the value for b produced by $M_{V'}$ above is in C_v. A then sends to B $c = b \oplus v$. This concludes the creation of the commitment, we set $QBC(b, r) = (m_1^1, ..., m_1^t, c)$, where r is the random string consumed by the simulator during the process.

5. To open the commitment, A releases the entire conversation produced in step 2, and B checks that V would have accepted, had it been given this conversation. He then determines v as in step 4, and determines the bit A was committed to as $b = v \oplus c$.

We can now prove the following about quasi bit commitments:

- If A follows the protocol, B cannot predict b from $QBC(b, r)$ essentially better than at random.

- With nonnegligible probability, A is committed after completion of steps 2-4, i.e. A can convince B about at most 1 value for b.

First, suppose by contradiction that B could predict b essentially better than at random based on $QBC(b, r)$. Let p be the success probability of B. Then B could be used to distinguish simulated conversations from real ones as follows: given a conversation $(m_1^i,\ b_1^i, ..., b_k^i,\ m_2^i), i = 1...t$, feed $m_1^1, ..., m_1^t$ to B, and accept from B a partitioning C_0, C_1. Now choose a bit b and feed $c = b \oplus v$ to B, where v is determined from the conversation as in step 4 above. Now output 1 if B's guess mathces b, and 0 otherwise.

On input a real conversation, this produces output 1 with probability $1/2$, since in this case there is no correlation between the m_1^i-values and b. On the other hand, the output is 1 with probability p if the input is simulated. By assumption $p - 1/2$ is non-negligible, contradicting the assumption that M_V is a good simulator.

Clearly, if (P, V) is perfect zero-knowledge, $p = 1/2$, and $QBC(b, r)$ contains no information about b.

Secondly, suppose by contradiction that given y, with non-negligible probability q, A can produce a set of conversations such that all m_1^i-values are constant for a fixed i, and the set of b-values constitute a fraction $\delta > 1/2$ of $\mathcal{B}$. This is what A needs to avoid any chance of committing himself, for if the set of b-values constituted at most half of $\mathcal{B}$, then this set might be contained in C_0 or C_1, which would mean that A would be committed.

We adopt the standard complexity theoretic definition of "non-negligible" and assume that $q \geq 1/Q(m)$ for some polynomial Q and all sufficiently large m.

Consider now the prover P^* with the following strategy: before sending its first message, it runs $mQ(m)$ times whatever method A has for producing sets of conversations as above. If at least one try was successful, it uses the m_1^i produced by this try in the i'th iteration of (P, V). Otherwise, it sends randomly chosen messages. Clearly,

$$Prob((P^*, V')\ accepts) > (1 - (1 - q)^{mQ(m)}) \cdot \delta$$

Since the first factor is exponentially close to 1 for large m, the acceptance probability larger than $1/2$ for all sufficiently large m. Further, since we assume that finding a solution x to $(x,y) \in R$ is a hard problem, and P^* only gets y as input, we may assume that any interrogator M_{P^*} fails to find x with probability essentially 1. Hence,

$$Prob(M_{P^*} \; fails \; AND \; (P^*,V') \; accepts) \simeq Prob((P^*,V') \; accepts) > 1/2,$$

for all sufficiently large m, contradicting the assumption that $\epsilon^t < 1/2$.

Thus we may assume that for each quasi commitment A creates, he will be committed with probability at least

$$p := \frac{(|B|/2)!^2}{|B|!},$$

which is a constant, since k and t are constants.

The rest of the proof is concerned with a protocol construction that uses quasi bit commitments to build ordinary ones:

1. The receiver of the commitment B chooses some $y \in Y$ according to the first condition in Theorem 1, and sends it to A.

2. To commit to a bit b, A creates m quasi commitments to b, $QBC(b,r_1),...,QBC(b,r_m)$.

3. To open the commitment, A reveals b and opens all the quasi commitments. B accepts this if and only if A opens all the quasi commitments correctly.

To prove that this constitutes a bit commitment scheme, we must first argue that B cannot predict b from $QBC(b,r_1),...,QBC(b,r_m)$ essentially better than at random. This follows from the fact that b cannot be guessed from 1 quasi commitment to b. The argument is completely similar to a corresponding one for probabilistic encryptions (see [GoMi]), and we will not repeat it here (of course, the argument becomes trivial if (P,V) is perfect zero-knowledge).

Secondly, we must prove that A is committed with large probability. Since 1 quasi commitment commits A with probability at least p, m quasi commitments will commit A with probability at least $1-(1-p)^m$, which converges exponentially to 1 as a function of m □

Corollary

Suppose the binary relation R satisfies the assumptions of Theorem 1. Then there exists a zero-knowledge proof of information possession for any binary relation S for which $(x,y) \in S$ can be verified efficiently.

Proof

Use R and Theorem 1 to construct a bit-commitment scheme. Represent the verification procedure for S as a (polynomial size) Boolean circuit, and use the protocol from [BrChCr] with the bit-commitment scheme just constructed□

As an example of a protocol satisfying our conditions,·consider the following protocol for proving possession of a discrete log, first found by [ChGr]:

We are given a prime p, a generator g of Z_p^*, and $y \in Z_p^*$. The prover claims to know x, such that $g^x = y \; mod \; p$. She convinces V as follows:

1. P chooses z at random, and sends $c = g^z$ to V.

2. V chooses a bit b, and sends it to P.

3. If $b = 1$, P sends z to V, otherwise he sends $x + z \; mod \; (p-1)$

4. IF $b = 1$, V checks that $c = g^z$, otherwise he checks that $cy = g^{x+z}$.

It is well-known that this constitutes a zero-knowledge proof system of information possession with $\epsilon =$ any constant larger than 1/2.

The resulting bit commitment scheme is the following: A is given y, but not its discrete log. Commitments are computed as follows:

$$BC(1,r) = g^r \; mod \; p, \quad BC(0,r) = yg^r \; mod \; p$$

This commitment scheme is well known, and was introduced in [ChDaGr], and independently in [BoKrKu]. [ToWo] give similar protocols for any random self-reducible relation, which shows that any random self-reducible relation representing a hard problem can be used as basis for a bit-commitment scheme.

3 Non MA-protocols

It is an open question whether our result can be proved for *any* zero-knowledge proof of information possession. But as shown by the following, there is a subclass of non-MA protocols for which our result does hold.

Consider proof systems where the conversation has the following form:

- V does some computation and sends a message m_1 to P.

- P sends a message m_2.

- V decides whether to accept or not

Theorem 2

Suppose the binary relation R satisfies

- It is easy to select y such that an x with $(x, y) \in R$ exists, but such that computing one is a hard problem.

- R admits a weakly zero-knowledge proof system of knowledge.

- The conversation has the form given above, and there is a one-one correspondence between initial messages m_1 and messages m_2 that will make V accept.

- As a function of m, the error probability vanishes faster than any polynomial fraction.

Then there exists a bit commitment scheme as defined in Section 1.

proof

Suppose m_1 and m_2 are in sets M_1 and M_2, respectively. Then the verifier's decision procedure defines a function $f : M_2 \rightarrow M_1$, such that $f(m_2) = m_1$ precisely if V accepts. Since the error probability is negligible, f is hard to invert almost everywhere, but since the protocol is simulatable, it is feasible to produce m_1, m_2 such that $f(m_2) = m_1$ (note that this is not necessarily the same as saying that $f(m_1)$ is easy to compute given m_1).

Using Yao's Xor-Theorem (see for example [Kr]), one can then from f construct a function f' with similar properties, but such that f' has "a hard bit", i.e. given $f'(x)$, there is a particular bit of x which cannot be guessed essentially better than at random by a polynomial time algorithm. From this, it is straightforward to construct a bit commitment scheme□

Remark: We have assumed the 1-1 correspondence between m_1's and m_2's mainly for simplicity. For our purposes, it would actually suffice if f mapped a constant number of elements to 1. Also, we could of course tolerate a non vanishing error probability by iterating the proof system many times. Note also that the recent work by [GoLe] may make it possible to generalize the result even further.

An example of a protocol of this kind: suppose $n = pq$, where p and q are primes congruent to 3 modulo 4. P knows the factorization of n, and so he can compute square roots modulo n. We then do the following:

1. V chooses x at random and sends $x^4 \mod n$ to P

2. P computes and sends to V $c = x^2 \mod n$ - note that although x^4 has 4 square roots modulo n, exactly one of them is itself a square, by the properties of p and q.

3. V accepts, iff $c^2 = x^4 \mod n$.

We leave it to the reader to show that this protocol satisfies all the conditions given in this section, assuming that factoring is hard.

Note that the protocol is weakly zero-knowledge, but not ordinary zero-knowledge: a cheating verifier can factor n after 1 interaction with P by sending a square for which he knows a square root of Jacobi symbol -1.

The function constructed from the protocol is of course $f(x) = x^2 \mod n$, which is a permutation of the quadratic residues modulo n, and is hard to invert, if factoring n is hard. In this case, Yao's Xor Theorem is not necessary to obtain a bit commitment scheme - it is known that guessing the least significant bit of x from $f(x)$ is polynomially equivalent to factoring n [ACGS].

References

[AABFH] Abadi, Allender, Broder, Feigenbaum and Hemachandra: "On Generating Solved Instances of Computational Problems", Proc. of CRYPTO 88, Springer.

[ACGS] Alexi, Chor, Goldreich, Schnorr: "RSA and Rabin Functions: Certain Parts are as Hard as the Whole", Siam J. Compt., vol.17, no.2, 1988, pp.194-209.

[BoKrKu] Boyar, Krentel and Kurtz: "A Discrete Logarithm Implementation of Zero-knowledge Blobs", Tech. Report, Dept. of Computer Science, University of Chicago, 1987.

[Br] Brassard: Modern Cryptology, Lecture Notes in Computer Science, vol.325, Springer-Verlag, 1988.

[BrChCr] Brassard, Chaum, Crépeau: "Minimum Disclosure Proofs of Knowledge", JCSS, vol.37, no.2, Oct. 1988, pp.156-189.

[ChDaGr] Chaum, Damgård, van de Graaf: "Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result", Proc. of Crypto 87.

[ChGr] Chaum, van de Graaf: "An Improved Protocol for Demonstrating possession of a Discrete Log", Proc. of EuroCrypt 87.

[Da] Damgård: "The Application of Claw Free Functions in Cryptography", PhD-Thesis, Aarhus University, Denmark, May 1988.

[FiSh] Fiege and Shamir: "Zero-Knowledge Proofs of Knowledge in Two Rounds", these proceedings.

[FiFiSh] Fiat, Fiege, Shamir: "Zero-Knowledge Proof of Identity", Proc. of STOC 87.

[FLM] Feigenbaum, Lipton and Mahaney: "A Completeness Theorem for Almost-Everywhere Invulnerable Generators", manuscript, AT& T Bell Labs. Tech. Memo, Febr. 89.

[GoLe] Goldreich nd Levin: "A Hard-Core Predicate for all One-Way Functions", Proc. of STOC 89, pp.25-32.

[GoMiRa] Goldwasser, Micali: "Probabilistic Encryption", JCSS, vol 28, no 2, 1984, pp 270-299.

[GoMiRa] Goldwasser, Micali, Rackoff: "The Knowledge Complexity of Interactive Proof Systems", Proc. of STOC 85, pp.291-304.

[GoMiWi] Goldreich, Micali, Wigderson: "Proof that Yield Nothing but the Validity of the Assertion, and the Methodology of Cryptographic Protocol Design", Proc. of FOCS 86.

[GoMiWi2] Goldreich, Micali and Wigderson: "How to Play any Mental Game", Proc. of FOCS 87.

[Kr] Kranakis: Primality and Cryptography, Wiley-Teubner Series in Computer Science, 1986.

[Na] Naor: "Bit Commitment using Pseudo-Randomness", these proceedings.

[ToWo] Tompa, Woll: "Random Self-Reducibility and Zero-Knowledge Proofs of Information Possession", Proc. of FOCS 87.