One-Way Group Actions

Gilles Brassard [*]

Département d'informatique et de R.O.
Université de Montréal
C.P. 6128, succ. "A"
Montréal, Québec CANADA H3C 3J7

Moti Yung

IBM T. J. Watson Research Center
Yorktown Heights
NY 10598, U.S.A.

Abstract

Bit commitment schemes are central to all zero-knowledge protocols [GMR89] for **NP**–complete problems [GMW86, BC86a, BC86b, BCC88, BCY89, FS89, etc.]. One-way group actions is a natural and powerful primitive for the implementation of bit commitment schemes. It is a generalization of the one-way group homomorphism [IY88], which was not powerful enough to capture the bit commitment scheme based on graph isomorphism [BC86b]. It provides a unified theory for all the known bit commitment schemes that offer unconditional protection for the originator of the commitments, and for many of those that offer her statistical protection. (Unconditional protection means that the value of the bit committed to is always perfectly concealed. Statistical protection either means that this is almost always the case, or that only an arbitrarily small probabilistic bias about this bit can leak; in either cases, statistical protection must hold even against unlimited computing power.)

Bit commitment schemes based on one-way group actions automatically have the chameleon property [BCC88] (also called trap-door [FS89]), which is useful for the parallelization of zero-knowledge protocols [BCY89, FS89]. Moreover, these bit commitment schemes allow the originator of two commitments to convince the receiver that they are commitments to the same bit, provided that this is so, without disclosing any information about which bit this is.

In addition, one-way group actions are also a natural primitive for the implementation of claw–free pairs of functions [GMRi88].

1 Bit commitment schemes

Central to all zero-knowledge interactive protocols [GMR89] for **NP**–complete statements [GMW86, BC86a, BC86b, BCC88, BCY89, FS89, etc.] is the notion of *bit commitment scheme*. The purpose of a bit commitment scheme is to allow one party, the *originator*, to commit to the value of a bit in a way that prevents the other party,

[*] Supported in part by Canada NSERC grant A4107.

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 94-107, 1991.

the *receiver*, from learning it without the first party's help, but also in a way that prevents the first party from changing its value. At any time after she[1] has committed to a bit, the originator can show the receiver which bit she had committed to, a process known as *opening* the bit commitment. Of course, she should not be able to cheat by "changing her mind", i.e. not showing the genuine bit that she had in mind when she committed.

Bit commitment schemes can be implemented in a great many ways (see [BCC88] for several examples). When they are based on cryptography and computational complexity (rather than being implemented through physical protection, such as using quantum cryptography [BB84, BC90] or sealed envelopes), they are necessarily imperfect. This imperfection can take place in two very different ways. If it is impossible for the originator to change her commitments in the receiver's back, then it can be at best infeasible (i.e. within a reasonable amount of time) for the receiver to determine the bits committed to without the originator's help. Such bit commitment schemes are called *unconditionally binding*, or *unconditionally secure for the receiver*. Conversely, if it is impossible for the receiver to determine the bits committed to by the originator, or even to obtain partial or probabilistic information about them, then it can be at best infeasible for the originator to cheat and change her mind about a bit she had committed to when (and if) she subsequently decides to open the commitment. Such commitments are called *unconditionally concealing*, or *unconditionally secure for the originator*.

If the impossibility in the above definition is replaced by a near impossibility, the scheme is *statistically* (rather than unconditionally) binding or concealing, whichever the case may be. By "nearly impossible", we mean an event that can occur only with arbitrarily small probability, regardless of the computing power of the parties under consideration. A bit commitment scheme can be statistically concealing in two different ways. It could be that the receiver is expected to learn an arbitrarily small fraction of one bit of information about the bit committed to [IN89], or it could be that he has an arbitrarily small probability of learning the bit with certainty [Cha86, BC86b, NOVY90, etc.]. Statistically concealing bit commitment schemes that fall in the second category usually require a cheating receiver to be *daring* in the sense that he is guaranteed to learn nothing unless he is willing to risk being caught cheating by the originator with near certainty.

All the known perfect zero-knowledge interactive protocols for statements about **NP**–complete problems[2] are based on unconditionally concealing bit commitment schemes [BCC88, BCY89]. Similarly, statistically zero-knowledge interactive protocols can be obtained from statistically concealing bit commitment schemes [Cha86, BC86b]. In this paper, we are concerned exclusively with unconditionally and statistically concealing bit commitment schemes. We also restrict our attention to bit commitment schemes that can be implemented in practice by probabilistic

[1] We shall refer to the originator as a "she" and to the receiver as a "he".

[2] These perfect zero-knowledge protocols are *not* interactive proof–systems in the sense of [GMR89] because they allow a prover with unlimited computing power to cheat by changing her commitments. Hence, they are merely *computationally convincing*.

polynomial-time originator and receiver. In particular, we require of an unconditionally concealing bit commitment scheme that it be impossible for the receiver to learn anything about the bits committed to even if he had unlimited computing power, yet probabilistic polynomial-time computing power must be sufficient in order to carry out the honest protocol. Read [Bra91] for a comprehensive survey of bit commitment schemes, including those that require one of the players to be unreasonably powerful.

As long as we do not consider bit commitment schemes that cannot be implemented in practice, it is known that unconditionally binding bit commitment schemes exist if [ILL89, Nao89, Hås90] and only if [IL89] one-way functions exist. It was recently shown that one-way *permutations* (or even one-way functions with known entropy) are sufficient to implement *statistically* concealing bit commitment schemes [NOVY90]. It would be very nice if *unconditionally* concealing bit commitment schemes could be designed under the mere assumption that one-way *functions* exist. Although we are not able to do this, here we show that it is sufficient to assume the existence of **one-way certified group actions**, which is the main new notion introduced in this paper. This is a generalization of the *one-way group homomorphism* introduced in [IY88] and used in [BCY89] for the purpose of implementing bit commitment schemes. One-way certified group actions are sufficiently general to capture all the known unconditionally concealing bit commitment schemes. Moreover, one-way *un*certified group actions capture many of the statistically concealing bit commitment schemes, but not the schemes of [IN89, NOVY90]. Also, we wish to emphasize that bit commitment schemes based on one-way group actions are truly practical.

Beyond their primary purpose, some bit commitment schemes also offer additional capabilities. For instance, a bit commitment scheme has the *equality property* if, given two different commitments to the same bit, the originator can easily convince the receiver that these commitments are indeed both to the same bit, without disclosing anything about whether this bit is a 0 or a 1 (by "easily", we mean that interaction is not needed; otherwise all bit commitment schemes would have the equality property). The *unequality property* is defined similarly. Bit commitment schemes based on one-way group actions automatically have the equality property, but not necessarily the unequality property.

Another useful additional property that some bit commitment schemes have was first put forward by Brassard, Chaum and Crépeau under the name of "chameleon property" [BCC88]. After reading [BCC88], Shamir proposed to call it the "trap-door property" [FS89], and we agree that this is a more self-explanatory terminology, albeit less poetic. A bit commitment scheme has the *trap-door property* if, in addition to the usual requirements of bit commitment schemes, there exists a secret, known as the *key* to the trap-door, that would allow the originator to cheat her commitments any time she wanted if only she knew this key. More precisely, knowledge of this key would make it possible for her to offer fake "commitments" that she could subsequently "open" either way at her choice of the moment, and these fake commitments are information-theoretically indistinguishable from genuine ones. Bit commitment schemes that have the trap-door property are interesting because they make it possible to implement zero-knowledge interactive protocols in a constant number of rounds [BCY89, FS89]. Clearly, the trap-door property cannot exist for bit commitment schemes that are unconditionally (or even statistically) binding. Bit commitment schemes based on one-way group actions automatically have the trap-door property.

The first unconditionally concealing bit commitment scheme ever proposed is reviewed in the Appendix. It possesses the equality, unequality and trap-door properties. We encourage readers unfamiliar with the notion of bit commitment schemes to read the Appendix before proceeding with the next section.

2 One-way group actions

Let us recall what a *group action* is. Let G be a finite group whose operation is simply denoted by juxtaposition and let ε denote the identity element of G. Let S be any finite set. We say that G *acts* on S if each element of G induces a permutation of S such that the permutation induced by gh is the composition of the permutations induced by g and by h, where g and h are any elements of G. More formally, we have a function $T : G \rightarrow (S \rightarrow S)$ such that

(1) $(\forall g \in G)(\forall h \in G)(\forall s \in S)[(T(gh))(s) = (T(g))((T(h))(s))]$, and

(2) $(\forall g \in G)(\forall t \in S)(\exists! s \in S)[(T(g))(s) = t]$.
It is an easy exercise to show that this unique s is $(T(g^{-1}))(t)$.

Given condition (1), it is elementary to prove that condition (2) is equivalent to saying that the function induced by the group identity element is the identity function. In other words, G act on S through T if and only if conditions (1) above and (3) below are satisfied.

(3) $(\forall s \in S)[(T(\varepsilon))(s) = s]$.

To avoid cluttering the text with parentheses, it is customary (but sometimes confusing —see section 6.2) to denote $(T(g))(s)$ simply by gs. Therefore, condition (1) can be restated simply as $(gh)s = g(hs)$. (Despite the appearance of this formula, it does not really have anything to do with associativity!)

Let us now suppose that G acts on S. Let s_0 be a fixed element of S. The group action is s_0*–one-way* if

- Membership in G and S can be tested efficiently.
- It is feasible to draw randomly within G with uniform distribution (this condition can be relaxed somewhat). By $g \in_R G$, we mean that g is chosen randomly within G with uniform distribution.

- The group operation, the group inversion, and the group action can be computed efficiently. In other words, given any $g \in G$, $h \in G$ and $s \in S$, it is easy to compute gh, g^{-1}, and gs.

- Consider $g \in_R G$ and let $t = gs_0$. Given s_0 and t, it is infeasible to compute any $\hat{g}$ such that $\hat{g}s_0 = t$, except with negligible probability, where the probability is taken over all choices of g and possibly over the random choices taken by the efficient algorithm trying to defeat this property. Note that the problem is *not* to find some $\hat{g} \neq g$, which may or may not exist.

A group action is *one-way* if it is feasible to find an $s_0 \in S$ such that the group action is s_0–one-way. Such an s_0 will be referred to as the *source* of the action. Given any $s \in S$, let $Q(s)$ denote the orbit $\{t \in S \mid (\exists g \in G)[t = gs]\}$. A one-way group action whose source is s_0 is *certified* if

- Given any $t \in S$, it is easy to decide whether or not $t \in Q(s_0)$. Of course, the easy thing is to decide on the *existence* of a $g \in G$ such that $t = gs_0$, not to actually *discover* any such g.

3 Commitments with certified group actions

Let G be a group and S be a set, and consider a one-way certified group action.

- In order to set up a bit commitment scheme, the receiver chooses a source s_0 for the action and a $g_0 \in_R G$. He computes $s_1 = g_0 s_0$ and he gives s_0 and s_1 to the originator. The originator checks that $s_0 \in S$, $s_1 \in S$, and $s_1 \in Q(s_0)$.

- In order to commit to bit $x \in \{0, 1\}$, the originator chooses $g \in_R G$ and computes $b = gs_x$. She keeps g as her witness to the effect that b is a commitment to bit x.

- In order to open a commitment b as bit x, the originator shows the corresponding witness g. The receiver checks that $b = gs_x$.

Commitments to 0 are produced by computing gs_0 whereas commitments to 1 are produced by computing $gs_1 = g(g_0 s_0) = (gg_0)s_0$, where $g \in_R G$. Therefore, such commitments are unconditionally concealing because the effect of computing gg_0 for a fixed $g_0 \in G$ and a g chosen randomly with uniform distribution within G is in fact to choose randomly an element of G with uniform distribution. In other words, nothing distinguishes commitments to 0 from commitments to 1, except for the witness known by the originator alone. The condition that the one-way group action should be certified is *crucial* here: if the receiver were able to get away with giving the originator some $s_1 \notin Q(s_0)$, the set of commitments to 0 would be *disjoint* from the set of commitments to 1.

On the other hand, assume for a contradiction that the originator is able to open a given commitment b both as a 0 and as a 1. In order to do this, she must know x_0 and x_1 in G such that $b = x_i s_i$. But then $x_1^{-1}x_0s_0 = x_1^{-1}x_1s_1 = s_1$. If the originator computes $g = x_1^{-1}x_0$, she will have found a $g \in G$ such that $gs_0 = s_1$, which is precisely what was assumed to be infeasible by the one-wayness of the action.

Bit commitment schemes based on one-way certified group actions automatically have the trap-door and equality properties. The trap-door property is obvious, with g_0 as key. To see the equality property, let b_1 and b_2 be commitments to the same bit, and let g_1 and g_2 be the originator's witnesses for b_1 and b_2, respectively. If the originator provides the receiver with $h = g_2g_1^{-1}$, the receiver can check that $hb_1 = b_2$. We leave it for the reader to verify that the only way the originator can provide an $h \in G$ that transforms b_1 into b_2 is if she can open both commitments to show the same bit (unless she can open neither one of them!). We also leave it for the reader to verify that no additional information about which way these commitments can be opened leaks when the originator gives h to the receiver.

Curiously, bit commitment schemes based on one-way certified group actions do not seem to feature the unequality property in general, whereas the bit commitment scheme based on the more specific certified discrete logarithm assumption does (see the Appendix).

4 Commitments with uncertified group actions

Even though it is crucial that the one-way group action be certified if it is to be used to implement an *unconditionally* concealing bit commitment scheme, this is not necessary if one is satisfied with a *statistically* concealing bit commitment scheme. Therefore, even a one-way group action that is not certified can be used to implement a computationally convincing statistically zero-knowledge interactive protocol for an arbitrary **NP** statement. This is obvious if one is not concerned much about practical considerations. Indeed, the fact that $s_1 \in Q(s_0)$ is an **NP**–statement whose witness g_0 is known of the receiver (where g_0, s_0 and s_1 are as in the first step of the unconditionally concealing commitment described is section 3). Therefore, the receiver can use a general computationally zero-knowledge interactive proof–system [GMW86, BCC88] in order to convince the originator beyond any reasonable doubt that $s_1 \in Q(s_0)$. In this sub-protocol, the receiver will need to originate unconditionally binding bit commitments, which is possible by the work of [ILL89, Nao89, Hås90] since the assumed existence of a one-way group action clearly implies that of a one-way function.

A moment's thought allows us to do much better. The key observation is that the problem of deciding membership in $Q(s_0)$ is necessarily random self-reducible [AL83, AFK89, TW87]. This is sufficient for using the general constant–round *perfect* zero-knowledge interactive proof–system of [BMO90], which allows the receiver (without any need for an unproved assumption) to convince the originator that $s_1 \in Q(s_0)$.

This approach is vastly more efficient for practical purposes than that suggested in the previous paragraph. Moreover, it is carried out in a constant number of rounds, which is crucial if the resulting trap-door statistically concealing bit commitment scheme is to be used as main building block for a constant-round computationally convincing statistically zero-knowledge interactive protocol for an arbitrary **NP** statement [BCY89, FS89].

5 Claw free pairs of functions

One-way group actions can also be used in order to implement *claw free* pairs of functions [GMRi88], which is not surprising because there is an obvious direct connection between such pairs and bit commitment schemes (implicit in [IY88]). Let G be a group and S be a set, and consider a one-way group action. Let s_0 be a source for the action and let g_o be a random element of G. Consider the functions $f_1, f_2 : G \to S$ defined by $f_1(g) = gs_0$ and $f_2(g) = gg_0s_0$. Clearly, finding a "claw" $g_1, g_2 \in G$ such that $f_1(g_1) = f_2(g_2)$ is as hard as finding a $g \in G$ such that $gs_0 = g_0s_0$ (possibly $g = g_0$). Hence, finding such a claw is infeasible for anyone who knows s_0 and g_0s_0 (which is necessary to compute f_1 and f_2), but who does not know g_0. The action should moreover be certified if it is important that the party unable to find a claw should nonetheless be certain that such claws exist.

6 Examples of one-way group actions

6.1 Based on the discrete logarithm [CDG88, BKK90]

Assuming the certified discrete logarithm assumption (see the Appendix), a one-way certified group action can be built as follows. Let p be a prime for which the factorization of $p-1$ is known and let α be a generator for $\mathbf{Z}_p^*$. Let G be $\mathbf{Z}_{p-1}$, let S be $\mathbf{Z}_p^*$, and let s_0 be 1. Given $g \in G$ and $s \in S$, the group action is defined as $gs = s\alpha^g$. It is easy to see that all the requirements for a one-way certified group action are satisfied. In this case, the one-wayness of the group action follows directly from the certified discrete logarithm assumption and the one-way group action is certified because recognizing elements of $Q(s_0)$ is trivial since $S = Q(s_0)$ follows from the fact that α is a certified generator.

6.2 Based on factoring [BC86b]

Assuming that factoring large RSA integers is infeasible, a one-way group action can be built as follows. Let n be the product of two large distinct primes (in practice, n would be chosen by the receiver, unless an authority can be trusted for choosing n and never disclosing its factors). Let both G and S be $\mathbf{Z}_n^*$, the multiplicative group of invertible integers modulo n. Let s_0 be 1. Given any $g \in G$ and $s \in S$, the group action is defined as $gs = g^2 s$ (please do not allow this mixed notation to confuse you... the gs on the left of the equation means "the result of group element g acting on set element s", whereas the $g^2 s$ on the right refers to one squaring and one multiplication in $\mathbf{Z}_n^*$). Once again, it is easy to see that all the requirements for a one-way group action are satisfied. The one-wayness of the action comes from the fact that extracting square roots modulo n is as hard as factoring n, which we assumed to be infeasible.

Although this group action is probably not certified, it has a significant advantage in practice over the group action based on the discrete logarithm: it is much more efficient to compute two modular multiplications than one exponentiation.

6.3 Based on group homomorphism [BCY89]

Consider any one-way group homomorphism $h : X \rightarrow Y$ (see ICALP version of [BCY89] for a definition) such that membership in X and Y can be tested efficiently (an important condition forgotten in [BCY89]). Let G be X, S be Y, and s_0 be the identity element of Y. Given $g \in G$ and $s \in S$, the group action is defined as $gs = s * h(g)$. Details that this defines a one-way certified group action are left for the reader.

6.4 Based on graph isomorphism [BC86b]

The notion of one-way group homomorphism described in [BCY89] provides a generalization of the bit commitment scheme based on the certified discrete logarithm assumption, but it is probably not as general as one-way certified group actions. Indeed, we now describe a one-way group action that does not correspond to a one-way group homomorphism. The one-wayness of our group action depends on a cryptographic assumption introduced in [BC86b]. Unfortunately, this group action does not seem to be certified.

Let n be a fixed large integer. Let G be the group of permutations of $X_n = \{1, 2, \ldots, n\}$ under composition (where $(gh)(i) = h(g(i))$). Let S be the set of all graphs with X_n as vertex set. Let us assume the existence of a *hard graph* in the sense of [BC86b]: a graph is hard if it is infeasible to figure out an isomorphism between it and a random isomorphic copy of it, except with negligible probability. Let $s_0 \in S$ be such a hard graph. Given $g \in G$ and $s = (X_n, E) \in S$, the group action is defined as $gs = (X_n, \hat{E})$, where $(u, v) \in \hat{E}$ if and only if $(g(u), g(v)) \in E$. This group action is one-way by assumption. Intuitively, the reason why it cannot be recast as a group homomorphism is that there is no natural group operation that one could put on S.

7 Unifying perfect zero-knowledge proof–systems

When the notion of zero-knowledge proof-systems was presented and examples of languages were given, an intriguing fact was observed by many. That is, the proof-systems used to prove that membership in various languages can be proved in perfect zero-knowledge are very "similar". These languages are in two groups, which may be called "positive examples" and "negative examples". For instance, some positive examples are quadratic residuosity [GMR89], graph isomorphism [GMW86] and proving an element to be a generator, i.e., primitivity [TW87]. The corresponding negative examples (in the same works, respectively) are: quadratic non-residuosity, graph non-isomorphism and not-generated (by a), i.e., the language $\mathbf{Z}_p^* - < a >_p$.

Viewing all these problems as group actions[3], we can show that indeed the perfect zero-knowledge proof–systems are instances of two basic protocol schemes; one scheme for the positive examples and another scheme for the negative ones. This shows that the similarity of the original protocols is not just a matter of coincidence. This observation is not new for the positive examples: Burmester, Desmedt, Piper and Walker have developed an algebraic framework that unifies a large number of zero-knowledge proof systems (in which the use of group actions is but a special case) [BDPW89]. A similar observation was also made by Blum and Kannan with respect to program checkers [BK89]. In this section, we review the use of group actions to unify the positive examples, and we extend it to the negative examples.

The protocol scheme which gives as instances the positive examples demonstrates that an element (or a set of elements) can be in the range of a group action acting on a given source. Showing an element to be a quadratic residue is showing that the group of residues acting on the unity ($s = 1$) includes that element, namely that the group of quadratic residues acting on the input element gives us back the group of quadratic residues. The case of graph isomorphism is showing that the group of permutation acting on the first input graph includes the second input graph (or its automorphism group). Finally, showing primitivity is demonstrating that the claimed generator's group $< g >_p$ acting on 1 gives the entire $\mathbf{Z}_p^*$. These positive examples have been previously characterized as "random self-reducible" problems [AL83, AFK89, TW87]. Independently from our work, the notion of group action has been used by several other researchers to implement random self-reducibility [FKN90, SI90].

Using group action, one can show that the proof–systems for the so-called negative examples are instances of a protocol scheme which demonstrates that one input element from a set S_1 cannot be in the range of the group action by a group G acting on an element from a set S_2 as a source (where actions by G on both sets are defined and the underlying assumption is that telling apart the two sets S_1 and S_2 is hard). Notice that there is an information-theoretic difference between these two inputs: two non-isomorphic graphs, a quadratic non-residue (which is different from the quadratic

[3] This section deals with group actions in general, not necessarily those that are one-way, but the problem of membership in the set(s) they act upon must be assumed to be hard for the corresponding protocols to be interesting.

residue $s = 1$ modulo n), or an element b not generated by another element a in $\mathbf{Z}_p^*$ (which means that b is not an element in the range of the group generated by a acting on $s = 1$). This difference can be detected by the powerful prover, a fact used in the various instances of this proof–system.

In summary, the group action abstraction helps in generalizing many of the known protocols for perfect zero-knowledge proof–systems.

Open questions

How general is the notion of one-way group action? Andy Klapper has suggested that this notion might extend to that of one-way *monoid* action. Is there any unconditionally concealing bit commitment scheme that could be obtained from a one-way monoid action but not a one-way group action? Better yet, can one design an unconditionally concealing bit commitment scheme under the only assumption that one-way functions exist? If not, what about a statistically concealing bit commitment scheme? Recall that one-way permutations or even one-way functions with known entropy are sufficient to build a statistically concealing bit commitment scheme [NOVY90]. Note that if one does not insist that the scheme be usable by probabilistic polynomial-time players, it is known that one-way functions are sufficient to implement statistically (but not unconditionally) concealing bit commitment schemes [OVY90].

Conversely, is it possible to design a one-way certified group action (or perhaps monoid action) under the sole assumption that unconditionally concealing bit commitment schemes exist? Notice that a positive answer to this question would imply that all unconditionally concealing bit commitment schemes can be made to have the trapdoor and the equality properties, which would be surprising since this does not seem to be the case for statistically concealing bit commitment schemes [IN89, NOVY90].

Finally, are one-way *certified* group actions really more general than one-way group homomorphisms? Recall that our only example of a one-way group action that did not correspond to a one-way group homomorphism (section 6.4) was probably not certified.

Appendix:
Concrete example of a bit commitment scheme

The first unconditionally concealing bit commitment scheme ever proposed was designed independently by Damgård [CDG88] and by Boyar, Krentel and Kurtz [BKK90]. We describe it here to provide a concrete example of the type of bit commitment scheme that can be obtained by one-way certified group actions (see Section 6.1).

Let us first review some elementary number theory [Kra86]. If p is a prime number, let us denote by $\mathbf{Z}_p^*$ the multiplicative group of non-zero integers modulo p, i.e. $\{1, 2, \ldots, p-1\}$ under multiplication modulo p. Similarly, let us denote by $\mathbf{Z}_{p-1}$ the additive group of integers modulo $p-1$, i.e. $\{0, 1, \ldots, p-2\}$ under addition modulo $p-1$. Notice that $\mathbf{Z}_p^*$ and $\mathbf{Z}_{p-1}$ contain the same number of elements. For any integers a, b and c such that $a \not\equiv 0 \pmod{p}$ and $b \equiv c \pmod{p-1}$, we have by Fermat's theorem that $a^b \equiv a^c \pmod{p}$. Therefore, it makes sense to speak of x^i for $x \in \mathbf{Z}_p^*$ and $i \in \mathbf{Z}_{p-1}$. An element α of $\mathbf{Z}_p^*$ is called a *generator* of $\mathbf{Z}_p^*$ if each element of $\mathbf{Z}_p^*$ can be obtained as a power of α. Thus, α is a generator if and only if the function $exp_\alpha : \mathbf{Z}_{p-1} \to \mathbf{Z}_p^*$ defined by $exp_\alpha(i) = \alpha^i$ establishes a one-one correspondence.

In order to set up the bit commitment scheme, the originator and receiver initially agree on a large prime p for which they both know the factorization of $p-1$ (finding large primes p with known factorization of $p-1$ can be done efficiently in practice [Mau89]). They also agree on a generator α of $\mathbf{Z}_p^*$. Thanks to their knowledge of the factors of $p-1$, they can both verify with certainty that p is a prime and that α is a generator of $\mathbf{Z}_p^*$. Moreover, the density of generators is high enough that one can be found reasonably efficiently by random trial and error. Actually, the parameters p and α need not be changed each time a bit commitment scheme has to be set up. Rather, they could be in the public domain (together with the factorization of $p-1$) after having been selected once and for all by an authority that does not need to be trusted. Given any $i \in \mathbf{Z}_{p-1}$, it is easy to compute α^i efficiently by a divide-and-conquer approach, but no efficient algorithm is known to invert this process (even if the factors of $p-1$ are known, provided they are not too small [PH78]), an operation known as *extracting the discrete logarithm.*

Once the parameters p and α have been agreed upon, the receiver chooses a random $s \in \mathbf{Z}_p^*$ and gives it to the originator. We assume the *certified discrete logarithm assumption*, namely that the originator is not capable of computing the discrete logarithm of s while the protocol is in progress ("certified" because the factors of $p-1$ are known to all parties in order that α be a certified generator — since this could make computing the discrete logarithm easier, this assumption is stronger than the usual discrete logarithm assumption of Blum and Micali [BM84]; nevertheless, the usual assumption is good enough to implement a bit commitment scheme statistically secure for the originator).

In order to commit to bit $x \in \{0, 1\}$, the originator selects a random $r \in \mathbf{Z}_{p-1}$ and she computes $b = \alpha^r s^x$. She gives b to the receiver but she keeps r as her secret *witness.* Subsequently, if the originator wishes to convince the receiver that b was a commitment to bit x, she simply shows him the corresponding witness r. The receiver can then check that indeed $b = \alpha^r s^x$.

Because the function exp_α is a one-one correspondence, any element of $\mathbf{Z}_p^*$ can be used by the originator as commitment to 0 just as well as to 1, depending only on which witness she knows. Moreover, all commitments are randomly and independently distributed according to the uniform distribution over $\mathbf{Z}_p^*$. Therefore, it is information-theoretically impossible for the receiver to distinguish a commitment to 0 from a commitment to 1, regardless of his computing power. On the other hand, the originator is able to cheat and open a given commitment both ways if and only if she knows (or can efficiently compute) the discrete logarithm of s, which we assumed to be infeasible for her.

A moment's thought suffices to see that this bit commitment scheme is trap-door, and that its key is the discrete logarithm of s. Moreover, it is easy to see that this bit commitment scheme has both the equality and the unequality properties.

Acknowledgements

It is a pleasure to acknowledge fruitful discussions with Claude Crépeau, Yvo Desmedt, Russel Impagliazzo, Andy Klapper, Moni Naor, Rafail Ostrovsky, and Hiroki Shizuya. The terms "concealing" and "computationally convincing" are due to David Chaum. We also acknowledge the referees for their suggestions.

References

[AFK89] Abadi, M., J. Feigenbaum, and J. Kilian, "On hiding information from an oracle", *Journal of Computer and System Sciences*, Vol. 39, 1989, pp. 21 - 50.

[AL83] Angluin, D. and D. Lichtenstein, "Provable security of cryptosystems: A survey", Technical Report YALEU/DCS/TR-288, Department of Computer Science, Yale University, 1983.

[BMO90] Bellare, M., S. Micali, and R. Ostrovsky, "Perfect zero-knowledge in constant rounds", *Proceedings of the 22nd ACM Symposium on Theory of Computing*, 1990, pp. 482-493.

[BB84] Bennett, C.H. and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing", *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, Bangalore, India, December 1984, pp. 175 - 179.

[BK89] Blum, M. and S. Kannan, "Designing programs that check their work", *Proceedings of the 21st ACM Symposium on Theory of Computing*, 1989, pp. 86 - 97.

[BM84] Blum, M. and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits", *SIAM Journal on Computing*, Vol. 13, 1984, pp. 850 - 864.

[BKK90] Boyar, J. F., M. W. Krentel, and S. A. Kurtz, "A discrete logarithm implementation of zero-knowledge blobs", *Journal of Cryptology*, Vol. 2, no. 2, 1990.

[Bra91] Brassard, G., "Cryptology column — Bit commitment schemes", *Sigact News*, in preparation, 1991.

[BCC88] Brassard, G., D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge", *Journal of Computer and System Sciences*, Vol. 37, no. 2, 1988, pp. 156 - 189.

[BC86a] Brassard, G. and C. Crépeau, "Zero-knowledge simulation of Boolean circuits", *Advances in Cryptology: CRYPTO '86 Proceedings*, Springer–Verlag, 1987, pp. 224 – 233.

[BC86b] Brassard, G. and C. Crépeau, "Non-transitive transfer of confidence: A *perfect* zero-knowledge interactive protocol for SAT and beyond", *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, 1986, pp. 188 – 195.

[BC90] Brassard, G. and C. Crépeau, "Quantum bit commitment and coin tossing protocols", *Advances in Cryptology:* these *CRYPTO '90 Proceedings*, Springer–Verlag.

[BCY89] Brassard, G., C. Crépeau, and M. Yung, "Everything in **NP** can be argued in *perfect* zero-knowledge in a *bounded* number of rounds", *Proceedings of the 16th International Colloquium on Automata, Languages and Programming*, Springer–Verlag, 1989, pp. 123 – 136. Final paper to appear in *Theoretical Computer Science* under the title of "Constant-round perfect zero-knowledge computationally convincing protocols".

[BDPW89] Burmester, M. V. D., Y. G. Desmedt, F. Piper, and M. Walker, "A meta zero-knowledge scheme", *Proceedings of CO89 Combinatorial Optimization Conference*, University of Leeds, July 1989. Submitted for journal publication.

[Cha86] Chaum, D., "Demonstrating that a public predicate can be satisfied without revealing any information about how", *Advances in Cryptology: CRYPTO '86 Proceedings*, Springer–Verlag, 1987, pp. 195 – 199.

[CDG88] Chaum, D., I. B. Damgård, and J. van de Graaf, "Multiparty computations ensuring privacy of each party's input and correctness of the result", *Advances in Cryptology: CRYPTO '87 Proceedings*, Springer–Verlag, 1988, pp. 87 – 119.

[FS89] Feige, U. and A. Shamir, "Zero knowledge proofs of knowledge in two rounds", *Advances in Cryptology: CRYPTO '89 Proceedings*, Springer–Verlag, 1990, pp. 526– 544.

[FKN90] Feigenbaum, J., S. Kannan, and N. Nisan, "Lower bounds on random self-reducibility", *Proceedings of the 5th IEEE Structure in Complexity Theory Conference*, 1990.

[GMW86] Goldreich, O., S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design", *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, 1986, pp. 174 – 187.

[GMR89] Goldwasser, S., S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems", *SIAM Journal on Computing*, Vol. 18, no. 1, 1989, pp. 186 – 208.

[GMRi88] Goldwasser, S., S. Micali, and R. Rivest, "A secure digital signature scheme", *SIAM Journal on Computing*, Vol. 17, no. 2, 1988, pp. 281 – 308.

[Hås90] Håstad, J., "Pseudo-random generators under uniform assumptions", *Proceedings of the 22nd ACM Symposium on Theory of Computing*, 1990, pp. 395 – 404.

[ILL89] Impagliazzo, R., L. Levin, and M. Luby, "Pseudo-random generation from one-way functions", *Proceedings of the 21st ACM Symposium on Theory of Computing*, 1989, pp. 12–24.

[IL89] Impagliazzo, R. and M. Luby, "One-way functions are essential for complexity based cryptography", *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, 1989, pp. 230–235.

[IN89] Impagliazzo, R. and M. Naor, "Efficient cryptographic schemes provably as secure as subset sum", *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, 1989, pp. 236–241.

[IY88] Impagliazzo, R. and M. Yung, "Direct minimum-knowledge computations", *Advances in Cryptology: CRYPTO '87 Proceedings*, Springer–Verlag, 1988, pp. 40–51.

[Kra86] Kranakis, E., *Primality and Cryptography*, Wiley–Teubner Series in Computer Science, 1986.

[Mau89] Maurer, U. M., "Fast generation of secure RSA–moduli with almost maximal diversity", *Advances in Cryptology: EUROCRYPT '89 Proceedings*, Springer–Verlag, to appear.

[Nao89] Naor, M., "Bit commitment using pseudo-randomness", *Advances in Cryptology: CRYPTO '89 Proceedings*, Springer–Verlag, 1990, pp. 128–136.

[NOVY90] Naor, M., R. Ostrovsky, R. Venkatesan, and M. Yung, manuscript, 1990.

[OVY90] Ostrovsky, R., R. Venkatesan, and M. Yung, manuscript, 1990.

[PH78] Pohlig, S. and M.E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Transactions on Information Theory*, Vol. IT–24, 1978, pp. 106–110.

[SI90] Shizuya, H. and T. Itoh, "A group-theoretic interface to random self-reducibility", *Transactions of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan*, Section E, Vol. E73, no. 7, July 25, 1990.

[TW87] Tompa, M. and H. Woll, "Random self-reducibility and zero-knowledge proofs of possession of knowledge", *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, 1987, pp. 472–482.