An Integer Commitment Scheme based on Groups with Hidden Order

(Preliminary Version)

Ivan Damgård and Eiichiro Fujisaki

BRICS, Dept. of Computer Science, Aarhus University and NTT Labs

Abstract. We present a commitment scheme allowing commitment to arbitrary size integers, based on any Abelian group with certain properties, most importantly that it is hard for the committer to compute its order. Potential examples include RSA and class groups. We also give efficient zero-knowledge protocols for proving knowledge of the contents of a commitment and for verifying multiplicative relations over the integers on committed values. This means that our scheme can support, for instance, the efficent interval proofs of Boudot[1]. The scheme can be seen as a modification and a generalization of an earlier scheme of Fujisaki and Okamoto [5], and in particular our results show that we can use a much larger class of RSA moduli than the safe prime products proposed in [5]. Also, we correct some mistakes in the proofs of [5] and give what appears to be the first multiplication protocol for a Fujisaki/Okamoto-like scheme with a complete proof of soundness.

1 Introduction

The notion of commitment is at the heart of almost all non-trivial cryptographic protocols. The basic functionality one wants from a commitment is that the committer may choose in private a secret s from some set S and release some information, *the commitment* to a verifier, such that: even though the scheme is *hiding*, i.e., the verifier cannot compute s from the commitment, it is also *binding*, i.e., the committer cannot change his mind after having committed, but he can later open the commitment to reveal s, and convince the verifier that this was indeed the original value committed to.

In many applications, one wants extra functionality from a commitment scheme, for instance that the committer can prove in zero-knowledge that he knows how to open a given commitment, in particular that he knows the value committed to. Also, if S has an algebraic structure, say as a ring or a group, it can be very useful to have a *multiplication protocol*, i.e., a zero-knowledge protocol in which the committer can prove that committed values a, b, c satisfy $ab = c$. If S is a ring, one can often in addition achieve that that from commitments to $a, b \in S$, the verifier can compute a commitment to $a + b$ without interacting with the committer.

One example of such a scheme where $S = Z_q$, where q is a prime, is the scheme of Pedersen [6], for the associated protocols and additional examples, see

[2]. In the wast majority of examples known, the set S is Z_m for some m, where m may or may not be a prime. A multiplication protocol for such a scheme can show that for committed numbers a, b, c, it holds that $ab = c \bmod m$. However, there are several important cases where what you need is to be able to prove that $ab = c$ holds *over the integers*. One example of this is if you want to show that a committed number s is an RSA signature on a given message a w.r.t. public key $n, 3$. What we want to know is that $a = s^3 + tn$ for some t, and this of course must be true over the integers and not just modulo m. Of course, one might be able to solve this by choosing the commitment scheme such that $m = n$, but this requires at least that you know n at the time the commitment scheme is set up, and also a new instance of commitment scheme for each n. This is often unreasonable in practice. There are other ways around the problem, see for instance [3], but the protocols are far from optimal, typically one has to resort to "binary cut-and-choose", which means communication complexity at least quadratic in the security parameter. Another example of the need for relations over the integers is the efficient zero-knowledge proofs of Boudot[1] for demonstrating that a committed number is in a given interval. Here, it is crucial for efficiency that one can prove efficiently that committed numbers a, b satisfy $b = a^2$ over the integers.

It should be clear that what we really need here is an *integer* commitment scheme, that is, a scheme where $S = Z$ (or at least some large finite interval), and where there is an efficient multiplication protocol that works over the integers. Here, by efficient, we mean constant round protocols requiring only communication linear in the security parameter. One possible approach is to build on any of the schemes we discussed earlier: any such scheme can be used to commit to a bit, a 0/1 value. We can then commit to an integer a by committing to each bit of a individually. But since each commitment is usually a large multiprecision number, this approach is only efficient when the numbers committed to are very small.

In [5], Fujisaki and Okamoto present the first efficient integer commitment scheme, and also suggested an efficient multiplication protocol. The scheme is based on the strong RSA assumption and is still the most efficient example known. Unfortunately, as pointed out by Markus Michels in private communication to Fujisaki, the proof of soundness of the associated protocols was not complete. Thus, until recently, it has been open whether efficient and secure multiplication protocols exist for commitment schemes of the type given in [5]. Later in the paper we give a short explanantion of the problem in the proof from [5].

In this paper, we present a commitment scheme that may be seen as a generalization of the Fujisaki-Okamoto scheme. We start from an arbitrary Abelian group G, with some basic properties. We assume that the verifier can choose the group and publish a *description* of it that allows anyone to compute the group and inversion operations in G. For the RSA case, this amounts to publishing the modulus n. The most important extra property we need is that it it hard, given the description, to extract roots of a given random element in G. This is just the natural generalization of the strong RSA assumption. Some extra technical conditions are needed as well, we detail those later. We then build from this an integer commitment scheme, as well as a zero-knowledge protocol for proving knowledge of how to open a commitment, and an efficient zero-knowledge multiplication protocol.

If we specialize to the case where $G = Z_n^*$ for an RSA modulus n, we obtain - modulo some technical changes - the commitment scheme of Fujisaki and Okamoto, in particular we get what appears to be the first secure multiplication protocol for this type of scheme. In addition, the conditions we need on G turn out to translate into conditions on n that are much milder than those needed originally by Fujisaki and Okamoto, namely that $n = pq$ is a safe prime product. We only need that $p = q = 3 \bmod 4$, that $gcd(p-1, q-1) = 2$ and that $p-1, q-1$ do not have too many small prime factors (a precise description follows below). Finally, our construction is applicable to other groups than RSA, for instance class groups. Here, it should be noted that finding roots in a class group seems to require finding the order of the group, and this problem is known to be at least as hard as factoring, and may in fact be harder.

We do not know if one can give correct proof for exactly the commitment scheme and protocols suggested in [5] - as mentioned our commitment scheme and protocols are not exactly the same as those of [5], even when specialized to $G = Z_n^*$. However, since our protocols are as efficient as those of [5], this question seems to be of minor importance.

2 Model

Suppose we are given a way to construct a group G of unknown order, such as an RSA group or a class group. More precisely, we have an probabilistic polynomial time algorithm $\mathcal{G}$ which on input 1^k outputs a description $descr(G)$ of a group G. The algorithm may also output some side information, such as the order of G, or the prime factorization of the order; it may even be possible to ensure that the order of the group satisfies certain conditions. This can be the case with RSA, but not with class groups, given our current knowledge. We assume that the description includes also a positive integer C. The role of C is that the protocols to follow will be designed to have error probability $1/C$, up to a negligible additive term. This will follow because C is assumed to satisfy certain conditions w.r.t. G, as detailed below. As a function of the security parameter C will be assumed to be superpolynomially large, in the examples we know of, however, it is typically much smaller than the group order.

Given $descr(G)$, we assume that one can compute efficiently some estimates on the order, $2^A \leq ord(G) \leq 2^B$, where A and B are polynomial in k. We also assume that elements can be sampled randomly from the group and that inversion and group operation can be computed efficently. As usual, a probability $\epsilon(k)$ will be called negligible if for all polynomials $f()$, we have $\epsilon(k) \leq 1/f(k)$ for all large enough k.

We make the following assumptions about groups output by $\mathcal{G}$:

Strong root assumption Let A be any probabilistic polynomial time algorithm. We run $(\mathcal{G})$ on input 1^k to get $descr(G)$. We give $descr(G)$ and a random $h \in G$ as input to A. Suppose A outputs $y \in G$ and a number t. We require that the probablity that $t > 1$ such that $y^t = h$ is negligible.

Small order assumption Let A be any probabilistic polynomial time algorithm. We run $(\mathcal{G})$ on input 1^k to get $descr(G)$. We give $descr(G)$ as input to A. Suppose A outputs $b \in G$ and a number σ. We require that the probablity that $b \neq 1$, $0 < \sigma < C$, $b^\sigma = 1$ and $b^2 \neq 1$ is negligible.

No high 2-powers in orders Any element of form a^{2t} has odd order.

Many elements with only large prime factors in orders If h is chosen randomly in G, then there is a significant probability that the order of h has no prime factors less than C. We say that *$ord(h)$ is C-rough* (as opposed to being C-smooth, which means the order has only prime factors less than C).

Some comments on the assumptions: The first assumption is a direct generalization of the strong RSA assumption. The second one says that elements of relatively small known order should hard to find, except possibly for order 2. This is to take account of the fact that in the RSA case, -1 always has order 2. The third assumption is always true if $ord(G)$ is odd, and otherwise we need that elements of order 2 are the only elements of order a 2-power. Finally, the fourth assumption basically is a condition on the prime factorization of $ord(G)$: if we write $ord(G) = FD$, where F has only prime factors less than C and D has only prime factors greater than C, then the assumption is satisfied iff F is at most polynomial in the security parameter.

To justify the assumptions, we show that RSA moduli can be constructed such that the assumptions are satisfied, based only on the strong RSA assumption. Suppose we make a k-bit modulus $n = pq$ such that $p = q = 3 \bmod 4$, and that $gcd(p-1, q-1) = 2$. We choose C as a function of k in such a way that numbers less than C are feasible to factor. With the subexponential factoring algorithms currently known, one may choose C to be superpolynomial in k, $C = O(k^{\log k})$ is one possibility. We construct p, q such that the parts of $p-1, q-1$ with prime factors less than C are $O(k)$. We then set $G = Z_n^*$. and $descr(G) = n, C$. Now, the strong root assumption is simply the strong RSA assumption. Finding a non-trivial pair b, σ with $b^\sigma = 1, b^2 \neq 1$ is as hard as factoring n: given such a pair, we can factor σ and so we can find an element $\tilde{b}$ of known prime order $s \neq 2$. Now, s cannot divide both $p-1$ and $q-1$ and therefore $\tilde{b}$ must be congruent to 1 modulo one of p, q and different from 1 modulo the other. It follows that $gcd(b-1, n)$ is a non-trivial factor of n. The assumption on no large 2-powers in orders follows directly from $p = q = 3 \bmod 4$, since then 2 divides $p-1$ and $q-1$ only once. Finally the construction of p, q implies that a random element in Z_n^* has a C-rough order with probability that is $\Omega(1/k)$. We may even choose a larger C (and hence get smaller error probability for the protocols), but then the small order assumption must be made as a separate intractability assumption.

Note that a special case of this construction of n is when $n = pq$ is a safe prime product, i.e., $(p-1)/2, (q-1)/2$ are primes, but evidently the construction covers a much larger class of moduli.

3 The Commitment Scheme

Based on the above model, the goal is to make a commitment scheme with protocols to verify various claims on committed values. The basic scheme is that the verifier V (the receiver of commitments) will run $\mathcal{G}$ and send $dscr(G)$ (and more information to be described later) to the prover P (the committer). We assume that P can verify easily that $dscr(G)$ actually describes a group.

Note that for the following version of our commitment scheme, we need more than what we assumed in the general set of conditions above: It must be possible for the party who chooses G to efficiently select an element h such that it is *guaranteed* to have C-rough order. This is possible if the $\mathcal{G}$ outputs the factorization of the order of G. This can be assumed without loss of generality in the RSA case. We later look at ways to do without this assumption.

Set-up V runs $\mathcal{G}$ and chooses a random element $h \in G$, such that $ord(h)$ is C-rough. Now V sets $g = h^\alpha$, where α is randomly chosen in $[0..2^{B+k}]$. V sends $descr(G), g, h$ to P and proves that $g \in < h >$, by the standard zero-knowledge discrete log protocol with binary challenges: in one iteration of this, V sends $a = h^R$ for a random $R \in [0..2^{B+2k}]$. P selects a random bit b, and V replies with $z = R + b\alpha$. P checks that $h^z = ag^b$. Repeating this k times results in a soundness error of 2^{-k}, and the protocol is easily seen to be statistical zero-knowledge. This is not a very efficient solution, but it only needs to be done once and for all in the set-up phase.

Commit To commit to an integer x, P chooses r at random in $[0..2^{B+k}]$, and sends $c = g^x h^r$ to V.

Open To open a commitment, P must send x, r, b such that $c = g^x h^r b$, $b^2 = 1$. An honest prover can always use $b = 1$. The reason for giving a dishonest prover this extra freedom will become clear later.

As for hiding, note that P verifies initially that $g \in < h >$. Hence, since r is chosen with bit length at least twice that of the order of h, c is statistically close to uniform in $< h >$, for any value of x.

As for binding, we show that a polynomial time prover P^* can create c, and $(x, r, b), (x', r', b')$, valid openings with $x \neq x'$, with only negligible probability.

For contradiction, asssume that some P^* could do this. Then we get $g^x h^r b = c = g^{x'} h^{r'} b'$. Recall that V creates g as $g = h^\alpha$. Plugging this in and squaring both sides of the equation, we get that $h^{2(\alpha(x-x')+r-r')} = 1$. Since α is chosen to be much larger than the order of h, P^* does not have full information on α: if we write $\alpha = q \cdot ord(h) + res$ for integers q, res with $0 \leq res < ord(h)$, then from P^*'s point of view, res is uniquely determined from g, whereas there is almost no information on q (the only source of information is the proof that $g \in < h >$ which is statistical zero-knowledge). This and $x - x' \neq 0$ means that, except with negligible probability, we have $M := (\alpha(x - x') + r - r') \neq 0$. This follows because if we fix res (and hence g), the resulting distribution of x, x', r, r' is (almost) independent of q, and every fixed choice of res, x, x', r, r', there is a at most 1 value of q that will imply $M = 0$. However, there are exponentially many possibilities for q. If indeed M is non-zero, it is a multiple of the order of h.

It now follows that if, as we assumed, P^* could break the binding property with non-negligible probability, V and P^* together could solve the strong root problem on input h: V will use the given h in the set-up phase instead of choosing one itself. With non-negligible probability, h will have C-rough order. Given that this happens, there is a non-negligible probability that P^* will break the binding as described above, and this allows us to compute M, a multiple of the order of h. Now choose any t that is relatively prime to M and output $h^{t^{-1} \bmod M}, t$.

4 Auxiliary Protocols

4.1 Proving you know how to open

The following protocol can be used by P to show that he can open a given commitment $c = g^x h^r$.

We will be assuming that the numeric value of x is at most T, where T is a public constant. T can be chosen arbitrarily large, and is only used to control the size of the prover's random choices, to ensure that the protocol hides the value of x, whenever $-T \leq x \leq T$. In any application of the scheme, one simply chooses T large enough to acomodate any choice of x the prover could possibly make in the given scenario. Note that the protocol is *not* designed to guarantee the verifier that $-T \leq x \leq T$. To prove x is in some interval, other techniques exist, see e.g. [1].

1. P chooses $y \in [0..TC2^k[, s \in [0..C2^{B+2k}[$ at random and sends $d = g^y h^s$ to V.
2. V chooses at random $e \in [0..C[$ and sends to P.
3. P sends $u = y + ex, v = s + er$. V checks that $g^u h^v = dc^e$

Completeness of this protocol is clear. It is honest verifier statistical zero-knowledge, to simulate we can choose at random $u \in [0..TC2^k[$, $v \in [0..C2^{B+2k}[$, $e \in [0..C[$ and set $d = g^u h^v$. There are then a number of known techniques by which a zero-knowledge protocol can be constructed from it. We sketch one way to do this below.

For soundness, we have the following

Lemma 1. *There exists a probabilistic algorithm A with the following property: assume that some prover P^* can execute the protocol on input commitment c with success probability at least $1/C + 1/f(k)$ for some polynomial $f()$. Then A, using P^* as oracle, runs in expected polynomial time and outputs a valid opening of c except with negligible probability.*

Protocols of the form we consider here often have an error probability that is 1 divided by the number of challenges, here $1/C$. However, in our case, we must use in our proof the computational assumptions we made about the group. So the reason why our probability of cheating may be (negligibly) larger than C is that P^* may have a slight chance of breaking these assumptions.

To show the lemma, note first that by assumption on P^*, using standard rewinding techniques, our algorithm A can obtain a situation where, for a given d, P^* could answer two different values e and e' with numbers u, v and u', v', so we get $g^{u-u'}h^{v-v'} = c^{e-e'}$. Now, suppose that $(e-e')$ divides both $(u-u')$ and $(v-v')$. Then the element $b = g^{(u-u')/(e-e')}h^{(v-v')/(e-e')}c^{-1}$ satisfies that $b^{e-e'} = 1$. It follows by the small order assumption on G that except with negligible probability $b^2 = 1$, and so c can be correctly opened by sending $(u-u')/(e-e'), (v-v')/(e-e'), b$. Therefore, we are done, if we can prove that the case where $e-e'$ does not divide both of $u-u', v-v'$ happens with negligible probability.

So assume that this "bad" case does indeed happen with non-negligible probability. We will show that this would mean that we could construct an algorithm violating our asumptions on the group. Suppose we get as input $h \in G$ chosen at random. By the assumptions, there is significant probability that $ord(h)$ is C-rough, so we assume this in the rest of the analysis. We then set $g = h^\alpha$ for random $\alpha \in [0..2^{2B}]$. Note that g, h have exactly the same distribution as in "real life". We send g, h to the adversary and do the proof that we know the discrete log of g base h. We then do the above rewinding based approach and hope that we get to a situation where we have $g^{u-u'}h^{v-v'} = c^{e-e'}$ and $e-e'$ does not divide both of $u-u', v-v'$. Let E be the event that this happens. We have by assumption that E occurs with non-negligible probability.

If we plug in $g = h^\alpha$, we get

$$h^{\alpha(u-u')+(v-v')} = c^{e-e'}.$$

Suppose wlog that $e > e'$. Then the rest of the analysis splits in two cases:

$e-e'$ **does not divide** $\alpha(u-u')+(v-v')$.

In this case, let $d = gcd(e-e', \alpha(u-u')+(v-v'))$ (where by assumption $d < e-e' \leq C$). Choose γ, δ such that

$$\gamma(e-e') + \delta(\alpha(u-u')+(v-v')) = d$$

We then get that

$$\begin{aligned} h^d &= h^{\gamma(e-e')+\delta(\alpha(u-u')+(v-v'))} \\ &= (h^\gamma c^\delta)^{e-e'} \end{aligned}$$

If we set $\tilde{b} = (h^\gamma c^\delta)^{(e-e')/d}h^{-1}$, it is clear that $\tilde{b}^d = 1$, and furthermore

$$h\tilde{b} = (h^\gamma c^\delta)^{(e-e')/d}$$

If $\tilde{b} = 1$, we have a solution to the strong root problem. Otherwise, we have $\tilde{b} \neq 1, 0 < d \leq C$, so we can break the small order assumption unless $\tilde{b}^2 = 1$. In this case, if $(e - e')/d$ is odd, then $\tilde{b}^{(e-e')/d} = \tilde{b}$, inserting this in the above yields again a solution to the strong root problem. But if $(e - e')/d$ is even, then (by the group assumptions) $(h^\gamma c^\delta)^{(e-e')/d}$ has odd order, which contradicts the fact that $ord(h\tilde{b}) = 2ord(h)$. In summary, if $e - e'$ does not divide $\alpha(u - u') + (v - v')$, we can break the assumptions on the group.

$e - e'$ **divides** $\alpha(u - u') + (v - v')$.

Note that even in this case, we still have that $e - e'$ does not divide both of $u - u', v - v'$. The goal will be to show that since the adversary does not know full information about our choice of α, this case happens with probability at most 1/2, given that E occurs. Hence the previous case where we could break the assumptions happens with probability at least 1/2, given E. Let q be some prime factor in $e - e'$ such that q^j is the maximal q-power dividing $e - e'$, and at least one of $u - u', v - v'$ are non-zero modulo q^j (such a q must exist since $e - e'$ does not divide both of $u - u', v - v'$). Note that if q^j divides $u - u'$, it would have to divide $v - v'$ as well, which is a contradiction. So $u - u' \neq 0 \bmod q^j$. We can then write $\alpha = y + z \cdot ord(h)$, where $y = \alpha \bmod ord(h)$. Note that g represents all information the adversary has about α (since the interactive proof that $g \in < h >$ is statistical zero-knowlegde), and y is uniquely determined from g, whereas z is completely unknown. Now, if indeed q^j divides $\alpha(u - u') + (v - v')$, we have

$$\alpha(u - u') + (v - v') = z(u - u')ord(h) + y(u - u') + (v - v') = 0 \bmod q^j$$

Note that since $q < C$ we have $ord(h) \neq 0 \bmod q$. Now, from the adversary's point of view, z is chosen uniformly among at least 2^B values, and must satisfy the above equation in order for the bad case to occur. The number of solutions modulo q^j of this equation is at most $gcd((u - u')ord(h), q^j)$. This number is a power of q, but is at most q^{j-1}. Then, since 2^B is much larger than q^j, it follows that the probability that z satisfies the equation is statistically close to $1/q \leq 1/2$.

4.2 A multiplication protocol

Using techniques similar to the above, we can also get a protocol for proving that three given commitments c_1, c_2, c_3 contain numbers x_1, x_2, x_3 such that $x_3 = x_1 x_2$. We assume that $c_i = g^{x_i} h^{r_i}$, and as before that the x_i's are numerically smaller than T. Note that then we have $c_3 = c_1^{x_2} h^{r_3 - x_2 r_1}$. We exploit this in the second step below.

1. P proves using the protocol from above that he can open c_1.
2. (a) P chooses at random $y \in [0..CT2^k[, s_2 \in [0..C2^{B+2k}[, s_3 \in [0..CT2^{B+2k}[$ and sends $d_2 = g^y h^{s_2}, d_3 = c_1^y h^{s_3}$ to V.
   (b) V chooses at random e between 0 and C and sends to P.
   (c) P sends $u = y + ex_2, v_2 = s_2 + er_2$ and $v_3 = s_3 + e(r_3 - x_2 r_1)$. V checks that $g^u h^{v_2} = d_2 c_2^e$ and $c_1^u h^{v_3} = d_3 c_3^e$.

We prove security of this protocol. As before, completeness is trivial and honest verifier statistical zero-knowledge follows by a similar argument as for the proof of opening protocol.

For soundness, assume as before that some prover P^* can execute the protocol with success probability $1/C + 1/f(k)$ for some polynomial $f()$. We then want to construct an algorithm that computes valid openings of c_1, c_2, c_3 revealing values x_1, x_2, x_3 with $x_1 x_2 = x_3$.

We can first use the result from the previous subsection to extract from the first step a way to open c_1 correctly, i.e. we have x_1, s_1, b such that $c_1 = g^{x_1} h^{s_1} b$ and $b^2 = 1$. Using standard rewinding in the second step, we can, for a given d_2, d_3, obtain correct answers u, v_2, v_3 and u', v_2', v_3' to challenges e, e', in expected polynomial time. Observe that we can in fact ensure that $e - e'$ is always an even number: fix any state for P^* just before it receives the challenge, and let S be the subset of challenges that it answers correctly. Since the number of challenges is superpolynomial, we may assume that the size of S is superpolynomial too. Then since more than half the numbers in S is even or more than half are odd, the probability that two random elements drawn from S have the same parity is at least a constant (in fact at least about $1/4$).

Now, since the verifier accepts, we have equations

$$g^u h^{v_2} = d_2 c_2^e, \quad c_1^u h^{v_3} = d_3 c_3^e$$

$$g^{u'} h^{v_2'} = d_2 c_2^{e'}, \quad c_1^{u'} h^{v_3'} = d_3 c_3^{e'}$$

dividing corresponding equations, we get

$$g^{u-u'} h^{v_2 - v_2'} = c_2^{e-e'}, \quad c_1^{u-u'} h^{v_3 - v_3'} = c_3^{e-e'}$$

Using the first equation in exactly the same argument as for the previous protocol, we can show that, unless the group assumptions are broken, it must be the case that $e - e'$ divides $u - u'$ and $v_2 - v_2'$, and so we get a correct way to open c_2, where the value contained in c_2 will be $x_2 := (u - u')/(e - e')$. If plug into the second equation our expression for c_1, we get

$$b^{u-u'} g^{x_1(u-u')} h^{s_1(u-u') + v_3 - v_3'} = c_3^{e-e'}$$

But since $e - e'$ is even and divides $u - u'$, we have $b^{u-u'} = 1$. Now we have an equation of the same form as the one we used in the proof of the previous protocol. Thus, if $e - e'$ divides both $x_1(u - u')$ and $s_1(u - u') + v_3 - v_3'$, we get a correct way to open c_3, and the value contained will be $x_1(u - u')/(e - e') = x_1 x_2$ and we are done. If there is a significant probability that this is not the case, we can use the same argument as above: we play the rewinding game against P^* in a situation where we know the discrete log of g base h, and show that we can break the group assumptions. The only difference is that the game is played such that we only continue to the end if $e - e'$ is even. But this makes no difference as the argument above is independent of the particular value of $e - e'$.

4.3 What was wrong with the proofs in [5]?

For completeness, we briefly indicate here what the problem was with the proof of soundness for the protocols suggested in [5]: those protocols are very similar to the ones we suggest here, in particular they have the same 3-move form, with a challenge e from the verifier as the second message. So [5] uses a rewinding argument as we do here, to obtain correct answers from the prover to challenges e, e'. However, a problem occurs in the last part of the proof, which roughly speaking coresponds to the last case in our analysis ($(e-e')$ divides $\alpha(u-u')+(v-v')$). Translated to our notation, it is claimed that the adversary cannot make this case occur with large probability unless $e-e'$ divides both $u-u'$ and $v-v'$, because he does not have enough information about α. However, if $e-e'$ is a small number, the bad case may in fact happen with large probability, and nothing in the proof can ensure that $e-e'$ is large. Moreover, even we know that $e-e'$ is large, there are additional tricks the adversay can play if $e-e'$ has a small prime factor, so additional ideas such as what we provide here seem necessary for this type of proof to go through.

We do not know if one can give correct proof for exactly the commitment scheme and protocols suggested in [5] - as mentioned our commitment scheme and protocols are not exactly the same as those of [5], even when specialized to $G = Z_n^*$. However, since our protocols are as efficient as those of [5], this question seems to be of minor importance.

4.4 Making the auxiliary protocols be zero-knowledge

We sketch here one of several possible techniques for constructing zero-knowledge protocols from the honest verifier zero-knowledge protocols we have shown above. Of course, one can always use the Fiat-Shamir heuristic (where the challenge is computed by applying a hash function to the first message) to make the protocols be non-interactive. They can then be shown to be zero-knowledge if one is willing to assume the random oracle model.

Here, we show another option that works without random oracles and uses no additional computational assumptions. For this, we make the following addition to the set-up phase of the commitment scheme: let Q be a prime chosen such that $Q > 2^B$ and also $log_2 Q$ is larger than the length of any message sent in the above protocols. This implies that Q does not divide $ord(G)$. It is simple to specify a way to choose Q such that both prover and verifier can compute Q efficiently. Now, the verifier chooses at random $H' \in G$, and sets $H = H'^Q$. He sends H to the prover and proves in zero-knowledge knowledge of the Q'th root H'. This can be done using the well-known protocol of Guillou and Quisquater, with challenges restricted to 1 bit to make the protocol be zero-knowledge (we loose some efficiency this way, but we only need to do this once and for all in the set-up phase).

Now, the prover can commit to a number m modulo Q by sending $C = H^m R^Q$ where R is chosen randomly in G. To open, one reveals m, R. Such a commitment is uniformly random in G, independently of m,and so is perfectly hiding. Also, if the prover could open a commitment in two different ways, it is easy to see that he could then compute a Q'th root of h, and this contradicts the strong root assumption. Finally, this is a *trapdoor commitment scheme*: given a Q'th root H' of H, it is easy to make a commitment, and then open it in any way desired - we simply set $C = R^Q$ for a random R. To "open" this to reveal m, send m, RH'^{-m}.

Now, observe that the protocols we have suggested so far are built in the standard 3-move form: the prover sends a message m, the verifier sends a challenge e, and the provers replies with some string z, and we have show these building blocks to be honest verifier zero-knowledge. In fact, each of the honest verifier simulations is of a form that allows to first decide on e and then compute m, z with the correct distribution given e.

Each of these auxiliary protocols is now transformed as follows: we ask the prover to send a commitment $C = H^m R^Q$ in stead of m. The verifier sends e as before, and the prover replies with R, m, z, where z is answer that would normally be sent in reponse to e. The verifier checks that m, R opens C correctly and that (m, e, z) is an acceptable conversation in the original protocol.

With this change, it is easy to show that the set-up protocol followed by any number of commitments and executions of the auxiliary protocols is zero-knowledge: the simulator extracts the Q'th root of H from the set-up phase using rewinding of the verifier, and then uses this to simulate the rest. To simulate the execution of one of the auxiliary protocols, first show a fake commitment $C = R^Q$ to the verifier. When e is returned, compute m, z with the correct distribution and an R' such m, R' is a valid opening of C.

Furthermore, soundness is also preserved: assuming that some prover P^* is succesful with non-negligible probability, standard rewinding allows to get good ansers to two different values of e. These answers will either break the commitment scheme based on H, or give correct answers to different challenges in the original protocol. Since the first case occurs with negligible probability by the strong root assumption, we can apply the soundness proof for the original protocol.

5 Applying the Scheme in Class Groups and Beyond

We do not give any detailed introduction to class groups here - or more precisely, class groups of quadratic number fields. It is enough to know, that each such group is defined by a single number, the *discriminant* Δ. Given this number, one can choose elements in the group and compute the group and inversion operations. Finding the order of the class group (the class number) from Δ appears to be a hard problem, and is at least as hard as factoring Δ (if Δ is composite). Therefore, root extraction also appears to be a hard problem, and it seems reasonable to conjecture that if Δ is chosen randomly from a large set of values, then the class number will contain large and random prime factors, and will not have a very large factor consisting of only small primes. Various heuristics (but no proofs) supporting this are known. Finally, it is known that if Δ is a prime, then the class number is odd. All this together makes it a reasonable conjecture that class groups constructed from large, random and prime discriminants would satisfy the assumptions we made in the beginning, for some appropriate choice of C[1].

The only difficulty is that we assumed in our description of the set-up procedure that V can choose the element h such that its order is guaranteed to be C-rough. There is no known way to do this with a class group, because there is no efficient algorithm known that allows to generate a class group with known order. However, we can reasonably conjecture that it is hard for the prover to distinguish a random h from an h with C-rough order. Assuming this, V can choose h at random, and P will be not able to do significantly better in this case than if h had C-rough order.

In general, if we add this indistinguishability condition on h to our basic group assumptions, then our scheme will work in all cases, even for generators $\mathcal{G}$ that output nothing but the public description of G.

6 Extentions

A first observation is that a group G may satisfy the requirements we make here, even if some information about its order is publically available. For instance, this is the case if it is a direct product $G \simeq H \times G'$, where the order of H is public with only large prime factors and G' is a group that already has the properties we require here. This observation is relevant when $G = Z_{n^2}^*$ for an RSA modulus n, since then G is the direct product of a cyclic group of order n and a group isomorphic to Z_n^*.

Another observation is that our construction and protocols can also be applied to cases where we want a computationally hiding (but unconditionally binding) commitment scheme: Consider the case where the parameters g, h are chosen such that $h = g^\alpha$ for some α, but commitments are still of the form $g^x h^r$. In this case, the commitment does relase information about x if the group generated by h is smaller than the group generated by g, but it may still be computationally hiding, if elements from different cosets of $< h >$ in $< g >$ are computationally indistinguishable. It is an unconditionally binding commitment to a particular such coset. Along these lines, one can build a commitment scheme with associated protocols, where a commitment is an Okamoto-Uchiyama encryption [7].

References

1. Boudot: *Efficient Proof that a Comitted Number Lies in an Interval*, Proc. of EuroCrypt 2000, Springer Verlag LNCS series 1807.

[1] There are some heuristics known for how the factorization of a class number can be expected to behave, C should be chosen with this in mind.

2. Cramer and Damgård: *Zero-Knowledge Proofs for Finite Field Arithmetic or: Can Zero-Knowledge be for Free?*, proc. of Crypto 98, Springer Verlag LNCS series 1462.
3. Damård: *Practical and Provably Secure release of a Secret and Exchange of Signatures*, J.Cryptology, vol. 8 1995, pp. 201-222.
4. Fujisaki: *A simple Apporach to Secretly Sharing a Factoring Witness in a Publically-Verifiable Manner*, Manuscript, 2000.
5. Fujisaki and Okamoto: *Statistical Zero-Knowledg Protocols to prove Modular Polynomial Relations*, proc. of Crypto 97, Springer Verlag LNCS series 1294.
6. T. Pedersen: *Non-Interactive and Information Theoretic Secure Verifiable Secret Sharing*, proc. of Crypto 91, Springer Verlag LNCS, vol. 576, pp. 129–140.
7. Tatsuaki Okamoto , Shigenori Uchiyama: *A New Public-Key Cryptosystem as Secure as Factoring* Proceedings of EuroCrypt 98, Springer Verlag Lecture Notes in Computer Science, 1403.