コンピュータセキュリティ 14－6
(2001. 7. 25)

A Note on Computationally Sound Proof in Group of Unknown Order

Ivan Damgård Eiichiro Fujisaki

Aarhus University,
Ny Munkegade AARHUS, C DK-8000 Denmark
e-mail: ivan@daimi.au.dk

NTT Laboratories
1-1 Hikarino-oka Yokosuka-shi,
239-0847 JAPAN
e-mail: fujisaki@isl.ntt.co.jp

Abstract

Suppose we are given an Abelian group G of unknown order, such as RSA group $(\mathbb{Z}/n\mathbb{Z})^{\times}$, where the group operations in G can be efficiently computed. Let g, h be elements in G and let $c = g^x h^r$ be a commitment to x (where the group operation is defined as the multiplication). In this paper we revisit a sound-proof-of-knowledge protocol for the representation problem in a group of unknown order — that is, a protocol in which the prover convinces the verifier that he knows the representation of c to base g, h in G.

The proof of soundness for this protocol was initially provided in [5], but we have recently found it incomplete, although the protocol and its variants appear in many literatures, for instance PVSS [6], group signature [3, 4] and optimistic fair-exchange [2, 1].

In this paper we fix a bug in [5] and prove this protocol indeed sound, trying to make the setting more general and fundamental.

Keywords: Computationally sound proof, argument, zero-knowledge.

1 Introduction

Suppose we are given a way to construct an Abelian group G of unknown order, such as an RSA group or a class group. More precisely, we have an PPT algorithm $\mathcal{G}$ which on input 1^k outputs a description $\mathtt{descr}(G)$ 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. Given $\mathtt{descr}(G)$, we assume that one can compute efficiently some estimates on the order, $2^A \leq \mathtt{ord}(G) \leq 2^B$, where A and B are polynomial in k. We also assume that elements can be sampled randomly from the group and all the group operation (thereby, including inversion) can be computed efficiently. Throughout this paper, the group operation is defined as multiplication.

Here is the assumptions about the group:

Strong root assumption. Given $\mathtt{descr}(G)$, an appropriate subgroup H in G, and a random element $y \in H$, it is hard to find $x \in H$ and an integer $t > 1$ such that $y = x^t$.

No elements with known order. $\mathtt{descr}(G)$ includes an integer C. This number is increasing as a function of k. However, it must be small enough, so that the numbers less than C can be factored in time polynomial in k. It should be hard to compute a pair (b, σ) such that $b \in G - \{1\}$, $1 < \sigma < C$, $\sigma \neq 2$, and $\mathtt{ord}(b) = \sigma$.

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

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

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. Note that the condition on C means that if one can find $b \neq 1, \beta$ such that $0 < \beta \leq C, b^\beta = 1$, this will allow you to find the order of b by first factoring β. This will therefore violate the assumption unless $\mathtt{ord}(b) = 2$. The third assumption is always true if $\mathtt{ord}(G)$ 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 $\mathtt{ord}(G)$: if we write $\mathtt{ord}(G) = FD$, where F has only prime factors less than C and D has only prime factors greater than C, then the assumption is satisfied if and only if 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. 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 according to the restrictions above, $C = 2^{\sqrt{k}}$ is one possibility, and 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 $\mathtt{descr}(G) = n, C$. Now, the root assumption is simply the strong RSA assumption. Finding elements of known order different from 2 and less than C is as hard as factoring n: given such an element, we factor the order and so we can find an element b of known prime order s. Unless $s = 2$, s cannot divide both $p-1$ and $q-1$ and therefore 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 $\Omega(1/k)$.

Previous works. The first computationally sound proof protocol for the representation problem in a group of unknown order appeared in [7]. This (or this kind of) protocol is like this: For given $c = g^s \bmod n$, the verifier accepts that the prover knows s if $e = H(g^x c^e \bmod n)$ for (u, e) sent by the prover, where n is a composite of large primes and H denotes a hash function. Later, the security analysis about soundness for such a protocol was given by the authors in [5] (Precisely speaking, in the case $c = g^s h^r \bmod n$). From [5], we can conclude:

> *Slightly modified statement (in the case $c = g^s$) derived from [5]:*
>
> *If there exists algorithm* Adv *that takes (g, n), makes c following her strategy, and finally outputs (c, x, e) such that $e = H(g^x c^e \bmod n)$ (in the random oracle model) with a non-negligible probability, then there is a knowledge extractor $\boldsymbol{K}$ that runs with an expected polynomial time and outputs s such that $c = g^s \bmod n$ except for some negligible error (the probability of breaking the strong (or flexible) RSA problem).*

This is because: Suppose Adv outputs (c, x, e) and (c, x', e') such that $g^x c^e = g^{x'} c^{e'}$. Then $c^{\Delta e/d} = g^{\Delta x/d}$ holds, where $\Delta e = e - e'$, $\Delta x = x' - x$ and $d = \gcd(\Delta e, \Delta x)$. Then if $d \neq \Delta e$, Adv is a solver of the strong RSA problem, because $g = g^{(\Delta x/d)a + (\Delta e/d)b} = (c^a g^b)^{\Delta e/d}$, where $a, b \in \mathbb{Z}$ such that $(\Delta x/d)a + (\Delta e/d)b = 1$, otherwise $\boldsymbol{K}$ can output s such that $c = g^s \bmod n$ by $s = \frac{\Delta x}{\Delta e} \in \mathbb{Z}$. (Precisely speaking, if $c \in <g>$, then $c = g^s \bmod n$ otherwise $-c = g^s \bmod n$. See the "$\pm c = g^s$ problem below.) Hence, sound proof is completed in this case.

The case $c = g^s h^r$ is slightly different from the above case. Similarly, we can get $c^{\Delta e/d} = g^{(\Delta x + \alpha \Delta y)/d}$, where $h = g^\alpha \bmod n$ and $d = \gcd(\Delta e, \Delta x + \alpha \Delta y)$. From the intractability of the SRSA problem, we can assume $d = \Delta e$ as well. [5] concluded that Δx, Δy must be divided by Δe, because α is unpredictable except "$\alpha \bmod \mathtt{ord}(g)$", where α is large integer given by the knowledge extractor. However, if Δe is very small then the following event might happen with some

probability: "$\Delta e|(\Delta x + \alpha \Delta y)$, but Δx is not divided by Δe". In [5], this part of discussion is missing.

Independently from this work, Poupard and Stern in [8] gave a sound proof for $c = g^s$, though in a *weaker* soundness model.

2 Commitment Scheme

Based on the above, 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 $\texttt{descr}(G)$ (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. The protocols will be constructed to have error probability $1/C$, where C is the number from the assumptions above.

Consider the following commitment scheme:

Set-up. V runs $\mathcal{G}$ and chooses a random element $h \in G$, such that $\texttt{ord}(h)$ is C-rough. Now V sets $g = h^\alpha$, where α is randomly chosen in $[0..2^{2B}]$. V sends $\texttt{descr}(G), g, h$ to P and proves that $g \in < h >$, by the standard zero-knowledge discrete log protocol with binary challenges. This is slow, but works in any group and only needs to be done once and for all.

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

Open. To open a commitment, P must send x, r, b, β such that $c = g^x h^r b$, and $b^2 = 1$. 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, suppose some prover P^* could create c, and $(x, r, b), (x', r', b')$, valid openings with $x \neq x'$. 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 $h^{\tilde{\sigma}(\alpha(x-x')+r-r')} = 1$. Since α is chosen much larger than the order of h, P^* does not have full information on α, it is only determined modulo the order of h. Hence (since $x - x' \neq 0$), there is a non-negligible probability that $(\alpha(x - x') + r - r') \neq 0$. If this number is non-zero, it is a multiple of the order of h, and it follows that V and P^* together could solve the strong root problem on input h.

3 Computationally Sound Proof in Group of Unknown Order

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

1. P chooses y, s at random and sends $d = g^y h^s$ to V.

2. V chooses at random e between 0 and 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 zero-knowledge if y, s are chosen at random such that they are much larger than x and er, respectively, for instance we can choose s in the interval $[0..2^{3B}[$. There are then a number of known techniques by which a zero-knowledge protocol can be constructed from it. Observe for instance that the set-up protocol of the commitment scheme followed by any number of instances where the prover commits and then executes the above protocol, is zero-knowledge: the simulator extracts the discrete log of g base h from the verifier using rewinding of the set-up phase, and can now easily simulate the rest.

To show soundness, we assume that some prover P^* can execute the protocol with a non-negligible success probability. This means that, using standard rewinding techniques, we 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 assumptions 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 assumptions on the group. Suppose we get as input $h \in G$ chosen at random. By the assumptions, there is significant probability that $\mathtt{ord}(h)$ 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 $u - u', v - v'$. 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 can break the assumption on "a few elements with known order". 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 $(h^\gamma c^\delta)^{(e-e')/d}$ has odd order, which contradicts the fact that $\mathtt{ord}(h\tilde{b}) = 2\mathtt{ord}(h)$. 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 $1/2$ - and hence the previous case where we could break the assumptions happens with significant probability. 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 $\alpha = y + z \cdot \mathtt{ord}(h)$, where $y = \alpha \bmod \mathtt{ord}(h)$. Note that g represents all information the adversary has about α (since the interactive proof that $g \in < h >$ is statistical zero-knowledge), 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

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

Note that since $q < C$ we have $\mathtt{ord}(h) \neq 0 \bmod q$. 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 $gcd((u - u')\mathtt{ord}(h), q^j)$. 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 $1/q \leq 1/2$.

Mod-Multi Protocol. We can then 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 $c_i = g^{x_i} h^{r_i}$. 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 y, s_2, s_3 at random 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 zero-knowledge follows if the provers random choices are from large enough intervals.

For soundness, assume as before that some prover P^* can execute the protocol with non-negligible success probability. We can first use the above result 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 super-polynomial, we may assume that the size of S is super-polynomial 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 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 in such a way that $e - e'$ is even. But this makes no difference as the argument above is independent of the particular value of $e - e'$.

References

[1] G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signature. In *Proceedings of the Fifth Annual Conference on Computer and Communications Security*, pages 138–146, Singapore, November 1999. ACM.

[2] F. Bao. An efficient verifiable encryption scheme for encryption of discrete logarithm. In J. J. Quisquater and B. Schneier, editors, *Smart Card Research and Aplications, CARDIS'98*, volume 1820 of *Lecture Notes in Computer Science*, pages 213—220, Louvain-la-Neuve, Belgium, 2000. Springer-Verlag.

[3] J. Camenisch and M. Michels. A group signature based on an rsa-variant. Technical report, Technical Report RS-98-27, Aarhus, November 1998.

[4] J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes. In M. Wiener, editor, *Advances in Cryptology — CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 413–430. Springer-Verlag, 1999.

[5] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer-Verlag, 1997.

[6] E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 32–46. Springer-Verlag, 1998.

[7] M. Girault. Self-certified public keys. In D. W. Davies, editor, *Advances in Cryptology — EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer-Verlag, 1991.

[8] G. Poupard and J. Stern. Security analysis of a practical "on the fly" authentication and signature generation. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 422–436. Springer-Verlag, 1998.