A Cautionary Note Regarding Cryptographic Protocols Based on Composite Integers

Steven D. Galbraith[1], Wenbo Mao, Kenneth G. Paterson[1]
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2001-284
November 8th , 2001+

E-mail: Steven.Galbraith@rhul.ac.uk, Kenny.Paterson@rhul.ac.uk, wm@hplb.hpl.hp.com

These days it is rather common in cryptology to see ideas which originated in the setting of finite fields being extended to $\mathbb{Z}^*_N$. However, the security results do not necessarily generalise to $\mathbb{Z}^*_N$. In this paper we illustrate this phenomenon by pointing out a flaw in the soundness proof of a zero-knowledge protocol in a timed commitment scheme of Boneh and Naor.

+ Internal Accession Date Only Approved for External Publication
[1] Information Security Group, Mathematics Department, Royal Holloway University of London, Egham, Surrey TW20 0EX, UK

A cautionary note regarding cryptographic protocols based on composite integers

Steven D. Galbraith[1]*, Wenbo Mao[2] and Kenneth G. Paterson[1]**

[1] Information Security Group, Mathematics Department,
Royal Holloway University of London, Egham, Surrey TW20 0EX, UK.
Steven.Galbraith@rhul.ac.uk, Kenny.Paterson@rhul.ac.uk
[2] Mathematics, Cryptography and Security Group,
Hewlett-Packard Laboratories, Bristol,
Filton Road, Stoke Gifford, Bristol BS34 8QZ, UK.
wm@hplb.hpl.hp.com

Abstract. These days it is rather common in cryptology to see ideas which originated in the setting of finite fields being extended to $\mathbb{Z}_N^*$. However, the security results do not necessarily generalise to $\mathbb{Z}_N^*$. In this paper we illustrate this phenomenon by pointing out a flaw in the soundness proof of a zero-knowledge protocol in a timed commitment scheme of Boneh and Naor.

1 Introduction

These days it is rather common in cryptology to see ideas which originated in the setting of finite fields being extended to $\mathbb{Z}_N^*$. However, there is usually an asymmetry for protocols using $\mathbb{Z}_N^*$ since one party knows the group order while the other parties don't. Therefore, the owner of the factorisation of N can potentially cheat in a way which cannot be detected by the other parties. The purpose of this paper is to emphasise this subtlety in the development of cryptographic protocols which use $\mathbb{Z}_N^*$.

As an illustration we study the timed commitment scheme of Boneh and Naor [1]. This is a commitment scheme where the committed value can be recovered, after a certain amount of computational effort (time), without the assistance of the committer. Part of the commitment in their scheme is the number $u = g^{2^{2^k}} \pmod N$ where N is a product of two large primes. When the factorisation of N is known then it is easy to compute u (just compute $a = 2^{2^k} \pmod{\varphi(N)}$ and then $u = g^a \pmod N$). The underlying assumption is that, when the factorisation of N is not known, computing u cannot be done any faster than the cost of 2^k serial squaring operations.

For their scheme it is important that a committer prove to a verifier that the number u is of the correct form. In Section 2 we recall the protocol suggested by Boneh and Naor to achieve this. In [1] it is claimed that a dishonest committer cannot cheat in that protocol with probability better than about k/d where d depends on the factorisation of N. A verifier knows that $d > B$ where B is a certain security parameter. Boneh and Naor suggest $k \sim 40$ and $B \sim 128$ so, as far as the verifier is concerned, the cheating probability is around $1/3$.

In Section 3 we show that a dishonest committer has a cheating strategy for which the probability of successful cheating is $1/2$ (regardless of the chosen values for k and B). The general principle behind the attack in Section 3 is a well-known trick available in groups which contain elements of small order. Such an attack is usually prevented by checking the orders of group elements during the verification process, but this cannot be performed when the verifier does not know the relevant group order.

* This author thanks Hewlett-Packard labs for support
** This work was undertaken while this author was employed by Hewlett-Packard labs, Bristol.

It is not possible to point out the error in the soundess proof of [1] because a proof is not given. Instead the authors simply refer to the work of [2], which is based on finite fields rather than $\mathbb{Z}_N^*$.

We stress that the basic principles of the scheme of Boneh and Naor are sound, though a solution with greater functionality has been provided by Mao [5].

We hope that this paper acts as a warning to researchers in the cryptologic community to be very careful when adapting protocols from finite fields to the case of $\mathbb{Z}_N^*$ (see Section 3 of [3] for some related examples).

2 The protocol of Boneh and Naor

The committer constructs $N = p_1 p_2$ to be a product of two primes. In [1] there is no requirement that the primes have a special form. The committer chooses an element $h \in \mathbb{Z}_N^*$ and sends it to the verifier. Using this, the committer and verifier can both construct an element $g \in \mathbb{Z}_N^*$ such that no primes $l < B$ divide the order of g. This condition on the order of g is intended to enable the proof of the soundness result, but as we will show, this condition is not sufficient. Let q be the order of g (which is only known to the committer).

The committer has published an element $u \in \mathbb{Z}_N$ and an integer k and wants to prove to a verifier that $u \equiv g^{2^{2^k}} \pmod{N}$.

To achieve this the committer publishes $b_i = g^{2^{2^i}} \pmod{N}$ for $i = 0, 1, \ldots, k$ (i.e., $b_k = u$). The committer then proves that each triple (g, b_{i-1}, b_i) is of the form (g, g^x, g^{x^2}) using the following protocol:

1. The verifer sends commitments to integers $0 \leq c_i \leq R$ for $i = 1, 2, \ldots, k$ (where R is a security parameter).
2. The committer chooses random numbers $\alpha_i \in \mathbb{Z}_q$ for $i = 1, 2, \ldots, k$ and sends the values $z_i = g^{\alpha_i} \pmod{N}$, $w_i = b_{i-1}^{\alpha_i} \pmod{N}$ to the verifier.
3. The verifier opens the commitments to the c_i.
4. The committer checks the commitments and, if they are all correct, sends $y_i = c_i 2^{2^{i-1}} + \alpha_i \pmod{q}$ for $i = 1, 2, \ldots, k$ to the verifier.
5. The verifier checks whether

$$g^{y_i} b_{i-1}^{-c_i} \equiv z_i \pmod{N} \qquad \text{and} \qquad b_{i-1}^{y_i} b_i^{-c_i} \equiv w_i \pmod{N}$$

for $i = 1, 2, \ldots, k$ and accepts the run of the protocol if all these congruences are satisfied.

It is easy to check that this protocol is complete (i.e., that a verifier will accept the protocol if the committer performs all operations correctly). In [1] it is claimed that the probability of sucessful cheating by the committer in this protocol is at most $k(1/d + \beta(d-\beta)/(dR^2))$ where d is the smallest prime factor of q and where $\beta = R \pmod{d}$. Since the verifier only knows that $d > B$ a suspicious verifier can only know that the probability of successful cheating is around k/B.

3 The flaw

Consider the following cheating strategy by a dishonest committer. Instead of publishing the correct value $g^{2^{2^k}} \pmod{N}$ one can publish $u = \xi g^{2^{2^k}} \pmod{N}$ where $\xi \in \mathbb{Z}_N^*$ has order l. Finding an element ξ is easy since the committer knows the factorisation of N and may have even constructed N specifically with this attack in mind. Even without knowing the factorisation one can always choose $\xi = -1$ and $l = 2$.

We now explain how to behave during the protocol described in Section 2. All the b_i may be calculated correctly for $i = 1, 2, \ldots, k-1$. The value for b_k is of course set to be equal to u. Then, all the z_i, w_i and y_i may be calculated correctly for $i = 1, 2, \ldots, k$ with the exception of w_k which

is taken to be $\xi^r b_{k-1}^{\alpha_k}$ where r is chosen at random in $\mathbb{Z}_l$. All the checks by the verifier will be successful except possibly for the ones involving b_k or w_k. The only equation involving either b_k or w_k is the test whether

$$b_{k-1}^{y_k} b_k^{-c_k} \equiv w_k \pmod{N}.$$

If $-c_k \equiv r \pmod{l}$ then this check will pass successfully.

It is obvious that the probability of successful cheating is $1/l$. In the case where ξ has order 2 then this is a higher probability of cheating than that claimed by Boneh and Naor.

It is possible to develop a scheme with a higher soundness probability using methods such as those in [4], [3] and [5]. However, we believe that it is necessary to use a modulus with a special structure to achieve this (e.g., a product of safe primes). Basically, it is not only necessary to ensure that a specific generator does not have small primes dividing its order, but to ensure that the group itself does not have small primes dividing its order (obviously the prime 2 has to be handled in a different way).

References

1. D. Boneh and M. Naor, Timed commitments, in M. Bellare (ed.), Crypto 2000, Springer LNCS 1880, 2000, 236–254.
2. D. Chaum and T. P. Pedersen. Wallet databases with observers, in E. Brickell (ed.), Crypto '92, Springer LNCS 740, 1993, 89–105.
3. S. D. Galbraith, W. Mao, and K. G. Paterson, RSA-based undeniable signatures for general moduli, to appear in proceedings of RSA 2002.
4. P.-A. Fouque and J. Stern, Fully distributed threshold RSA under standard assumptions, in C. Boyd (ed.), Asiacrypt 2001, Springer (2001).
5. W. Mao, Timed release cryptography, in the proceedings of SAC 2001.