A secret sharing scheme of prime numbers based on hardness of factorization

Kai-Yuen Cheong *

April 17, 2012

Abstract

Secret sharing schemes usually have perfect information theoretic security. This implies that each share must be as large as the secret. In this work we propose a scheme where the shares are smaller, while the security becomes computational. The computational security assumption is hardness of factorization, which is a simple and rather standard assumption in cryptography. In our scheme, the shared secret can only be a set of prime numbers.

1 Introduction

Secret sharing schemes usually have information theoretic security [2]. Computational security is rarely considered as it is not as good as information theoretic security. But this implies that each share must be as large as the secret. In this paper, we work on the computational security basis and propose a scheme where each share is smaller than the secret.

In a case where the secret is N bits and is shared among k people who must all come together to recover the secret, it is clear that the average share size must be at least N/k just for correctness of the recovered secret. For security in information theoretic sense, each share must be at least N bits, because it is required that information obtained must be zero by any $k-1$ users working together. Any scheme to overcome this bound must leak partial information of the secret. In the ideal case, the information leakage is uniform and can be controlled, as in the Ramp Secret Sharing (RSS) method [1]. Note that some kind of trivial RSS can be done by literally breaking the secret into k parts as a string, without any transformation. If computational security is used instead, there is no such restriction. The shares may be sized between N/k and N, and the secret is completely protected as long as it is computationally infeasible to obtain the secret from any $k-1$ shares.

*School of Information, Japan Advanced Institute of Science and Technology. Email: kaiyuen@jaist.ac.jp

In this work we proposed a special scheme to share $2^k - 1$ prime numbers. It is not a general secret sharing scheme, but since prime numbers are often used in cryptography, this scheme can be used in many situations. The share size is about half of the secret size. The computational assumption is the hardness of factorization and nothing else.

2 Size of input and output of the scheme

In our arguments about the size of secrets and shares, we note the following facts. First, the product of two n-bit numbers is a number of $2n$ bits. In terms of information, the product does not really have $2n$ bits because, while there are indeed $2n$ bits of information before the multiplication, some information is lost after the multiplication, as the product may have other forms of factorization. On the other hand, some $2n$-bit numbers are primes and have no factorization. But anyway the space required to store a product of two n-bit numbers is $2n$ bits. We call this the size, without regarding to the real information content. The same general argument applies to products of more numbers. This is how we define the size of share in our scheme.

Second, our secret is a set of prime numbers of n bits each. One may argue that there are less than n bits of information in one such prime number, because of the restriction that it be prime. The prime number theorem states that the number of primes smaller than x is approximately $x/\ln x$. It means that we need about $n - \log_2 n$ bits to address the list of primes up to n bits in their plain numerical values. Therefore it can be justified that prime numbers need not be encoded because the space saving would not be significant, and we argue that an n-bit prime number has close to n bits of information.

3 The secret sharing method

To begin with, a list of $2^k - 1$ prime numbers, each about n-bit, is generated. They are called p_j where $1 \le j \le 2^k - 1$. Next, define $B(j, i)$ as a function such that bit i of the binary expression of integer j is extracted, counting from the most significant bit. For example, $B(6, 1) = 1$ and $B(6, 3) = 0$.

Next, for each user i, where $1 \le i \le k$, calculate $s_i = \prod_{j=1}^{2^k-1} B(j, i)p_j$. This is the share to be given to user i. Note that each share is the product of about half of the secret prime numbers. So the share size is about half of total secret size, which is $N = n(2^k - 1)$ bits. To be more precise, each share is at most $n2^{k-1}$ bits. Hardness of factorization is required for security, as the shares are just products of the secret prime numbers.

As an example, assume that $k = 3$. Then $s_1 = p_4p_5p_6p_7$, $s_2 = p_2p_3p_6p_7$, and $s_3 = p_1p_3p_5p_7$.

4 The secret recovery method

First, all possible subsets of user shares are listed in the following manner. The set S_j includes s_i if and only if $B(j,i) = 1$. Sort these sets in order of descending size. The order for those with the same size is not important. For example, when $k = 3$, then the sorted list can be $(S_7, S_6, S_5, S_3, S_4, S_2, S_1)$, where $S_7 = \{s_1, s_2, s_3\}$ and $S_6 = \{s_1, s_2\}$, etc.

Let the items in this sorted list be S_{σ_j} for j from 1 to $2^k - 1$. In our example, $\sigma_1 = 7$, $\sigma_2 = 6$, etc. Next, the following iteration is done for j starting from 1 until $j = 2^k - 1$:

1. Calculate the GCD of the shares in S_{σ_j}. The result would be p_{σ_j}.
2. Update the shares in S_{σ_j} by dividing each share by p_{σ_j}.
3. Increase j by 1 and repeat the process.

At the end of this process, all the prime numbers p_j are recovered. In our example, the sequence of primes recovered is $(p_7, p_6, p_5, p_3, p_4, p_2, p_1)$.

5 Security proof

In the following we say that breaking the scheme is equivalent to factorization. Let there be an oracle which gives one p_j with non-negligible probability, by using an input of any $k - 1$ shares in the secret sharing scheme. This can be seen as the weakest oracle capable of breaking the scheme. Then we can try to solve the factorization problem.

First, the correctness of the number given by the oracle can be checked easily. So we can assume that there is no false-positive solution. Now let us say we are given a product $m = pq$ where p and q are primes. We create primes p_1 to p_{2^k-1} randomly, except that we define $p_r = p$ and $p_{r+1} = q$ without knowing them, for some random r such that r and $r + 1$ differ only on the last bit. This only requires that r is an even number.

Note that it is possible to create shares for user 1 to user $k - 1$ by knowing m but not p or q. These shares are the input to the oracle. If the oracle can randomly return one prime number, there is non-negligible probability that it is either p or q. Therefore the factorization problem can be solved in probabilistic polynomial time, where polynomial time means a polynomial in n, which is the size of the prime numbers p and q.

There seems to be a problem remaining. The oracle may require a particular set of shares rather than a random set of size $k - 1$. But our observation is that the scheme is actually symmetric for the users. Any permutation of the shares can be achieved through some permutation of the secret prime numbers. That means all shares are the same and the oracle cannot be designed to distinguish them. Therefore it must be that any $k - 1$ shares can be used as the input to the oracle.

6 Conclusion

The application of this scheme is limited to the case where the secrets are $2^k - 1$ prime numbers where k is the number of users, who must all be present to reconstruct the secrets. The computational assumption for security is the standard assumption on the hardness of factorization. The saving of space is an important feature of this scheme, where the share size is about half the total secret size. It is also important to note that the scheme is symmetric. That means the computational difficulty to break the scheme is the same for subsets of user of the same size.

References

[1] P. Paillier: On ideal non-perfect secret sharing schemes, In *Security Protocols Workshop*, pp.207–216, 1997.

[2] A. Shamir: How to share a secret, *Communications of the ACM*, 22(11), pp.612–613, 1979.