# A Patch Protocol for SHK Secret Transfer Scheme

Tian-Lung Lu† Jen-Chun Chang* Hsin-Lung Wu*

**Abstract**

Saied Hosseini Khayat proposed a scheme (SHK scheme) in 2008 to transfer a secret from sender to receiver such that the receiver cannot decrypt the secret without the consent of a group of trustees. This scheme is simple and excellent since it does not require any key exchange among sender, receiver, or trustees. But there is a problem in this scheme that the secret will be exposed when the private keys of the sender, receiver, and trustees are chosen improperly such that any one of some bad relations among the keys happens.

In this paper, a patch protocol for the SHK scheme is presented such that the private keys are guaranteed to be chosen properly and the privacy of all private keys is also assured.

## I. Introduction

The commutative property of encryption function has been explored and used in cryptography widely. For example, Shamir's keyless secret communication [3] is a good sample. Shamir also explored the power of commutativity in [5]. Agrawal [1] and Clifton [2] used the commutative property for security applications in distributed databases and data mining, respectively. Many related topics are surveyed and collected in Weis's MIT PhD dissertation [6].

Based on the commutative property of encryption, Saied Hosseini Khayat [7] considered the following situation. Suppose that Alice wants to transfer a secret to Bob securely such that bob cannot decrypt the secret unless a group of trustees agree. Though all involved parties can be trusted to follow a prescribed protocol, the communication channels are insecure. Furthermore, the secret must be protected not to be revealed to the trustees, nor to anyone but Bob. This situation often arises in many practical applications in commercial or military environments. At the first thought,

*Department of Computer Science and Information Engineering, National Taipei University, Taipei Country 237,Taiwan. E-mail: {jcchang, hsinlung@mail.ntpu.edu.tw}. This work was supported in part by the National Science Council of Taiwan under contract NSC-97-2218-E-305-001-MY2.

† Graduate Institute of Computer Science and Information Engineering, National Taipei University, Taipei County 237, Taiwan. E-mail: drake.lu@msa.hinet.net.

this problem seems easy to be solved by Shamir's threshold secret sharing scheme [4]. But in order to transfer the shares of the parties securely, the scheme normally requires some key exchange operations. It is inconvenient.

Saied Hosseini Khayat [7] proposed a scheme for the above problem. This scheme is very simple and excellent since it does not require any key exchange among sender, receiver, or trustees. But there is a problem in the scheme that the secret will be exposed if any bad relation among the private keys of the sender, receiver, and trustees occurs. The SHK scheme and the details of the secret leakage problem will be introduced in the next section.

In this paper, we design a patch protocol for the SHK secret transfer scheme. It is efficient and secure. When it is applied with/before the SHK scheme, the secret leakage problem is solved and all private keys are kept secret.

## II. The SHK secret transfer scheme

We briefly introduce the SHK scheme here. For readers want to know the details, please refer to [7]. Let $p$ be a large prime which can be published to all (even adversaries). The secret owner (Alice) is denoted by $P_0$. The recipient of the secret is denoted by $P_n$. The trustees are denoted by $P_1$, $P_2$, …, $P_{n-1}$. That is, there are $n + 1$ parties in total. The SHK protocol has three phases described below.

**Setup:**

Make a large prime $p$ public. $P_0$ has a secret $s \in Z_p$. For each $i \in \{0, 1, 2, \ldots, n\}$, party $P_i$ has his/her own private key pair $(a_i, b_i)$ such that $a_i, b_i \in Z_p$ , $\gcd(a_i, p - 1) = 1$, and $a_i b_i = 1 \pmod{p - 1}$.

**Locking (means "encryption"):** The secret is locked by all parties sequentially.

1. $P_0$ locks the secret $s$ by computing $c_0 = s^{a_0} \pmod{p}$ and sending $c_0$ to $P_1$.
2. For $i = 1, 2, \ldots, n - 1$ do

   $P_i$ locks the secret by computing $c_i = c_{i-1}^{a_i} \pmod{p}$ and sending $c_i$ to $P_{i+1}$.
3. $P_n$ locks the secret by computing $c_n = c_{n-1}^{a_n} \pmod{p}$ and sending $c_n$ to $P_0$.
4. $P_0$ removes her lock the secret by computing $s' = c_n^{b_0} \pmod{p}$ and sending $s'$ to $P_n$.

**Unlocking (means "decryption"):** The secret is unlocked by all trustees in an arbitrary order, and then unlocked by $P_n$ finally.

1. $P_n$ sends $s'$ to a trustee $P_i$, $i \in \{1, 2, \ldots, n-1\}$.
2. Each trustee $P_i$, in $\{P_1, P_2, \ldots, P_{n-1}\}$ (in an arbitrary order), removes his lock by computing $f(x) = x^{b_i} \pmod{p}$ on his received data and sends the result to the next trustee. The last trustee removes his lock and sends the result to $P_n$.
3. Finally, $P_n$ removes his own lock by computing $f(x) = x^{b_n} \pmod{p}$ on his received data and then finds the secret out.

The correctness and security based on discrete logarithm problem (DLP) have been shown in [7]. But, as the author said, there is nonzero possibility (though the probability is low) that the secret is exposed in the locking phase when the private keys are improperly selected such that any one of the following condition happens.

$$a_0 a_1 = 1 \pmod{p-1}, \text{ or}$$
$$a_0 a_1 a_2 = 1 \pmod{p-1}, \text{ or}$$
$$\ldots = \ldots$$
$$a_0 a_1 \ldots a_n = 1 \pmod{p-1}.$$

To solve the secret leakage problem, we want to make sure that

$$a_0 a_1 \neq 1 \pmod{p-1}, \text{ and}$$
$$a_0 a_1 a_2 \neq 1 \pmod{p-1}, \text{ and}$$
$$\ldots \neq \ldots$$
$$a_0 a_1 \ldots a_n \neq 1 \pmod{p-1}.$$

If any above inequality is not true, we must coordinate someone to change his private key pair until all above inequalities are true. Note that during the checking of inequalities and the coordination of changing keys, each private key should be prevent from exposing to others.

## III. Our proposed patch protocol

In this section, we will introduce a patch protocol to completely prevent the secret leakage problem in the locking phase. First, the secret leakage problem can be reformulated as follows.

**Environment:**

A large prime $p$ is public. For each $i \in \{0, 1, 2, \ldots, n\}$, party $P_i$ has his/her own private key pair $(a_i, b_i)$ such that $a_i, b_i \in Z_p$ , $\gcd(a_i, p-1) = 1$, and $a_i b_i = 1 \pmod{p-1}$. $Q$ is a proposition defined as follows.

$$Q = \begin{cases} a_0 a_1 \neq 1 \pmod{p-1}, & \textit{and} \\ a_0 a_1 a_2 \neq 1 \pmod{p-1}, & \textit{and} \\ \cdots, & \textit{and} \\ a_0 a_1 \cdots a_n \neq 1 \pmod{p-1}. & \end{cases}$$

**Objectives:**

1. Check and make sure that proposition $Q$ is true. If $Q$ is false, try to change some private keys securely until $Q$ becomes true.
2. During the checking of $Q$ and changing of private keys, the privacy of all private key pairs should be maintained.

Our patch protocol is given below.

**Our SHK Patch Protocol**
**Begin**

1. $P_0$ chooses a random number $r$ in $Z_p$ and compute $l_0 = r^{a_0}$ .
2. For $i = 1, 2, \ldots, n$ do
   {
   $P_0$ sends $l_{i-1}$ to $P_i$.
   $P_i$ replies $l_i = l_{i-1}^{a_i}$ to $P_0$.
   $P_0$ stores $l_i$ to $A[i]$, the $i$-th cell of the array $A$.
   }
3. $P_0$ checks whether there is an $i$ such that $A[i] = r$.
   If there is no $i$ such that $A[i] = r$, the protocol stops and $Q$ is true,
   else
   {
   $P_0$ selects a new private key pair $(a_0^*, b_0^*)$.
   $P_0$ recomputes $A[i] \leftarrow (A[i])^{b_0 a_0^*} \bmod p$ , for each $i$.
   $P_0$ updates the private key pair $(a_0, b_0) \leftarrow (a_0^*, b_0^*)$.
   Go to Step 3 to recheck.

}

**End**

Figure 1 is a diagram to illustrate the communication flow among the parties. In addition, the contents of the array of $P_0$ are also given in Figure 2. When no element of the array is equal to the random number initially selected by $P_0$, the proposition $Q$ must be true. Thus the correctness of the patch protocol is straight-forward. In the next section, we will give a brief analysis for the security and efficiency of our patch protocol.

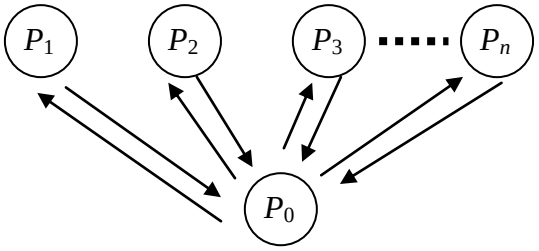


Figure 1: The communication flow among the parties.

| From $P_1$ | From $P_2$ | From $P_3$ | … | From $P_n$ |
|---|---|---|---|---|
| $l_1 = r^{a_0 a_1}$ | $l_2 = l_1^{a_2}$ | $l_3 = l_2^{a_3}$ | … | $l_n = l_{n-1}^{a_n}$ |

Figure 2: The contents of the array of $P_0$.

## IV. Security and Efficiency

**Security:**

The security of our patch protocol relies on the computational hardness of discrete logarithm problem (DLP). For an adversary, even partial break to find out the private key ($a_i$ or equivalently $b_i$) of one party is not easy.

**Efficiency:**

The efficiency can be discussed from two kinds of cost: the communications cost and the computation cost. The communication cost is low. Except that party $P_0$ needs to send out $n$ messages, each party needs to send out one message only. The computation cost of our patch protocol mainly depends on the number of times to execute Step 3. And this number is always equal to one plus the number of times for $P_0$ to reselect a

new private key pair. Let $T$ be the expected number of times for $P_0$ to re-select his key pair (The initial selection of his key pair is not included). Consider the set $S = \{(a_1)^{-1}(\text{mod } p–1), (a_1a_2)^{-1}(\text{mod } p–1), \ldots, (a_1a_2\ldots a_n)^{-1}(\text{mod } p–1)\}$. If $a_0$ is not in $S$, proposition $Q$ must be true. Since the strategy we use is to fix $(a_i, b_i)$ for $i \in \{1, 2, \ldots, n\}$ and randomly re-choose $a_0$ (or equivalently $b_0$) in $Z_{p-1}^*$ when necessary, the probability Pr{$Q$ is false} is at most $|S| / |Z_{p-1}^*| \le n / \phi(p-1)$, where $\phi$ is the Euler function. Let $q = n / \phi(p-1)$, this should be a very small value in practical settings. We thus have the following inequalities.

$$\Pr\{Q \text{ is true}\} \ge 1 - q,$$
$$\Pr\{Q \text{ is false}\} \le q,$$
$$T \le 0(1-q) + 1q(1-q) + 2q^2(1-q) + 3q^3(1-q) + \cdots = \frac{q}{1-q} = \frac{n}{\phi(p-1)-n}.$$

Since in practical situations $\phi(p-1)$ is far greater than $n$, so $T$ is far less than 1. That is, the expected number of times to re-select the private key pair of $P_0$ in our protocol is far less than one. Therefore, except $P_0$, each party needs one modular exponentiation only. The expected number of modular exponentiations needed by $P_0$ is $1 + 2n\text{T} = 1 + 2n^2 / (\phi(p-1) - n)$, which is very close to one. This is very efficient.

## V. Conclusion

In this paper, we proposed a patch protocol for the SHK secret transfer scheme to overcome the possible secret leakage problem. The patch protocol can be used with/before the original SHK protocol. From another point of view, our patch protocol checks a proposition defined over private keys of some parties and coordinates to change some key until the proposition is true. The privacy of all keys is always maintained during the checking of proposition and the coordination of key changing. In conclusion, our patch protocol has the following advantages.

1. It prevents the secret leakage problem in the locking phase of the SHK scheme.
2. The privacy of all private keys is maintained.
3. The communication cost is low. Except that party $P_0$ needs to send out $n$ messages, each party needs to send out one message only.
4. The computation cost is also low. Except $P_0$, each party needs one modular exponentiation only. The expected number of modular exponentiations needed by $P_0$ is $1 + 2n^2 / (\phi(p-1) - n)$, which is very close to one.

## References


[1] Agrawal, R., Evfimievski, A., and Srikant, R., “Information sharing across private databases,” International Conference on Management of Data (ACM SIGMOD), ACM Press, pp. 86-97, 2003.

[2] Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., and Zhu, M. Y., “ Tools for privacy preserving distributed data mining,” SIGKDD Explorations 4, 2, pp. 28-34, Jan. 2003.

[3] Menezes, A., Oorschot, P., Vanstone, S., “Handbook of Applied Cryptography,” CRC Press, pp. 500, 1997.

[4] Shamir, A., “How to share a secret,” Communications of the ACM, Vol. 22, No. 11, pp. 612-613, Nov. 1979.

[5] Shamir, A., “On the power of commutativity in cryptography,” ICALP80, pp.582-595, July 1980.

[6] Weis, Stephen A., “New Foundations for Efficient Authentication, Commutative Cryptography, and Private Disjointness Testing,” MIT Phd Dissertation, pp.1, May 2006.

[7] Saied Hosseini Khayat, “Using commutative encryption to share a secret,” http://eprint.iacr.org/2008/356.pdf, Aug. 2008.