A Lightweight Key Agreement Protocol Based on LPN Problem

Dang Nguyen Duc* and Kwangjo Kim*

*International Research Center for Information Security (IRIS)
Information and Communications University (ICU)

Abstract

In this short paper, we present a 2-party lightweight key agreement protocol based on a hard learning problem called Learning Parity with Noise (LPN for short) problem. Our protocol requires only basic Boolean operations and random number generation which is very suitable for low computational devices like sensor nodes and RFID tags.

I. Introduction

With the blossom of a wide range of low-cost (and low computational) devices, the need for efficient cryptographic protocols also rises. To meet this demand, the first approach one can consider is to seek for efficient implementation of current cryptographic protocols so that they can be implemented for low-cost devices. There are several shortcomings in this approach. First of all, it is not easy and even impossible in case of extremely low-cost devices like passive RFID tags. Secondly, if efficient implementation exists, such implementation might compromise security of the protocols. The second approach is to design new cryptographic protocols with efficiency constraint in mind. Although the second approach might result in less secure protocols comparing to current ones, it is arguably preferable as it allows designers to aim for a good balance between security and efficiency from the beginning.

A typical example for designing new cryptographic protocols for low-cost devices is security protocols RFID (Radio Frequency Identification) tags. An RFID tag is a tiny and extremely cheap computational device capable of short-range wireless communication. Because of its limited computational resource, an typical RFID tag can only perform basic Boolean operations, generate pseudo-random numbers possibly compute cryptographic hash operations. Given such tight constraints, it seems impossible to use cryptographic protocols employing block ciphers or public key cryptography for RFID tags. Therefore, many new cryptographic protocols for RFID devices have been proposed [9,10,11,12,13]. These protocols use only XOR, pseudo-random number generation and hash functions and therefore are well suited for RFID tags. One problem with those protocols are their lack of a security foundation, especially considering the state-of-the-art attacks on cryptographic hash functions. We think that lightweight cryptographic protocols also need a good security foundation so that rigorous security analysis could be achieved. A perfect example for this kind of protocols is the HB+ authentication protocol by Juels and Weis [14]. HB+ not only is very efficient (and does not require cryptographic hash function) but also
bases its security on a well-studied hard problem called Learning Parity in the Presence of Noise (LPN for short). The LPN problem has been shown to be NP-Complete and finds its cryptographic applications earlier due to the work of Hopper and Blum [7]. In [7], Hopper and Blum presented an human authentication protocol and proved its security based on the LPN problem. Indeed, HB+ is an enhanced version of HB.

Our contribution. We find that LPN problem is a very good foundation for lightweight protocol. In this paper, we present another application of the problem by presenting a lightweight key exchange protocol without the need for public key cryptography, block cipher and cryptographic hash function.

II. Previous Works

The LPN problem involves binary inner product of two k-bit numbers. The operation is defined as follows: given two k-bit number $a = (a_{k-1} ... a_1)$ and $x = (x_{k-1} ... x_1)$, the binary inner product of $a$ and $x$, denoted as $a \cdot x$ is computed as follows: $a \cdot x = (a_k \land x_k) \oplus (a_{k-1} \land x_{k-1}) \oplus ... \oplus (a_1 \land x_1)$. Clearly, this operation can be easily implemented in cheap hardware. Furthermore, as noted by Juels and Wels [14], there is no need to buffer all k bits of $a$ and $x$ at once when evaluating $a \cdot x$. Therefore, memory requirement for this operation is also very low.

The first cryptographic significance of binary inner product is due to Goldreich and Levin [18]. They proved that $a \cdot x$ is unpredictable if only either $a$ or $x$ is given. This result was subsequently used to construct a secure pseudo-random number generator (though not practical).

The first practical application related to binary inner product was introduced by Hopper and Blum [7]. They presented a human authentication protocol such that the human only needs to evaluate one binary inner product operation, and generate a random bit. The protocol is called HB and is shown to be provably secure under the assumption a so-called Learning Parity with Noise (LPN for short) problem is intractable. To better illustrate the LPN problem, we now describe the HB protocol. In the HB protocol, the human (denoted as $H$, also called the prover) and a machine (denoted as $C$, also called the verifier) share a secret $x$ of $k$-bit long. The protocol consists of several executions of a basic challenge-response protocol which is described in Fig. 1.

![Fig. 1. HB Protocol](image)

<table>
<thead>
<tr>
<th>Human($z, \eta$)</th>
<th>Machine($z, \eta$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v \leftarrow \text{Bern}_1$</td>
<td>$z \leftarrow \text{Select challenge } a \in {0, 1}^k$</td>
</tr>
<tr>
<td>Compute response $z = (a \cdot x) \oplus \eta$</td>
<td>$\text{Check if } z = a \cdot x$</td>
</tr>
</tbody>
</table>

Fig. 1. HB Protocol

Bern. denotes Bernoulli distribution with expected value $\eta$, where $\eta$ is in $(0, 1/\delta)$ range (that is the bit $\eta$ - known as noise bit - is generated independently for each protocol round and equals 1 with probability $\eta$). The purpose of $\eta$ is to prevent eavesdropping adversaries from extracting the secret $x$ after observing $k$ pairs $(a, z)$. The machine accepts the human after, say $r$ rounds of the above protocol if and only if human produces roughly $r/\delta$ incorrect responses.

It is straightforward that HB protocol is secure only if an eavesdropper observing messages exchanged between $H$ and $C$ has a negligible chance of impersonating $H$. More specifically, an eavesdropper $A$ obtains $r$ pairs $(a, z)$ and tries to deduce a $k$-bit number $x'$ such that using $x'$ to carry out HB protocol, $A$ would get accepted by $C$. The
problem of finding such $x'$ is called Learning Parity with Noise problem (LPN). However, as noted by Katz and Shin in [24], finding $x'$ is essentially equivalent to finding $x$ itself.

The LPN problem has been extensively studied in several research works including [4,5,6]. Those results show that LPN problem is very likely an intractable problem. To solve LPN problem as mentioned above, the best known algorithm by Blum et al. has sub-exponential complexity of $2^{0.302n}$.

III. Lightweight Key Agreement Protocol based on LPN Problem

Our Protocol. Let's assume that two entities $A$ and $B$ wish to establish a common secret key over an insecure channel. We aim to design a protocol with both security and efficiency in mind so that even low-cost $A$ and $B$ can implement the protocol. We present such a protocol whose security is based on the LPN problem. First, we observe that, given the pair $(α, z = (α \cdot x) \oplus u)$ where $x$ is secret and $(α, z)$ constitutes an instance of the LPN problem, then $(α, z)$ is an encrypted message of the noise bit $u$. As Katz and Shin showed in [17], $(α, z)$ is a pseudo-random bit string. Therefore, the above encryption scheme is semantically secure. Using this encryption scheme, $A$ can securely transport 1 bit to $B$ and vice versa. Then, one bit of the shared key can be computed by XORing two communicated bits. However, this trivial protocol is not secure against replay attack. To prevent replay attack and other more complicated attacks, we must use nonce when transport key material as well as add key confirmation. To do so, we borrow ideas of the HB+ authentication protocol [14]. Similar to HB+, two entities in our key exchange protocol also share two $k$-bit secrets, say $s_1$ and $s_2$. The protocol proceeds as follows:

- $A \rightarrow B$: $A$ sends $(α, z_A)$ to $B$ where $α$ in $(0, 1)^k$ and $z_A = (α \cdot s_1) \oplus u_A$ with $u_A$ is a randomly chosen bit.
- $B \rightarrow A$: $B$ replies with $(b, z_B)$ such that $b$ in $(0, 1)^k$ and $z_B = (b \cdot s_1) \oplus (α \cdot s_2) \oplus u_A \oplus u_B$ with $u_B$ is a randomly chosen bit.
- $A \rightarrow B$: if $u_A = u_B$ $A$ sends a key confirmation message $c = (α \cdot s_2) \oplus (b \cdot s_1)$. Otherwise, it sends $c = (b \cdot s_1) \oplus (α \cdot s_2)$.
- $B$: upon receiving $c$, $B$ verifies that $c$ either equals $(α \cdot s_1) \oplus (b \cdot s_2)$ or $(b \cdot s_1) \oplus (α \cdot s_2)$ if $u_A = u_B$ or otherwise, respectively.
- $A$, $B$: the two parties compute 1 bit of shared secret as $(s_1 \cdot s_2) \oplus u_A \oplus u_B$.

In the above protocol, at first $A$ securely sends its contribution to shared secret, $u_A$, to $B$. $B$ not only sends back its contribution $u_B$ to $A$ but also incorporates $α$ and $u_A$ into its message to prevent reflection attack. To prevent unknown key share attack, $A$ needs to send a key confirmation message $c$. In addition, the key derivation function needs to incorporate $u_A, u_B$ and both pre-shared secret $(s_1, s_2)$.

Comparison with other Protocols. There are many key agreement protocols known in cryptographic literature including Diffie-Hellman, Matsumoto-Takashima-Imai and Menezes-Qu-Vanstone protocols. All of these protocols are based on public key cryptography and therefore require significant computational resources. In contrast, our protocol requires only basic Boolean operations and pseudorandom number generation. Unfortunately, our protocol is not good at bandwidth utilization as it can only allow two parties to share 1 bit for 1 round. However, we think this property is suitable for low-cost devices without any
state-of-the-art block cipher implementation and the need to communicate a large amount of data. In case of low-cost devices, they usually needs to communicate a short amount of data and combining our key agreement protocol with one-time pad encryption scheme should be a appealing solution.

IV. Concluding Remarks

In this paper, we have presented HB+ protocol, an augmented version of HB+ protocol which can prevent the man-in-the-middle attack described in [15]. Our protocol can be seen as a combination of two instances of the HB and HB+ protocols. Comparing with the HB+ protocol, our protocol requires one more additional secret, two more binary inner product computation and one more bit to transfer by the reader. Therefore, HB+ can still be useful for tightly resource-constrained devices like RFID tags and sensor nodes.

References


