Toward Wireless Security without Computational Assumptions—Oblivious Transfer Based on Wireless Channel Characteristics

Zhuo Hao, Yunlong Mao, Sheng Zhong, Li Erran Li, Haifan Yao, and Nenghai Yu

Abstract—Wireless security has been an active research area since the last decade. A lot of studies of wireless security use cryptographic tools, but traditional cryptographic tools are normally based on computational assumptions, which may turn out to be invalid in the future. Consequently, it is very desirable to build cryptographic tools that do not rely on computational assumptions. In this paper, we focus on a crucial cryptographic tool, namely 1-out-of-2 oblivious transfer. This tool plays a central role in cryptography because we can build a cryptographic protocol for any polynomial-time computable function using this tool. We present a novel 1-out-of-2 oblivious transfer protocol based on wireless channel characteristics, which does not rely on any computational assumption. We also illustrate the potential broad applications of this protocol by giving two applications, one on private communications and the other on privacy preserving password verification. We have fully implemented this protocol on wireless devices and conducted experiments in real environments to evaluate the protocol. Our experimental results demonstrate that it has reasonable efficiency.

Index Terms—Oblivious transfer, physical channel characteristics, security

✦

1 INTRODUCTION

Wireless security has been an active research area since the last decade. A lot of studies of wireless security use cryptographic tools such as encryption, authentication, and key agreement in order to achieve security protection. These traditional cryptographic tools are very powerful, but most of them have a common weakness—normally, they are based on computational assumptions.

For example, consider one of the most frequently used cryptographic tools, symmetric key encryption. We have a number of very good existing encryption schemes, e.g., AES [1]. However, when we use AES to encrypt a message, we are actually making an implicit assumption: the AES block cipher is a psedorandom permutation. Intuitively, this assumption implies that it is infeasible for an adversary to find the cleartext message from the ciphertext. Nevertheless, the above assumption of pseudorandomness is based on the cryptologists' understanding of the *current* attacks on encryption schemes. It is possible that, in the future (maybe even in the near future), the AES scheme will be broken by newly invented cryptanalysis techniques.

In fact, there was a lesson a few years ago, when cryptologists broke several famous hash functions, including MD5 and SHA-0 [2], [3]. To be more precise, these hash functions had been assumed to be collision-resistant for more than ten years, but cryptologists found that these assumptions are invalid and there are quite efficient algorithms to find collisions of these hash functions. It is worth noting that the above discoveries were made after the hash functions became either national standards or de facto standards. Hence, it will be very desirable if we can remove cryptographic tools' dependence on such computational assumptions.

Of course, removing computational assumptions from the cryptographic tools, and thus from the wireless security systems, is a highly challenging problem. Consequently, in this paper, we do not intend to build a complete wireless security system that does not rely on computational assumptions. Instead, we would like to address a fundamental question as a crucial step towards solving this very challenging problem: Is it at all feasible to build wireless security systems without relying on computational assumptions?

Our answer to the above question is positive. Specifically, we propose that wireless security can be based on the physical channel characteristics rather than computational assumptions, as illustrated by a new type of protocols for key agreement in wireless networks [4]–[10].[1] In other words, the wireless channel characteristics can be used not only to achieve key agreement, but also to establish *any* cryptographic tool.

To be more precise, we use wireless channel characteristics to build a crucial cryptographic tool called 1-out-of-2 oblivious transfer. (For simplicity, hereafter we use OT to refer to oblivious

- *Z. Hao and N. Yu are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China. E-mail: hzhuo@mail.ustc.edu.cn, ynh@ustc.edu.cn.*
- *Y. Mao and S. Zhong are with the State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China. E-mail: zhongsheng@nju.edu.cn.*
- *L. E. Li is with the Networking Research Laboratory–Bell Labs, Alcatel-Lucent, Berkeley Height, NJ. E-mail: erranlli@research.bell-labs.com.*
- *H. Yao is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY. E-mail: haifanya@buffalo.edu.*

*Manuscript received 05 Dec. 2011; revised 15 Nov. 2012; accepted 14 Jan. 2013. Date of publication 14 Feb. 2013; date of current version 09 June 2014. Recommended for acceptance by Y. Yang
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2013.27*

---

[1] This is not the only way to do cryptographic operations without computational assumptions; quantum communications do not rely on computational assumptions as well. But quantum communications are out of the scope of this paper.

transfer, and use OT_1^2 to refer to 1-out-of-2 oblivious transfer.) The reason for choosing to work on OT_1^2 is that it plays a central role in cryptography. In fact, Kilian [11] has proved that OT_1^2 is "complete", meaning that for any polynomial-time computable function, we can build a cryptographic protocol using OT_1^2. For example, electronic voting protocols, anonymous communications protocols, digital cash protocols, privacy preserving data mining protocols, etc. can all be built using OT_1^2. Hence, once we get an OT_1^2 protocol independent of computational assumptions, we can actually use it to establish other cryptographic protocols independent of computational assumptions.

However, it is not easy to construct an OT_1^2 protocol based on wireless channel characteristics. The main idea underlying our work is to employ a novel technique from [12]. We point out that both our channel model and our OT_1^2 protocol are significantly different from those of [12]. Consequently, our use of their technique is non-trivial.

To illustrate the potential wide applications of our work, we give a method of private communications based on our OT_1^2 protocol. Just like traditional symmetric key encryption schemes, this method allows two wireless devices that have a common secret key to communicate with each other privately. Nevertheless, the security of this method depends on wireless channel characteristics, not on computational assumptions.

The other application of our OT_1^2 protocol is privacy preserving password verification. Using the method we present, one wireless device can verify a password from another wireless device in such a way that the password is not revealed to either the former device or any eavesdropper.

In summary, we have the following contributions in this paper.

- We are the first to construct an OT_1^2 protocol based on the physical characteristics of wireless channels. Our OT_1^2 protocol does not rely on any computational assumptions. Given the completeness of OT_1^2 proved by Kilian [11], our work can be considered a crucial step towards building strong wireless security systems without computational assumptions.
- Our OT_1^2 protocol has wide potential applications. In particular, we have given a method of private communications and a method of privacy preserving password verification based on our own OT_1^2 protocol.
- We have *completely* implemented our OT_1^2 protocol on *real, mobile* wireless devices, and evaluated it through extensive experiments. Our experimental results demonstrate that our OT_1^2 protocol has reasonable efficiency.

The rest of this paper is organized as follows. In Section 2, we present technical preliminaries. In Section 3, we design and analyze our OT_1^2 protocol. In Sections 4 and 5, we show the two applications of our OT_1^2 protocol. The implementation and experiments are described in Section 6. After briefly reviewing related work in Section 7, we conclude in Section 8.

2 TECHNICAL PRELIMINARIES

Throughout this paper, we follow the formulation presented in [8], [13]. For completeness, we briefly review the model of wireless channels and the quantization method in [8] and refer readers to [8] for more details. After that, we specify the requirements that an OT_1^2 protocol needs to satisfy and the security model we use to analyze OT_1^2 and its applications.

2.1 Model of Wireless Channel

Consider two parties A and B, and the wireless channel between them. Just like in [8], for ease of presentation, let h be the magnitude of the in-phase component of the Rayleigh fading process, which follows a Gaussian distribution. (Note that our protocol and analysis do not rely on this assumption of distribution. In fact, they can be easily extended to the general case; but the extension is notationally complex and less easy to understand.) Clearly, h can be viewed as a stochastic process; we use $h(t)$ to represent the value of h at time t.

A and B do not know the precise values of $h(t)$; they can only make estimates. Specifically, let $s(t)$ be a well known probe signal. Suppose that B sends a probe signal and A receives it at time t_1; A sends a probe signal and B receives it at time t_2. Then A and B can estimate the channel respectively, using their received signals. In this case, the signals A and B receive can be expressed as follows:

$$r_a(t_1) = h(t_1)s(t_1) + n_a(t_1), \tag{1}$$

$$r_b(t_2) = h(t_2)s(t_2) + n_b(t_2), \tag{2}$$

where $n_a(t_1)$ and $n_b(t_2)$ are the receiver noises at A and B.

By using existing techniques of channel estimation, e.g., [14], A (resp., B) can obtain an estimate $\hat{h}_a(t_1)$ (resp., $\hat{h}_b(t_2)$) from $r_a(t_1)$ (resp., $r_b(t_2)$). These estimates satisfy the following equations:

$$\hat{h}_a(t_1) = h(t_1) + z_a(t_1), \tag{3}$$

$$\hat{h}_b(t_2) = h(t_2) + z_b(t_2), \tag{4}$$

where $z_a(t_1)$ (resp., $z_b(t_2)$) represents the noise and interferences caused by $n_a(t_1)$ (resp., $n_b(t_2)$) during the process of channel estimation.

By the channel reciprocity[2], we can guarantee that $h(t_1)$ and $h(t_2)$ are correlated, if $t_2 - t_1$ is small in the above probe and estimation process. More precisely, we need that the pair of probe signals exchanged by A and B is within the *coherence time* [15], [8] of the wireless channel. Here the coherence time T_C is typically inversely proportional to the maximum Doppler frequency f_m [15], [8]:

$$T_C \approx \frac{1}{f_m} = \frac{\lambda}{v}. \tag{5}$$

In equation (5), λ is the wavelength of the carrier signal, and v is the maximum moving speed of objects in the environment.

Note that the above description refers to the exchange of one single pair of probe signals. As we will see, our OT_1^2 protocol actually requires exchanges of multiple pairs of probe signals. Unlike the short time interval between the two probe signals in the same pair, the time interval between any two different pairs of probe signals is chosen to be larger than the coherence time. In this way, the channel estimates derived from different pairs of probe signals can be seen as independent from each other.

2.2 Method of Quantization

When A and B have obtained their estimates $\hat{h}_a$ and $\hat{h}_b$, respectively, they quantize these channel estimates into bit

---

[2] If the involved wireless devices are not calibrated, methods similar to [8] can be used to reduce the problem introduced by the lack of calibration.

strings using a quantization function Q. The function Q is defined as follows:

$$Q(x) = \begin{cases} 1 & if\, x > q_+ \\ 0 & if\, x < q_-. \end{cases} \tag{6}$$

where q_+ and q_- are derived from the mean and standard deviation of channel estimates. Denote the mean by μ and the standard deviation by σ. Let α ($\alpha > 0$) be a system parameter. We have

$$q_{+,-} = \mu \pm \alpha \cdot \sigma. \tag{7}$$

2.3 Requirements for OT_1^2 and Security Model

Our main objective in this paper is to build an OT_1^2 protocol between A and B. In Section 3, we describe how to build this protocol, including how to use the method of quantization mentioned above. Before we build the OT_1^2 protocol, we need to first list the requirements for OT_1^2.

Assume that A has two bits b_0 and b_1 as her input, and that B has a bit s as his input. The requirements for an OT_1^2 protocol are that, when the protocol terminates,

1) B gets the bit b_s;
2) B gets no information about b_{1-s};
3) A gets no information about s.

Throughout this paper, we analyze the security of OT_1^2 and its applications in the semi-honest model, which is one of the standard security models [16]. In this model, each involved party follows the protocol, but they may be curious in learning private information that they are not supposed to learn. Furthermore, eavesdropping by outsiders (i.e., parties not supposed to participate in the protocol) are allowed in our model.

A critical assumption we use throughout this paper is that *all involved parties can see the quantized channel estimates only*. The original measurements of channels must remain confidential. In practice, such a requirement can be fulfilled by, e.g., a piece of tamper-proof hardware attached to each involved device. We leave to future study the case in which original channel measurements can be seen by the involved parties.

3 OT_1^2 Based on Wireless Channel Characteristics

Using the probing, estimation, and quantization process described in Section 2, now we design an OT_1^2 protocol and analyze it.

3.1 The OT_1^2 Protocol

Our OT_1^2 protocol consists of two stages. In the first stage, the two parties send multiple probe signals to each other alternately, estimate the channel, and convert the estimates into bits, using the quantization method described in Section 2. (Recall that the time interval between each pair of probe signals is within the coherence time, but the time interval between any two different pairs of probe signals is more than the coherence time.) The two parties terminate the first stage as soon as each of them has obtained at least N bits, where N is an even number and a system parameter, with a typical value of 100–300.

The main idea of the second stage is that A can xor her two secret bits with two sequences of masks respectively and then send the results to B. In order to guarantee that B gets only b_s but not b_{1-s}, we only need to make sure that the sequence of masks for b_s is known to B, but the other sequence is unknown to B. To achieve this objective, we have the following crucial observation:[3] Consider two pairs of probe signals such that A extracts the same bit from them using the quantization method in Section 2. From these two pairs of probe signals, if B also extracts the same bit, then it is very likely that the bit extracted by A is equal to the bit extracted by B. In contrast, if from the two pairs of probe signals B extracts two different bits, then B has no idea about what bit is extracted by A. Consequently, for both sequences of masks, we let A use bits extracted from probe signals by A such that the next extracted bits are the same. In order to ensure the sequence of masks for b_s is known to B, we make sure that the masks for b_s correspond to those bits extracted by B that are identical to their next bits. In order to ensure the sequence of masks for b_{1-s} is unknown to B, we make sure that the masks for b_{1-s} correspond to those bits extracted by B that are not identical to their next bits.

More details of the second stage are given below.

Suppose that, at the end of the first stage, A has obtained N bits from the quantized channel estimates: $\{BS_a(i)_{i=1,2,\ldots,N}\}$; B has also obtained N bits from the quantized channel estimates: $\{BS_b(i)\}_{i=1,2,\ldots,N}$. (Note that we use $BS_a(i)$ to denote the ith term in the sequence BS_a. Similar notations are used throughout the paper.) The second stage can be divided into four steps.

Step 1. A generates an index sequence I by extracting all index i such that $BS_a(2i-1) = BS_a(2i)$ ($i \in [1, N/2]$). A sends I to B using a reliable communication protocol, e.g., TCP. Note that, throughout this OT_1^2 protocol, communications using this reliable communication protocol need *not* to be encrypted.

Step 2. After B receives the index sequence I from A, B generates two disjoint index sequences I_s and I_{1-s}, where I_s is subject to the following constraints:

(1) $|I_s| = n$ (n is a security parameter, with a typical value of $10 \sim 30$), i.e., there are exactly n indices in the sequence I_s;
(2) $I_s \subseteq I$, i.e., I_s is a subsequence of I;
(3) for all $i \in I_s$, $BS_b(2i-1) = BS_b(2i)$;

and I_{1-s} is subject to the following constraints:

(1) $|I_{1-s}| = n$;
(2) $I_{1-s} \subseteq I$;
(3) for all $i \in I_{1-s}$, $BS_b(2i-1) \neq BS_b(2i)$.

Note that the above two sequences I_s and I_{1-s} can also be called I_0 and I_1 (not necessarily in the same order), because the value of s is either 0 or 1. Clearly, when $s = 0$, the sequence I_s is I_0 and the sequence I_{1-s} is I_1; when $s = 1$, the sequence I_s is I_1 and the sequence I_{1-s} is I_0.

Then B sends the two index sequences I_0 and I_1 to A, using a reliable communication protocol.

Step 3. Once A receives I_0 and I_1 from B, A generates two sequences L_0 and L_1 as follows: for each $e \in \{0, 1\}$ and each j such that $1 \leq j \leq n$,

$$L_e(j) = b_e \oplus BS_a(2 \cdot I_e(j)).$$

Then A sends L_0 and L_1 to B using a reliable communication protocol.

In this step, the variable b_e represents b_0 or b_1, depending on the value of e. Clearly, when $e = 0$, b_e is b_0; when $e = 1$, b_e is b_1.

---

[3] This observation is valid under the condition that the time interval between the two pairs of probe signals is more than the coherence time. Recall this condition is satisfied by our OT_1^2 protocol.

It is worth noting that the two variables b_0 and b_1 considered here are exactly the two variables b_s and b_{1-s} considered earlier (not necessarily in the same order). When $s = 0$, b_s is b_0 and b_{1-s} is b_1; when $s = 1$, b_s is b_1 and b_{1-s} is b_0.

Step 4. After B receives the L_0 and L_1 from A, B computes b'_s using the following formula:

$$b'_s = \mathsf{majority}(\{L_s(j) \oplus BS_b(2 \cdot I_s(j)), j \in [1, n]\}).$$

In the above equation, L_s represents L_0 or L_1, depending on the value of s. When $s = 0$, L_s is L_0; when $s = 1$, L_s is L_1. Since we have no idea whether $s = 0$ or 1, we have to write the formula in the general form, with L_s instead of L_0 or L_1.

When this step is completed, b'_s is supposed to be equal to b_s, the value B needs to obtain. (In Section 3.2, we prove there is a high probability that $b'_s = b_s$.)

A formal description of the second stage is shown in Algorithm 1.

Algorithm 1 Second Stage of OT_1^2 Protocol

Input: $\{BS_a(i)\}_{i=1,2,\ldots,N}$ and $\{BS_b(i)\}_{i=1,2,\ldots,N}$ A's secret bits $\{b_0, b_1\}$, B's secret bit s

Output: B outputs b'_s as an estimate of his chosen b_s

```
$A$:
$I \leftarrow$ empty sequence
foreach $i \in [1, N/2]$ do
    if $BS_a(2i-1) = BS_a(2i)$ then
        add $i$ into $I$
$A$ sends $I$ to $B$
```

```
$B$:
$I_s \leftarrow$ empty sequence,$I_{1-s} \leftarrow$ empty sequence
foreach $i \in I$ do
    if $BS_b(2i-1) = BS_b(2i)$ and $|I_s| < n$ then
        add $i$ into $I_s$
    else if $BS_b(2i-1) \neq BS_b(2i)$ and $|I_{1-s}| < n$ then
        add $i$ into $I_{1-s}$
    if $|I_s| = n$ and $|I_{1-s}| = n$ then
        break
$B$ sends $I_s$ and $I_{1-s}$ to $A$
```

```
$A$:
$L_0 \leftarrow$ empty sequence, $L_1 \leftarrow$ empty sequence
foreach $e \in \{0, 1\}$ do
    foreach $j \in [1, n]$ do
        $L_e(j) = b_e \oplus BS_a(2 \cdot I_e(j))$
$A$ sends $L_0$ and $L_1$ to $B$.
```

```
$B$:
$b'_s = \mathsf{majority}(\{L_s(j) \oplus BS_b(2 \cdot I_s(j)), j \in [1, n]\})$.
```

3.2 Protocol Analysis

Below we present an analysis to show that the three requirements for OT_1^2 are all satisfied by our protocol. The analysis is based on the semi-honest model and under the assumption that the eavesdropper is passive. We also assume that the eavesdropper is more than half a wavelength away from both A and B [6], [8], so that the fading channel he experiences is statistically independent from the fading channel between A and B. In the settings we consider, the wavelength is pretty short and thus it is very unlikely for two entities to have a distance shorter than half a wavelength. For example, in [8], Mathur, et al. notice that, "at 2.4 GHz, we only require that Eve be roughly $\lambda/2 = 6.25$ cm away from Alice and Bob to ensure that she gets no useful information." It is highly unlikely that an eavesdropper can be just a few centimeters away from a protocol participant without being detected.

Theorem 1. *Under the standard assumptions [4], [8], [17] that the stochastic process* $\mathbf{h}$ *is stationary and that* $h(t)$ *is a Gaussian random variable, when our* OT_1^2 *protocol is finished, for any* $\epsilon > 0$, *B gets b_s with probability* $1 - \epsilon$ *as long as* $n \geq \frac{ln(\frac{1}{\epsilon})}{2(q-\frac{1}{2})^2}$, *where for any* $i \in I_s$, $\mathsf{Pr}[BS_b(2i-1, 2i) = BS_a(2i-1, 2i) | BS_a(2i) = BS_a(2i-1)] = q > \frac{1}{2}$.

Proof. For any $i \in I_s$, denote by $\mathbf{x_1}$ and $\mathbf{x_2}$ two vectors of channel estimates corresponding to $[BS_b(2i-1), BS_a(2i-1), BS_b(2i), BS_a(2i)]^T$ and $[BS_a(2i-1), BS_a(2i)]^T$, respectively. Since $\mathbf{h}$ is a stationary Gaussian process, $\mathbf{x_1}$ and $\mathbf{x_2}$ are both random vectors following multivariate Gaussian distributions. Now we consider the following probability. For each $i \in I_s$, we have:

$$\begin{aligned}
&\mathsf{Pr}[BS_b(2i-1, 2i) = \text{“}11\text{”} | BS_a(2i-1, 2i) = \text{“}11\text{”}] \\
&= \frac{\mathsf{Pr}[BS_b(2i-1, 2i) = \text{“}11\text{”}, BS_a(2i-1, 2i) = \text{“}11\text{”}]}{\mathsf{Pr}[BS_a(2i-1, 2i) = \text{“}11\text{”}]} \\
&= \left(\int_{q_+}^{+\infty} \int_{q_+}^{+\infty} \int_{q_+}^{+\infty} \int_{q_+}^{+\infty} \frac{1}{(2\pi)^2 |\mathsf{Cov}_{4,4}(\mathbf{x_1})|^{1/2}} \right. \\
&\quad \left. \times \exp\left\{ -\frac{1}{2} (\mathbf{x_1} - \boldsymbol{\mu}_1)^T \cdot \mathsf{Cov}_{4,4}^{-1}(\mathbf{x_1}) \cdot (\mathbf{x_1} - \boldsymbol{\mu}_1) \right\} d^{(4)}x \right) / \\
&\left(\int_{q_+}^{+\infty} \int_{q_+}^{+\infty} \frac{1}{(2\pi) |\mathsf{Cov}_{2,2}(\mathbf{x_2})|^{1/2}} \right. \\
&\quad \left. \times \exp\left\{ -\frac{1}{2} (\mathbf{x_2} - \boldsymbol{\mu}_2)^T \cdot \mathsf{Cov}_{2,2}^{-1}(\mathbf{x_2}) \cdot (\mathbf{x_2} - \boldsymbol{\mu_2}) \right\} d^{(2)}x \right)
\end{aligned}$$

In the above equation, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the expectation vectors of $\mathbf{x_1}$ and $\mathbf{x_2}$; $\mathsf{Cov}_{4,4}(\mathbf{x_1})$ and $\mathsf{Cov}_{2,2}(\mathbf{x_2})$ are the covariance matrices of random vectors $\mathbf{x_1}$ and $\mathbf{x_2}$. Similarly,

$$\begin{aligned}
&\mathsf{Pr}[BS_b(2i-1, 2i) = \text{“}00\text{”} | BS_a(2i-1, 2i) = \text{“}00\text{”}] \\
&= \left(\int_{-\infty}^{q_-} \int_{-\infty}^{q_-} \int_{-\infty}^{q_-} \int_{-\infty}^{q_-} \frac{1}{(2\pi)^2 |\mathsf{Cov}_{4,4}(\mathbf{x_1})|^{1/2}} \right. \\
&\quad \left. \times \exp\left\{ -\frac{1}{2} (\mathbf{x_1} - \boldsymbol{\mu}_1)^T \cdot \mathsf{Cov}_{4,4}^{-1}(\mathbf{x_1}) \cdot (\mathbf{x_1} - \boldsymbol{\mu}_1) \right\} d^{(4)}x \right) / \\
&\left(\int_{-\infty}^{q_-} \int_{-\infty}^{q_-} \frac{1}{(2\pi) |\mathsf{Cov}_{2,2}(\mathbf{x_2})|^{1/2}} \right. \\
&\quad \left. \times \exp\left\{ -\frac{1}{2} (\mathbf{x_2} - \boldsymbol{\mu}_2)^T \cdot \mathsf{Cov}_{2,2}^{-1}(\mathbf{x_2}) \cdot (\mathbf{x_2} - \boldsymbol{\mu}_2) \right\} d^{(2)}x \right)
\end{aligned}$$

Since the underlying Gaussian process $\boldsymbol{h}$ is stationary, the Gaussian distributions of both $\mathbf{x_1}$ and $\mathbf{x_2}$ are symmetric. Also note that q_+ and q_- are symmetric with the mean as the center, so we can get the following equation:

$$\begin{aligned}&\mathsf{Pr}[BS_b(2i-1,2i)=``00"|BS_a(2i-1,2i)=``00"]\\&=\mathsf{Pr}[BS_b(2i-1,2i)=``11"|BS_a(2i-1,2i)=``11"].\end{aligned}\tag{8}$$

On the other hand, for each $i \in I_s$,

$$\begin{aligned}&\mathsf{Pr}[BS_b(2i-1,2i)]\\&=BS_a(2i-1,2i)|BS_a(2i)=BS_a(2i-1)\\&=\mathsf{Pr}[BS_a(2i)=1|BS_a(2i)=BS_a(2i-1)]\\&\quad\times\mathsf{Pr}[BS_b(2i-1,2i)=``11"|BS_a(2i-1,2i)=``11"]\\&\quad+\mathsf{Pr}[BS_a(2i)=0|BS_a(2i)=BS_a(2i-1)]\\&\quad\times\mathsf{Pr}[BS_b(2i-1,2i)=``00"|BS_a(2i-1,2i)=``00"].\end{aligned}\tag{9}$$

By combining (8) and (9), we get that

$$\begin{aligned}&\mathsf{Pr}[BS_b(2i-1,2i)]\\&=BS_a(2i-1,2i)|BS_a(2i)=BS_a(2i-1)\\&=\mathsf{Pr}[BS_b(2i-1,2i)=``11"|BS_a(2i-1,2i)=``11"]\\&=\mathsf{Pr}[BS_b(2i-1,2i)=``00"|BS_a(2i-1,2i)=``00"].\end{aligned}$$

Recall $\mathsf{Pr}[BS_b(2i-1,2i)= BS_a(2i-1,2i)|BS_a(2i)= BS_a(2i-1)]=q$ for any $i \in I_s$. From the way I_s is generated, we know that

$$\forall i \in I_s, BS_a(2i)=BS_a(2i-1), BS_b(2i)=BS_b(2i-1).$$

So, for any $i \in I_s$, $\mathsf{Pr}[BS_b(2i)=BS_a(2i)]=q$. We can rewrite it as $\mathsf{Pr}[BS_a(2\cdot I_s(j))=BS_b(2\cdot I_s(j))]=q$, where $j \in [1,n]$. The probability that B gets b_s is

$$\begin{aligned}&\mathsf{Pr}[b_s=b_s']\\&=\mathsf{Pr}[b_s=\mathsf{majority}(\{L_s(j)\oplus BS_b(2\cdot I_s(j)), j\in[1,n]\})]\\&=\mathsf{Pr}[b_s=\mathsf{majority}(\{b_s\oplus BS_a(2\cdot I_s(j))\oplus BS_b(2\cdot I_s(j)),\\&\qquad\qquad j\in[1,n]\})]\\&=\mathsf{Pr}\Big[|\{BS_a(2\cdot I_s(j))=BS_b(2\cdot I_s(j)), j\in[1,n]\}|>\frac{n}{2}\Big],\end{aligned}$$

Because the time interval between any two different pairs of probe signals is greater than the coherence time, the n events $\{BS_a(2\cdot I_s(j))=BS_b(2\cdot I_s(j))\}$, $j\in[1,n]$ are all independent. For each $j\in[1,n]$, define an indicator random variable

$$Ind_j=\begin{cases}1, & \text{if} \quad BS_a(2\cdot I_s(j))=BS_b(2\cdot I_s(j)),\\0, & \text{if} \quad BS_a(2\cdot I_s(j))\neq BS_b(2\cdot I_s(j)).\end{cases}$$

Then $Ind_1, Ind_2, \ldots, Ind_n$ are a sequence of independent Bernoulli random variables [18] with parameter q. Let $X(n,q)=|\{Ind_j=1, j\in[1,n]\}|$. Then $X(n,q)$ is a random variable following the binomial distribution $Binomial(n,q)$. Therefore,

$$\begin{aligned}&\mathsf{Pr}[b_s=b_s']\\&=\mathsf{Pr}\Big[|\{BS_a(2\cdot I_s(j))=BS_b(2\cdot I_s(j)), j\in[1,n]\}|>\frac{n}{2}\Big]\\&=\mathsf{Pr}\Big[|\{Ind_j=1, j\in[1,n]\}|>\frac{n}{2}\Big]=\mathsf{Pr}\Big[X(n,q)>\frac{n}{2}\Big]\\&=\sum_{i=\lfloor\frac{n}{2}\rfloor+1}^{n}\binom{n}{i}q^i(1-q)^{(n-i)}=1-\sum_{i=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n}{i}q^i(1-q)^{(n-i)}.\end{aligned}$$

Using the Hoeffding inequality [19], we can bound the above probability as follows:

$$\begin{aligned}&\mathsf{Pr}[b_s=b_s']=1-\sum_{i=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n}{i}q^i(1-q)^{(n-i)}\\&\geq 1-\mathsf{exp}\left(-2\cdot\frac{(nq-\frac{n}{2})^2}{n}\right)=1-\mathsf{exp}\left(-2n\cdot\left(q-\frac{1}{2}\right)^2\right).\end{aligned}$$

Because $q>\frac{1}{2}$, we can always make $\mathsf{Pr}[b_s=b_s']$ sufficiently close to 1 by increasing n. In particular, if we want the propability to be not less than $1-\epsilon$, then we only need to gurantee that $n\geq\frac{ln\left(\frac{1}{\epsilon}\right)}{2\left(q-\frac{1}{2}\right)^2}$. □

Remark. In Theorem 1 we have assumed $q>\frac{1}{2}$. We stress this is a realistic assumption because q can be controlled by adjusting α.

Theorem 2. *When our* OT_1^2 *protocol is finished,* B *gets no information about* b_{1-s}.

Proof. (Sketch) Let's consider the index sequence I_{1-s}. For each $i\in I_{1-s}$, we have that

$$\begin{aligned}&\mathsf{Pr}[BS_a(2i-1,2i)=``00"|BS_b(2i-1,2i)=``01"]\\&=\frac{\mathsf{Pr}[BS_a(2i-1,2i)=``00", BS_b(2i-1,2i)=``01"]}{\mathsf{Pr}[BS_b(2i-1,2i)=``01"]}\\&=\left(\int_{-\infty}^{q_-}\int_{-\infty}^{q_-}\int_{q_+}^{+\infty}\int_{-\infty}^{q_-}\frac{1}{(2\pi)^2|\mathsf{Cov}_{4,4}(\mathbf{x_1})|^{1/2}}\right.\\&\quad\left.\times\,\mathsf{exp}\{-\frac{1}{2}(\mathbf{x_1}-\boldsymbol{\mu}_1)^T\cdot\mathsf{Cov}_{4,4}^{-1}(\mathbf{x_1})\cdot(\mathbf{x_1}-\boldsymbol{\mu}_1)\}d^{(4)}x\right)\Big/\\&\qquad\mathsf{Pr}[BS_b(2i-1,2i)=``01"]\end{aligned}\tag{10}$$

Using the symmetry property of Gaussian distribution, we get that

$$\begin{aligned}&\left(\int_{-\infty}^{q_-}\int_{-\infty}^{q_-}\int_{q_+}^{+\infty}\int_{-\infty}^{q_-}\frac{1}{(2\pi)^2|\mathsf{Cov}_{4,4}(\mathbf{x_1})|^{1/2}}\right.\\&\quad\left.\times\,\mathsf{exp}\left\{-\frac{1}{2}(\mathbf{x_1}-\boldsymbol{\mu}_1)^T\cdot\mathsf{Cov}_{4,4}^{-1}(\mathbf{x_1})\cdot(\mathbf{x_1}-\boldsymbol{\mu}_1)\right\}d^{(4)}x\right)\Big/\\&\quad\mathsf{Pr}[BS_b(2i-1,2i)=``01"]\\&=\left(\int_{-\infty}^{q_-}\int_{q_+}^{+\infty}\int_{q_+}^{+\infty}\int_{q_+}^{+\infty}\frac{1}{(2\pi)^2|\mathsf{Cov}_{4,4}(\mathbf{x_1})|^{1/2}}\right.\\&\quad\left.\times\,\mathsf{exp}\left\{-\frac{1}{2}(\mathbf{x_1}-\boldsymbol{\mu}_1)^T\cdot\mathsf{Cov}_{4,4}^{-1}(\mathbf{x_1})\cdot(\mathbf{x_1}-\boldsymbol{\mu}_1)\right\}d^{(4)}x\right)\Big/\\&\qquad\mathsf{Pr}[BS_b(2i-1,2i)=``01"]\\&=\frac{\mathsf{Pr}[BS_a(2i-1,2i)=``11", BS_b(2i-1,2i)=``01"]}{\mathsf{Pr}[BS_b(2i-1,2i)=``01"]}.\end{aligned}$$

So,

$$\begin{aligned}&\mathsf{Pr}[BS_a(2i-1,2i)=``00"|BS_b(2i-1,2i)=``01"]\\&=\frac{\mathsf{Pr}[BS_a(2i-1,2i)=``00", BS_b(2i-1,2i)=``01"]}{\mathsf{Pr}[BS_b(2i-1,2i)=``01"]}\\&=\frac{\mathsf{Pr}[BS_a(2i-1,2i)=``11", BS_b(2i-1,2i)=``01"]}{\mathsf{Pr}[BS_b(2i-1,2i)=``01"]}\\&=\mathsf{Pr}[BS_a(2i-1,2i)=``11"|BS_b(2i-1,2i)=``01"].\end{aligned}\tag{11}$$

Since for each $i \in I_{1-s}$,

$$\mathsf{Pr}[BS_a(2i-1,2i) = \text{“}00\text{''}|BS_b(2i-1,2i) = \text{“}01\text{''}]+$$
$$\mathsf{Pr}[BS_a(2i-1,2i) = \text{“}11\text{''}|BS_b(2i-1,2i) = \text{“}01\text{''}] = 1$$

we have

$$\mathsf{Pr}[BS_a(2i-1,2i) = \text{“}00\text{''}|BS_b(2i-1,2i) = \text{“}01\text{''}] =$$
$$\mathsf{Pr}[BS_a(2i-1,2i) = \text{“}11\text{''}|BS_b(2i-1,2i) = \text{“}01\text{''}] = \frac{1}{2}$$

Similarly, we can get that

$$\mathsf{Pr}[BS_a(2i-1,2i) = \text{“}00\text{''}|BS_b(2i-1,2i) = \text{“}10\text{''}] =$$
$$\mathsf{Pr}[BS_a(2i-1,2i) = \text{“}11\text{''}|BS_b(2i-1,2i) = \text{“}10\text{''}] = \frac{1}{2}$$

From the above analysis, we can see that B gets no information about b_{1-s} from L_{1-s}. □

Theorem 3. *When our* OT_1^2 *protocol is finished,* A *gets no information about* s.

Proof. (Sketch) First we observe that A does not know which bits in BS_b are different from the corresponding bits in BS_a. So it is easy to see that, for any $i \in I_0 \cup I_1$, whether $i \in I_0$ or $i \in I_1$ is independent from the distribution of $BS_b(2i-1, 2i)$. So when the protocol is finished, A gets no information about s. □

The above theorems demonstrate the security guarantees of our OT_1^2 protocol. Nevertheless, all these theorems are proved in the semi-honest model and under the assumption that the eavesdropper is passive. In practice, if the participants of OT_1^2 can deviate from the protocol, or if there is an active adversary launching a man-in-the-middle attack, then our OT_1^2 protocol needs to be modified and improved.

4 APPLICATION I: PRIVATE COMMUNICATIONS

In this section, we develop a method based on our OT_1^2 protocol that, assuming A and B both know a secret key K, allows A to send a confidential message to B. Our target here is similar to symmetric key encryption and decryption in traditional cryptography. More precisely, we have (at least) the following requirements for our private communications method:

- If both A and B use the same key, then B should get the message sent by A.
- If A and B use two different keys, then B does not get the message sent by A.
- Any eavesdropper gets no information about the message sent by A.

However, we stress that our method is only similar to, *not identical* to symmetric key encryption and decryption in traditional cryptography. The reason is that our communication model is completely different from that of traditional cryptography and so the security model is also different. For example, with our method, there is no ciphertext in the traditional sense. Hence, issues like chosen plaintext attack (which allows an adversary to see the ciphertexts for his chosen plaintexts) and chosen ciphertext attack are not considered for our method.

The idea underlying our method of private communications is very simple: Imagine that the keys used by A and B are of only one single bit, and the message to be sent is also a single bit. In this (unrealistic) situation, A can easily send the message to B by executing an OT_1^2 with B. In this OT_1^2, B's secret bit is his key, and A's secret bit b_K is set to her message, where K is A's key. It is easy to verify that our requirements listed above are all satisfied.

Of course, in a realistic scenario, the keys and the message are much longer. So we need to extend the above idea to multiple bits. Nevertheless, there is a pitfall that we must avoid: If we use a straightforward extension of the above idea (i.e., doing an OT_1^2 for each bit of the key, assuming the key and the message are of equal length.), and if A and B use two different keys, then B may end up getting part of the message sent by A, each bit of which corresponding to a bit position at which the two keys agree. To avoid this pitfall, we let A hide her message using a random mask, and then the mask is sent from A to B using a number of OT_1^2 sessions. Therefore, if A and B have different keys, the mask B receives will be different from what A sends at a number of bit positions (where the two keys differ). But when B attemps to recover the message using the wrong mask, the error in the recovered message will not remain at these bit positions; instead, it will be spreaded over the entire message.

It is worth noting that not all security properties of our OT_1^2 protocol are needed in the construction of our private communications method. (For instance, the third security property of OT_1^2 is not needed in this application.) In other words, our method of private communications can actually be simplified and optimized, from a practical point of view. We present it in the current form just to demonstrate the power of our OT_1^2 protocol. In general, when we use a tool in an application, it may not be necessary to use all properties of that tool. For example, when we use encryption to protect all data in an application, the security property of encryption guarantees that all the involved plaintexts remain confidential. However, if the application itself wants to leak some plaintexts to the public, it should still be acceptable.

Below is our method of private communications.

Let p be a prime of length k (where k is a parameter) that is well known, i.e., everybody knows p. Suppose that A and B both know a key K that is of length k. Recall that the objective is to send a confidential message M from A to B. Without loss of generality, suppose $M \in Z_p$. The method consists of three steps.

Step 1. A selects a mask D from $[0, 2^k - 1]$ uniformly at random.[4] She then computes $C = (D \cdot M) \bmod p$, and sends C to B.

Step 2. Denote the jth bit of D by D_j, and the jth bit of K by K_j. For each $j \in [1, k]$, an OT_1^2 is executed between A and B, where A's two secret bits are $b_{K_j} = D_j$ and $b_{1-K_j} = 1 - D_j$, and B's secret bit is $s = K_j$.

Step 3. Once all the k OT_1^2 sessions are finished, B should have obtained all bits of D. Then B recovers M by computing $M = (C \cdot D^{-1}) \bmod p$.

The above private communications method is formally described in Algorithm 2.

---

[4] Ideally, the generation of random numbers in our protocol should not depend on computational assumptions. How to achieve this is out of the scope of this paper.

Algorithm 2 Private Communications Method

```
Data: $p, k, K; M \in Z_p$.
Result: $B$ receives $M$.

$A$:
    Select $D$ from $[0, 2^k - 1]$ uniformly at random.
    $C \leftarrow (D \cdot M) \bmod p$.
    Send $C$ to $B$.
    foreach $j \in [1, k]$ do
        perform $\mathrm{OT}_1^2$ $[b_{K_j} = D_j, b_{1-K_j} = 1 - D_j; s = K_j]$ with $B$.

$B$:
    $M = (C \cdot D^{-1}) \bmod p$
```

We emphasize that, in Algorithm 2, OT_1^2 should be our proposed OT_1^2 protocol based on wireless channel characteristics in Section 3.1. In particular, it should *not* be an OT_1^2 protocol based on conventional cryptography. The reason is that, if conventional cryptography is used, there can be much better ways to do private communications. In contrast, based on wireless channel characteristics, Algorithm 2 seems to be a natrual construction of private communications. Compared with private communications methods based on conventional cryptography, Algorithm 2 based on wireless channel characteristics does not rely on the hardness of any open computational problem, and so its security property is more reliable in the long run.

5 APPLICATION II: PRIVACY PRESERVING PASSWORD VERIFICATION

Besides private communications, our OT_1^2 protocol can also be applied to privacy preserving password verification. Today, password verification is still one of the major methods of user authentication. For example, in wireless LANs, many base stations authenticate users using their passwords at the beginning of sessions. However, it is clear that, when users send their passwords through wireless links, there is a risk that the passwords may be overheard by an adversary. Furthermore, an adversary may impersonate a base station or a password protected server to ask users for their passwords. Hence, it is important to consider the privacy protection of passwords when we use passwords for authentication.

In this section, we study privacy preserving password verification, which allows one wireless device to verify the password from another wireless device without the risk of revealing the password. More precisely, we have the following requirements when B verifies the password of A.

- If A's password matches the corresponding password in B's record, then B should accept.
- If A's password does not match the corresponding password in B's record, then B should reject.
- In any case, A learns nothing about the password in B's record except whether it matches A's password or not.
- In any case, B learns nothing about A's password except whether it matches the corresponding password in B's record or not.
- An eavesdropper should not learn anything about either A's password or the password in B's record.

In the above, the fourth requirement guarantees that, even if B is corrupted by an adversary, B will not be able to learn A's password as long as B has not already known it. (Of course, a corrupted device B might launch a probe attack, by repeatedly requesting A to do password authentication. Nevertheless, this is easy to prevent if A is required to stop trying after a few number of times.) So the fourth and fifth requirements together give a strong privacy protection for A's password. Similarly, the third and fifth requirements together give a strong privacy protection for the password in B's record.

To achieve the above objective, our main idea is to let A generate l pairs of random numbers and then execute OT_1^2 with B. After these OT_1^2, B receives one out of each pair of random numbers. So in total, B receives a sequence of l random numbers. Clearly, there are altogether 2^l such sequences, from which B choose to receive one. Among these 2^l sequences, only one sequence satisfies a special property: The product of all random numbers in this sequence is congruent to 1 (with respect to a prime modulus p). B will receive this special sequence through these OT_1^2 if and only if A's password matches the password in B's record. Therefore, in order to verify A's password, B only need to verify that the received sequence satisfies the special property described above.

Below are the details of our privacy preserving method for password verification.

Just like in Application I, let p be a well-known prime of length k, where k is a parameter. Without loss of generality, suppose that each password is of length l, where l is another parameter. Let Pass be A's password.

Step 1. A sends her user identity to B. Using this identity, B finds the corresponding password in B's record. Suppose that what B finds is Pass'.

Step 2. Denote by Pass_i (resp., Pass'_i) the ith bit of Pass (resp., Pass'). For each $i \in [1, l-1]$, A picks two random numbers $\beta_{0,i}, \beta_{1,i} \in Z_p$ independently and uniformly. Finally, A computes

$$\beta_{\mathsf{Pass}_\ell,\ell} = \left(\prod_{i=1}^{l-1} \beta_{\mathsf{Pass}_i,i} \right)^{-1} \pmod{p},$$

and picks $\beta_{1-\mathsf{Pass}_\ell,\ell} \in Z_p$ uniformly and independently.

Step 3. Denote by $\beta_{0,i,j}$ (resp., $\beta_{1,i,j}$) the jth bit of $\beta_{0,i}$ (resp., $\beta_{1,i}$). For each $i \in [1, l]$ and each $j \in [1, k]$, A and B execute an OT_1^2, where A's two secret bits are $\beta_{0,i,j}$ and $\beta_{1,i,j}$, and B's secret bit is Pass'_i; let $\beta'_{i,j}$ be what B receives in the OT_1^2.

Step 4. For each i, B puts together the k bits $\beta'_{i,1}, \beta'_{i,2}, \ldots, \beta'_{i,k}$ to get an integer β'_i. Then, B verifies that

$$\prod_{i=1}^{l} \beta'_i \equiv 1 (\bmod\ p).$$

A formal description of the above privacy preserving method for password verification is given in Algorithm 3.

Algorithm 3 Privacy Preserving Password Verification

```
Data: Pass, Pass', $p$, $k$, $l$.
Result: If Pass = Pass', then $B$ accepts $A$'s authentication
        request; otherwise $B$ rejects $A$'s authentication
        request.

A:
foreach $i \in [1, l-1]$ do
    pick two random numbers $\beta_{0,i}, \beta_{1,i} \in Z_p$ independently
    and uniformly.
$\beta_{\mathsf{Pass}_\ell,\ell} \leftarrow (\prod_{i=1}^{l-1} \beta_{\mathsf{Pass}_i,i})^{-1} (\mathrm{mod}\ p)$.
Pick $\beta_{1-\mathsf{Pass}_\ell,\ell} \in Z_p$ uniformly and independently.
foreach $i \in [1, l]$ do
    foreach $j \in [1, k]$ do
        perform $\mathrm{OT}_1^2\ [\beta_{0,i,j}, \beta_{1,i,j}; \mathsf{Pass}'_i]$ with $B$.
        (Denote the bit $B$ receives by $\beta'_{i,j}$.)

B:
foreach $i \in [1, l]$ do
    Combine $\beta'_{i,1}, \beta'_{i,2}, \ldots, \beta'_{i,k}$ to get $\beta'_i$
if $\prod_{i=1}^{l} \beta'_i \equiv 1 (\mathrm{mod}\ p)$ then
    accept $A$'s authentication request.
else
    reject $A$'s authentication request.
```

We stress that, just like in Algorithm 2, in Algorithm 3, OT_1^2 should be our proposed OT_1^2 protocol based on wireless channel characteristics. Specifically, it should *not* be an OT_1^2 protocol based on conventional cryptography. Compared with privacy preserving password verification methods based on conventional cryptography, Algorithm 3 based on wireless channel characteristics does not rely on any computational assumption, and thus its security property is more reliable in the long run.

6 IMPLEMENTATION AND EVALUATIONS

We completely implement our OT_1^2 protocol on two laptops, one with Intel Core2 CPU of 2.33 GHz and 2.0 GB memory, the other with Intel Pentium M CPU of 2.13 GHz and 1001.5 MB memory. Both laptops run the Ubuntu Linux 9.10 operating system and use Netgear WAG511 802.11abg wireless network cards. Both cards use ath5k [20] as drivers and are configured to operate in the 802.11a frequency band (specifically, the 5.745 GHz frequency band). The transmission power is set to be 30dBm for both cards.

In order that the two laptops can communicate directly without any intermediate relays, we configure one laptop in the access point (AP) mode, and configure the other laptop in the station mode. ICMP echo request packets are sent from the station to the AP at a constant rate. Once the AP receives the packet, it sends an ICMP echo reply packet back to the station.

We create one monitor interface on each of the two laptops, so that we can use tcpdump [21] to capture the packets. By customizing the tcpdump filters, we capture only ICMP echo request packets on the AP side and only ICMP echo reply packets on the station side. The received signal strength (RSS) in the radiotap header [22] is extracted from each captured packet. Because the transmission power levels for both sides are identical, the extracted RSS is a coarse measurement of the amplitude of wireless channel. (Ideally, rather than using RSS, our experiments should use raw physical layer complex channel impulse responses. However, in order to perform our experiments on *off-the-shelf* 802.11 network cards, we choose to use RSS, just like in [8], [9].) Each of the RSS measurements is quantized into one bit.

As pointed out in [8], [9], large-scale shadow fading can lead to long sequences of zeros and ones in the extracted bit strings. We use the adaptive quantization method [9] to mitigate this effect. Specifically, we divide all the RSS measurements into blocks and compute the quantization parameters [by equation (7)] for each block. We denote the size of each block by m, which is a configurable parameter.

We measure RSS profiles and the numbers of probings in three settings. In the first setting, the two laptops are stationary. In the second setting, the station moves at $1 \mathrm{m/s}$. In the third setting, the station moves at $2\ \mathrm{m/s} \sim 9\ \mathrm{m/s}$. In all three settings, we measure RSS profiles and minimum numbers of channel probings needed for an OT_1^2. The results are presented in Sections 6.1–6.3, respectively.

Besides the above experiments on RSS and the minimum number of channel probings, we have also experimentally studied the efficiency of our OT_1^2 protocol. The results are given in Section 6.4. We emphasize that all the experiments are carried out in real, physical environments (not in simulations).

Evaluation results of our OT_1^2 protocol's application can be found in [23].

Finally, we compare our OT_1^2 protocol with a simple OT_1^2 protocol using traditional cryptography [24]. The results are shown in Section 6.5.

6.1 OT_1^2 between Stationary Devices

In the first setting, we place the two laptops at fixed locations. Specifically, we place them on two tables in a library, and the distance between them is 15 meters. A number of people are walking in the library at speeds of $0.5 - 1 \mathrm{m/s}$, which causes variations in the wireless channel between the AP and the station. This environment is illustrated in Fig. 1.

In this setting, we first do an experiment to measure the RSS, which lasts for 300 seconds. During these 300 seconds, each laptop sends one probe signal every 100 milliseconds. From the captured packets, the RSS values are extracted and quantized into bit strings. Note that at both laptops we have implemented mechanisms to deal with packet losses and retransmissions, so that lost packets are removed from considerations and retransmitted packets are not repeatedly counted.

The extracted RSS sequences in the above experiment are shown in Fig. 2. Due to the channel reciprocity, the measured RSS profiles are mostly consistent. Inconsistencies exist because of receiver noises and interferences. We note that the absolute values of signal strengths have no influence on our

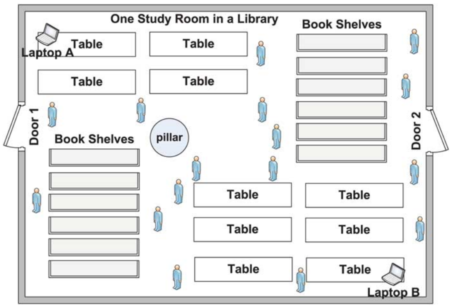


Fig. 1. The environment in the first setting.

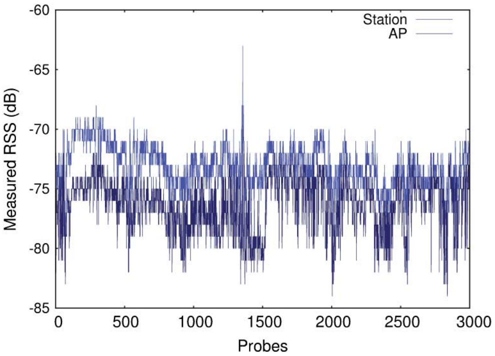


Fig. 2. Measured RSS profiles—the stationary setting.

OT_1^2 protocol because we use the adaptive quantization method.

Next, we do a number of experiments to measure the minimum number of channel probings required to achieve a certain error probability. (Here by error probability we mean the probability that the received bit in an OT_1^2 is not equal to b_s.) We repeat our experiment for different error probabilities between 0.01 and 0.0001, and for different combinations of quantization parameters m (size of block which RSS measurements are divided into) and α [a system parameter of equation (7)]. Fig. 3 shows our results. We can see that, to achieve an error probability of 10^{-3}, we only need about 150 channel probings when $m = 50$ and $\alpha = 0.25$.

6.2 OT_1^2 with Station Moving at $1\ m/s$

In the second setting, we place the AP on a table, and let the station move at a speed of $1\ \mathrm{m/s}$. The environment of this setting and the moving pattern of the station are shown in Fig. 4. The station moves along the arrowed path cyclically. Because the network cards are set to send and receive data in the 5.745 GHz frequency band, we can calculate the approximate channel coherence time according to the following equation, in which c is the speed of light and f is the central transmission frequency

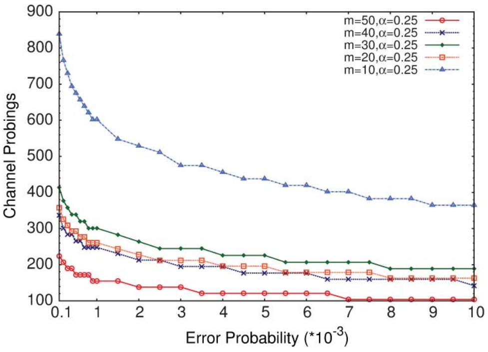


Fig. 3. The minimum channel probings to achieve required error probabilities—the stationary setting.

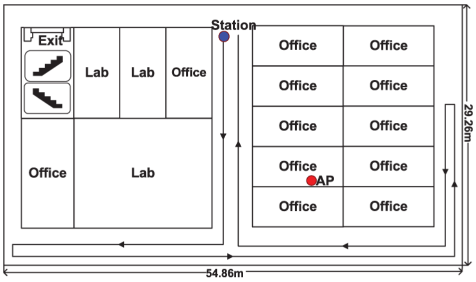


Fig. 4. The environment in the second setting.

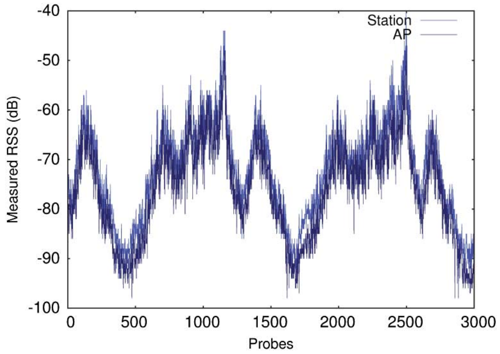


Fig. 5. Measured RSS profiles—the second setting.

$$T_C \approx \frac{\lambda}{v} = \frac{c}{f} = \frac{3 \cdot 10^8\, m/s}{5.745 \cdot 10^9\, Hz} \approx 52.219\, ms.$$

In this setting, we first do an experiment to measure the RSS, which lasts for about 160 seconds. During these 160 seconds, each laptop sends one probe signal every 53 milliseconds. From the captured packets, the RSS values are extracted and quantized into bit strings. The results are given in Fig. 5. We can see that due to the relative speed of $1\ \mathrm{m/s}$, there are more major fluctuations of signal strengths than in the first setting.

Next, just like in the first setting, we do a number of experiments to measure the minimum number of channel probings required to achieve a certain error probability. We repeat our experiment for different error probabilities between 0.01 and 0.0001, and for different combinations of quantization parameters m (size of block which RSS measurements are divided into) and α [a system parameter of equation (7)]. Fig. 6 gives our results. We can see that, to achieve an error probability of 10^{-3}, we only need about 100 channel probings when $m = 50$ and $\alpha = 0.2$.

6.3 OT_1^2 with Station Moving at $2 - 9m/s$

In the third setting, we do experiments in an empty ground. We place the AP on a car at the center of the ground, and drive another car around the AP. The experiment environment is shown in Fig. 7.

When the relative speed is increased, the channel coherence time decreases, which makes it hard to keep ICMP echo

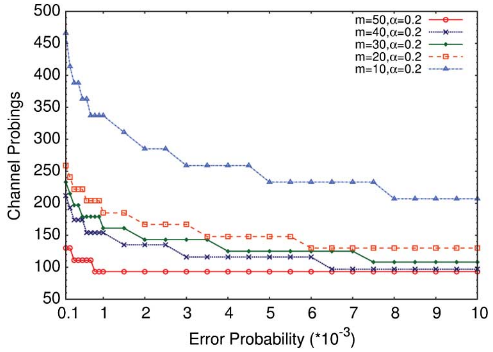


Fig. 6. The minimum channel probings to achieve required error probabilities—the second setting.

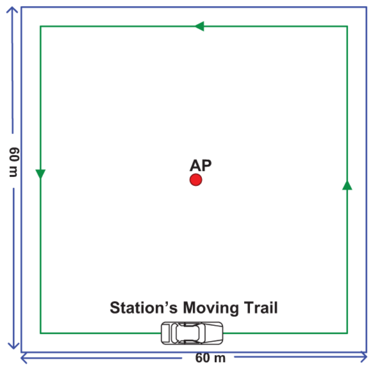


Fig. 7. The environment in the third setting.

request and reply in one coherence time. In order to solve this problem, we extract both the timestamp and the RSS from each message in the form of $(timestamp, RSS)$ pair. For each such pair of the AP (denote it by $(timestamp_1, RSS_1)$), we find $(timestamp_2, RSS_2)$ from the measurements of the station satisfying $|timestamp_1 - timestamp_2| \leq coherencetime$ and treat (RSS_1, RSS_2) as the effective channel measurements from one channel probing. We use the ntpd tool [25] to keep the system time synchronized at the two computers. Furthermore, when the relative speed increases, the small scale multi-path fading has more impact on channel variations. In order to keep the effects of small scale multi-path fading, we use the quantization parameter of a smaller $m (m = 2)$ and $\alpha = 0.1$. And after that, we make a permutation to the bitstrings to increase the probability of "11"s and "00"s at A.

We do a number of experiments in which the relative speed is from $2 \mathrm{m/s}$ to $9 \mathrm{m/s}$. At each relative speed, A and B measure RSS values and quantize them. The measured RSS values at the relative speed of $5 \mathrm{m/s}$ and $9 \mathrm{m/s}$ are shown in Figs. 8 and 9, respectively. Due to limit of space, we omit the measured RSS profiles at other relative speeds here. Interested readers can refer to [23] for these data. From Figs. 8 and 9, we can see that the channel fluctuations are more severe when the relative speed is high. However, the RSS measurements at A and B still have strong correlations.

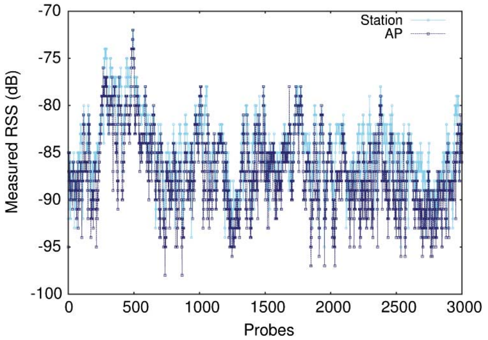


Fig. 8. Measured RSS profiles at the relative speed of $5 \mathrm{m/s}$.

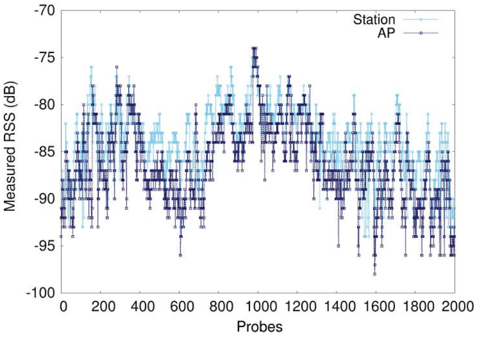


Fig. 9. Measured RSS profiles at the relative speed of $9 \mathrm{m/s}$.

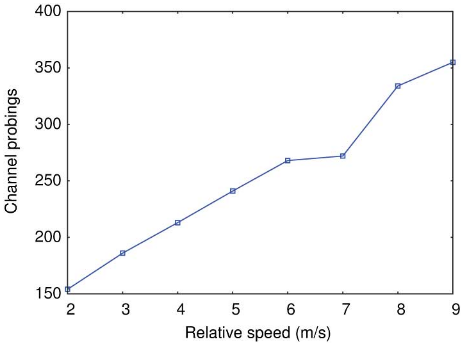


Fig. 10. The minimum channel probings at different relative speeds between AP and station.

We measure the minimum number of channel probings at different relative speeds in order to achieve 0.1% error probability, using the quantization parameter of a smaller $m (m = 2)$ and $\alpha = 0.1$. The results are shown in Fig. 10.

From Fig. 10 we can see that there is an increase of the minimum number of channel probings as the relative speed is increased. This is caused by two factors. Firstly, when the relative speed is increased, the number of lost packets also increases. Secondly, those $(timestamp, RSS)$ pairs that cannot find a match are discarded from the final channel measurements. When the relative speed is larger, the channel coherence time becomes smaller, which decreases the matching probability of message timestamps. Nevertheless, the performance of the protocol becomes better when the speed is increased (see Fig. 11), because the time interval between two pairs of channel probings can be smaller.

6.4 Efficiency of OT_1^2

To test the efficiency of our OT_1^2 protocol, we measure the running time at different relative speeds. At each relative speed, we run the OT_1^2 protocol for 100 times and record the average running time. For the $1\ m/s$ setting, each execution of the OT_1^2 protocol includes 100 channel probings. For the $2-9\ m/s$ settings, we use at least the minimum number of

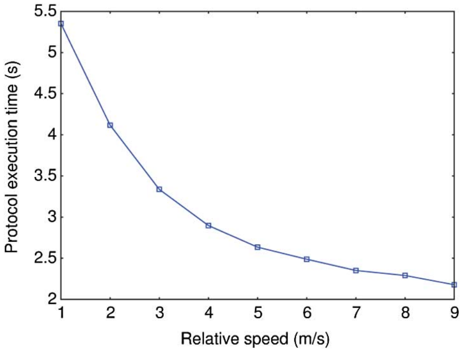


Fig. 11. OT_1^2 protocol execution time at different relative speeds.

channel probings shown in Fig. 10. The results are shown in Fig. 11.

From Fig. 11 we can see that our OT_1^2 protocol can be completed within several seconds if one participant moves relatively to the other at a normal walking speed. When the relative speed is increased, the protocol execution time decreases very quickly. For example, at a typical city driving speed of 20.1 mph ($9\ \mathrm{m/s}$), the OT_1^2 protocol can be finished within 2.5 seconds. The underlying reason for this phenomenon is that, when the relative speed increases, the channel coherence time decreases [as shown in Equation (5)]. On the other hand, as mentioned in Section 3.1, paragraph 1, in our protocol, a constraint on the time interval between any two different pairs of probe signals is that it must be more than the channel coherence time. Hence, when the relative speed increases, the time interval between two different pairs of probe signals can be decreased. This implies that the total execution time of the entire protocol can be decreased.

6.5 Comparison with OT_1^2 Using Traditional Cryptography

Finally, we compare our OT_1^2 protocol with OT_1^2 using traditional cryptography. For this purpose, we implement a simple 1-2 oblivious transfer protocol using RSA encryption [24]. This time, we use two identical laptops, both of which have Intel i5 CPU of 2.5 GHz and 2048 MB memory. Both laptops run the Ubuntu 12.04 operating system and use Realtek RTL8188CE 802.11bgn WiFi Adapter. Both adapter are configured to operate in the 802.11 g frequency band (specifically, the 2.462 GHz frequency band). Other settings in this set of experiments remain the same as the experiments we described in the previous sections.

One of the laptops is configured in the access point (AP) mode, and the other laptop in the station mode. We do this set of experiments in an empty ground, with the AP at the center of the ground. The laptop in the station mode moves around the AP, keeping a distance of 8 m from it.

We measure the running times of the two OT_1^2 protocols, at different relative speeds from $1\mathrm{m/s}$ to $15\mathrm{m/s}$. We run the simple 1-2 OT tansfer protocol using RSA for 100 times and record the average running time. Then, we also run our OT_1^2 protocol for 100 times (with each execution including the necessary number of channel probings), and record the average running time. The results are shown in Fig. 12.

It is pretty clear that there is a gap of efficiency between our OT_1^2 protocol and the one using RSA. This gap can be reduced if we increase the relative speed or allow the error probability of our OT_1^2 protocol to be greater. In the future, improved design of OT_1^2 based on physical channel characteristics is needed, in order to achieve efficiency similar to OT_1^2 using traditional cryptography.

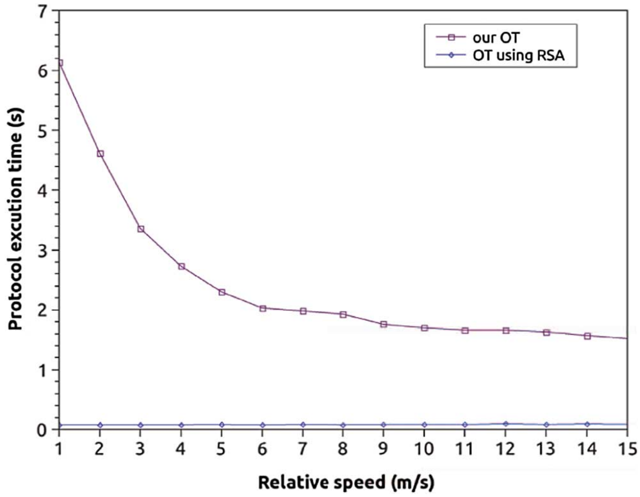


Fig. 12. The comparison of execusion time.

Similar results can be obtained if we compare the throughput of these two OT_1^2 protocols. For relative speeds from $1\mathrm{m/s}$ to $15\mathrm{m/s}$, the results on throughput comparison are shown in Fig. 13. Again, we can see there is a gap between these two protocols, which illustrates the need for improvement of design of OT_1^2 based on physical channel characteristics.

7 RELATED WORK

As we have mentioned, our work is motivated by the previous works on key agreement using wireless channel characteristics. In [26], [27], it is shown that secure key agreement can be achieved using the correlated information between two wireless devices as long as they share an authenticated channel beforehand. In [28], Hershey et al. propose a key agreement protocol that extracts secret bits from phase differences of continuous waves. After that, many other methods [4]–[10], [29]–[32] are proposed to enhance the security and/or

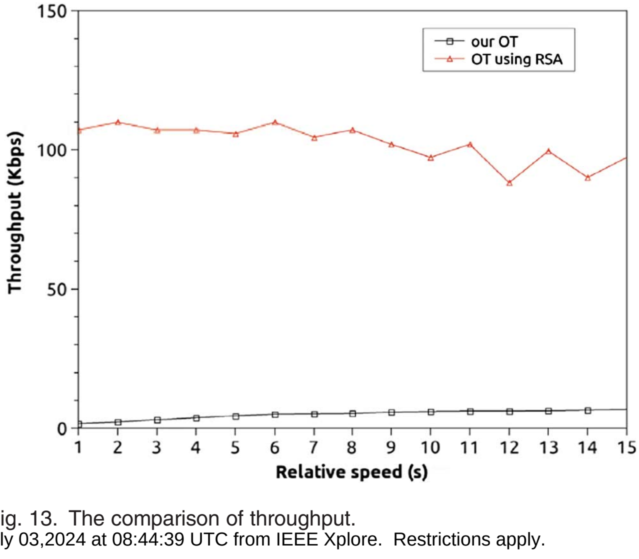


Fig. 13. The comparison of throughput.

improve the performance. In particular, Li et al. [4] propose a set of wireless security mechanisms, including wireless channel-based authentication, key extraction and key dissemination. In [6], Azimi et al. propose to achieve key agreement by quantizing the deep fading in mobile radio channels. The techniques of information reconciliation and privacy amplification [33] are used in their constructions.

Recently, Mathur et al. [8] propose a very practical method for secret key extraction from an unauthenticated wireless channel. They design a level crossing algorithm for achieving key agreement between the protocol participants. Their method is resistant to spoofing attack. To improve the secret bit rate efficiently, Jana et al. [9] design an adaptive and multi-bit quantization method for secret bit extraction. They do extensive experiments under a diversity of environments and make comparisons among them. In [10], a high rate uncorrelated bit extraction scheme is proposed, which further improves the efficiency by using fractional interpolation, de-correlation transformation and multi-bit adaptive quantization. Another recent work by Ye et al. [13] presents improvements in both efficiency and generality of channel state distributions.

While the aforementioned works are on key agreement, our work is on oblivious transfer (OT), or more precisely, OT_1^2. OT is a fundamental cryptographic tool that has been used in constructions of many complex cryptographic protocols. It is first proposed by Rabin [34]. Even, Goldreich and Lempel [35] propose OT_1^2, an important variant of OT. Crépeau [36] shows that OT_1^2 is equivalent to the original version of OT proposed by Rabin. The importance of OT is reflected by its completeness [11], [37]–[39]. In his seminal work, Kilian [11] shows that any general two-party cryptographic protocol can be built using OT. In [37]–[39], this result is extended to multiparty protocols.

In a theoretical work [12], Crépeau and Kilian propose an OT_1^2 protocol based on noisy channels. Crépeau also proposes another OT_1^2 protocol in a follow-up work [40] to increase the efficiency. The noisy channels they consider are simple discrete memoryless channels. In contrast, our OT_1^2 protocol is based on wireless channels, which are much more realistic and complicated, having severe fluctuations with varying time and locations. Furthermore, in addition to theoretical analysis, we have fully implemented our OT_1^2 protocols with off-the-shelf 802.11 network cards and carried out extensive experiments.

We have demonstrated two applications of our OT_1^2 protocol, private communications and privacy preserving password verification. In fact, there have been a number of works on private communications using the secrecy capacities of the wireless channel, e.g., [17], [41]–[45], among others. In particular, Vasudevan et al. [42] try to defend against the eavesdroppers by sending artificial noises to them, and focus on the scaling laws of secret communications without computational assumptions. In contrast, our private communications method is more practical in the sense that it does not need to control the received signals at the eavesdroppers. On the other hand, we stress that our private communications method is to illustrate the application of our OT_1^2 protocol. We choose this application because it is simple and easy to understand, *not* because our private communications method is more efficient than the existing works on private communications.

To the best of our knowledge, there is no existing work using physical channel characteristics to do privacy preserving password verification. In contrast, there are numerous works (e.g., [46]–[48]) using traditional cryptography to do privacy. For example, Chai et al. [46] propose an efficient password-based authentication scheme, together with a key exchange protocol. Their scheme protects users' privacy in that the identities of users are not leaked in the process of authentication. Li et al. [47] design a password verification protocol that protects user passwords from the server using a security authority and encryption techniques. With their protocol, the server does not even have a password verification table in storage. Another password verification protocol proposed by Das et al. [48] eliminates the need for a password verification table as well.

8 CONCLUSION

In this paper, we propose an OT_1^2 protocol in the setting of a wireless network and give two applications of this protocol to illustrate its potential broad applications. The main advantage of our OT_1^2 protocol is that it does not rely on any computational assumption. For security critical applications in wireless networks, such an advantage is of great importance, because as we have seen in the history, cryptographic tools based on computational assumptions may be broken after being used for years.

Although at this moment, our OT_1^2 protocol is still not as fast as the traditional OT_1^2 protocols based on computational assumptions, it has shown the feasibility of basing wireless security on physical channel characteristics, rather than on computational assumptions. Hence, our work can be considered a crucial step towards building wireless security systems that do not rely on computational assumptions.

In terms of security, our OT_1^2 protocol and its applications are secure in the semi-honest model, and under the assumption that there is only a passive eavesdropper besides the protocol participants. Another precondition of the security of our OT_1^2 protocol is that the involved wireless channels are fading channels and that the adversary has a distance of at least half a wavelength from any protocol participant. Note that if this precodition is not satisfied (e.g., if the communications are in an open space), then our protocol may not be usable. We leave to future work the considerations of fully malicious model, active man-in-the-middle attack, and/or communications in open space.

ACKNOWLEDGMENTS

Sheng Zhong is the corresponding author. This work was supported in part by RPGE, NSFC- 61321491, NSFC-61300235, and National Science and Technology Major Project of China (2010ZX03004-003-03). This work was partly done while Zhuo Hao was a visiting student at SUNY Buffalo and supported by NSF CNS-0845149. Part of the results were presented at IEEE INFOCOM 2011.

REFERENCES

[1] FIPS Publication 197, *Advanced Encryption Standard*. NIST, 2001.

[2] X. Wang and H. Yu, "How to break MD5 and other hash functions," *Proc. 24th Ann. Int. Conf. Theory Appl. Cryptogr. Techn. (EUROCRYPT'05)*, 2005, pp. 19–35.

[3] X. Wang, H. Yu, and Y. L. Yin, "Efficient collision search attacks on SHA-0," *Proc. 25th Ann. Int. Conf. Adv. Cryptol. (CRYPTO'05)*, 2005, pp. 1–16.

[4] Z. Li, W. Xu, R. Miller, and W. Trappe, "Securing wireless systems via lower layer enforcements," *Proc. 5th ACM Workshop Wireless Security (WiSe '06)*, 2006, pp. 33–42.

[5] C. Ye, A. Reznik, and Y. Shah, "Extracting secrecy from jointly Gaussian random variables," *Proc. IEEE Intl. Symp. Inf.*, Jul. 2006, pp. 2593–2597.

[6] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," *Proc. 14th ACM Conf. Comput. Comm. Security (CCS'07)*, 2007, pp. 401–410.

[7] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2008)*, 2008, pp. 3013–3016.

[8] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," *Proc. 14th Int. Conf. Mobile Comput. Netw. (MobiCom'08)*, 2008, pp. 128–139.

[9] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," *ACM Proc. 15th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom'09)*, 2009, pp. 321–332.

[10] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Trans. Mobile Comput*, vol. 9, no. 1, pp. 17–30, Jan. 2010.

[11] J. Kilian, "Founding crytpography on oblivious transfer," *Proc. 12th Ann. ACM Symp. Theory Comput. (STOC'88)*, 1988, pp. 20–31.

[12] C. Crépeau and J. Kilian, "Achieving oblivious transfer using weakened security assumptions," in *Proc. 29th Ann. Symp. Found. Comput. Sci. (FOCS'88)*, Oct. 1988, pp. 42–52.

[13] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 240-254, Jun. 2010.

[14] J. Tugnait, L. Tong, and Z. Ding, "Single-user channel estimation and equalization," *IEEE Signal Process. Mag.*, vol. 17, no. 3, pp. 16–28, May 2000.

[15] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[16] O. Goldreich, *Foundations of Cryptography*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[17] Y. Liang, H. Poor, and S. Shamai, "Secure communication over fading channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2470–2492, June 2008.

[18] S. Ross, *A First Course in Probability*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[19] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 13-30, 1963.

[20] *Ath5k Linux Wireless Network Card Driver* [Online]. Available: http://linuxwireless.org/en/users/Drivers/ath5k

[21] *Tcpdump* [Online]. Available: http://www.tcpdump.org/

[22] *Radiotap Header* [Online]. Available: http://www.radiotap.org/

[23] Z. Hao, S. Zhong, L. E. Li, H. Yao, and N. Yu, "Towards wireless security without computational assumptions," SUNY Buffalo Comput. Sci. Eng. Depart., SUNY Buffalo, Buffalo, NY, Tech. Rep. 2011-03, http://www.cse.buffalo.edu/tech-reports/2011-03.pdf, Apr. 2011.

[24] Y.-F. Chang, "Non-interactive t-out-of-n oblivious transfer based on the RSA cryptosystem," *Proc. 13rd Int. Conf. Intell. Inf. Hid. Multimedia Signal Process., (IIHMSP '07)*, 2007, pp. 45–50.

[25] *Network Time Protocol (NTP) Daemon* [Online]. Available: http://linux.die.net/man/1/ntpd.

[26] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[27] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography. I. Secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121–1132, Jul. 1993.

[28] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Trans. Commun.*, vol. 43, no. 1, pp. 3–6, Jan. 1995.

[29] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *Digit. Signal Prog.*, vol. 6, no. 4, pp. 207–212, 1996.

[30] M. Tope and J. McEachen, "Unconditionally secure communications over fading channels," *Proc. IEEE Military Comm. Conf. (MILCOM 2001)*, 2001, vol. 1, pp. 54–58.

[31] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Trans. Antennas Propag.*, vol. 53, no. 11, pp. 3776-3784, Nov. 2005.

[32] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin, "Wireless information-theoretic security," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2515-2534, Jun. 2008.

[33] C. Cachin and U. M. Maurer, "Linking information reconciliation and privacy amplification," *J. Cryptol.*, vol. 10, no. 2, pp. 97–110, 1997.

[34] M. O. Rabin, "How to exchange secrets with oblivious transfer," Aiken Computation Lab, Harvard Univ., Cambridge, MA, Tech. Memo TR-81, 1981.

[35] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Commun. ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[36] C. Crépeau, "Equivalence between two flavours of oblivious transfers," *Proc. Conf. Theory Appl. Cryptogr. Techn. Adv. Cryptol. (CRYPTO '87)*, 1988, pp. 350–354.

[37] S. Goldwasser and L. A. Levin, "Fair computation of general functions in presence of immoral majority," *Proc. 10th Ann. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO '90)*, 1991, pp. 77–93.

[38] C. Crépeau, J. v. d. Graaf, and A. Tapp, "Committed oblivious transfer and private multi-party computation," *Proc. 15th Ann. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO '95)*, 1995, pp. 110–123.

[39] Y. Ishai, M. Prabhakaran, and A. Sahai, "Founding cryptography on oblivious transfer—efficiently," *Proc. 28th Ann. Conf. Cryptol.: Adv. Cryptol. (CRYPTO'08)*, 2008, pp. 572–591.

[40] C. Crépeau, "Efficient cryptographic protocols based on noisy channels," *Proc. 16th Ann. Int. Conf. Theory Appl. Cryptogr. Techn. (EUROCRYPT'97)*, 1997, pp. 306–317.

[41] P. Gopala, L. Lai, and H. El Gamal, "On the secrecy capacity of fading channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4687–4698, Oct. 2008.

[42] S. Vasudevan, D. Goeckel, and D. Towsley, "Security-capacity trade-off in large wireless networks using keyless secrecy," *Proc. 11th ACM Int. Symp. Mobile ad hoc Netw. Comput. (MobiHoc'10)*, 2010, pp. 21–30.

[43] E. Perron, S. Diggavi, and E. Telatar, "On cooperative wireless network secrecy," *Proc. IEEE Conf. Comput. Commun. (INFOCOM '09)*, Apr. 2009, pp. 1935–1943,.

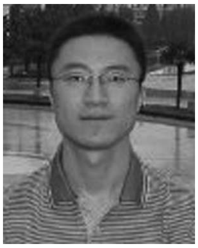
[44] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, "Improving wireless physical layer security via cooperating relays," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1875–1888, Mar. 2010.

[45] L. Dong, Z. Han, A. Petropulu, and H. Poor, "Secure wireless communications via cooperation," *Proc. 46th Ann. Allerton Conf. Comm., Control, Comput.*, Sept. 23–26, 2008, pp. 1132–1138.

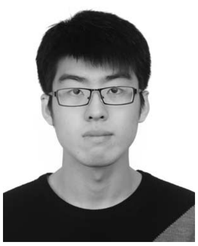
[46] Z. Chai, Z. Cao, and R. Lu, "Efficient password-based authentication and key exchange scheme preserving user privacy," *Lect. Note Comput. Sci.*, vol. 4138, pp. 467–477, 2006.

[47] L.-H. Li, I.-C. Lin, and M.-S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1498–1504, Nov. 2001.

[48] A. G. V. Das and M. L. Saxena, "A dynamic ID-based remote user authentication scheme," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629-631, May 2004.



Zhuo Hao recieved the BE and PhD degrees in information security from the University of Science and Technology of China, Hefei, China, in 2006 and 2011, respectively. Currently, he works at Micro Strategy, Hangzhou, China.



Yunlong Mao is an undergraduate student of computer science in Nanjing University, Nanjing, China, will recieve the BS degree and become a PhD student of Computer Scince in Nanjing University next year.

Sheng Zhong recieved the BS and ME degrees in computer science from Nanjing University, Nanjing, China, in 1996 and 1999, respectively and PhD degree in computer science from Yale University, New Haven, CT, in 2004. His research interests include security, privacy, and economic incentives.

Haifan Yao received the BE and ME degrees in computer science from Beijing University of Post and Telecommunications, Beijing, China, in 2005 and 2008, respectively. He is pursuing the PhD degree at the University at Buffalo, New York.

Li Erran Li received the PhD degree in computer science from Cornell University, Ithaca, NY, in 2001 where Joseph Y. Halpern was his advisor. Since graduation, he has been with Bell Labs. He is also an adjunct professor with the Department of Computer Science, Columbia University, New York. His research interests include networking and systems with a focus on wireless networking and mobile computing. He is an Associate Editor of *IEEE Transactions on Networking*, *IEEE Transactions on Mobile Computing (TMC)*, *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, *WINET*, and a Guest Editor for JSAC Special Issue on Non-Cooperative Behavior in Networking. He co-organized a number of workshops: DIMACS Workshop on Software Defined Networking, December 2012, first ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet) 2012, DIMACS Workshop on Systems and Networking Advances in Cloud Computing, December 2011, first ACM Workshop on Mobile Cloud Computing and Services (MCS) co-located with MobiSys, June 2010. He has published over 70 papers and holds 11 US Patents.

Nenghai Yu is a professor at the University of Science and Technology of China, and Director of Information Processing Center. His research interest includes information security.