Secure Key Generation for Distributed Inference in IoT

Invited Presentation

Henri Hentilä*, Visa Koivunen*, H. Vincent Poor†, and Rick S. Blum‡
* Department of Signal Processing and Acoustics, School of Electrical Engineering, Aalto University, Finland
† Department of Electrical Engineering, Princeton University, NJ, USA
‡ Department of Electrical and Computer Engineering, Lehigh University, PA, USA

***Abstract*—A secret key generation scheme is proposed for generating keys to be used for one-time pad encryption. This type of encryption is suitable for e.g. short packet communication in distributed inference in IoT. The scheme exploits the phase of the channel fading coefficient in a Rayleigh fading channel to extract highly correlated key bits at two legitimate parties. Compared to other existing methods, the proposed scheme trades off higher bit error probabilities in the keys for lower error correction communication requirements. The bit error of generated keys is characterized via an approximate upper bound, which is shown to be fairly tight for reasonable signal-to-noise ratios.**

I. Introduction

Data security has traditionally been handled via cryptographic methods that scramble the bits of the data using another set of bits referred to as the key. Because the key itself must be kept secret from potential attackers, the distribution or generation of such keys is a problem of its own.

One approach to generating secret keys is based on the concept of information theoretic secrecy introduced by Shannon [1]. In particular, [2] and [3] established bounds on the *common information* of two correlated random variables. This common information describes how many shared bits two parties with access to different but correlated random variables can theoretically establish between themselves without leaking information of these bits to outside parties.

The first attempts at constructing practical schemes based on the common information of correlated random variables include [4] and [5]. In these works, the wireless medium used in radio communication is found to be a suitable extractor of common information. Some more recent approaches include [6], [7], [8], [9].

In this paper, we propose a key generation scheme that is similar to the ones in [7], [8], [9]. However, unlike the keys generated in [7], [8], [9], we do not intend for the generated key to be used in any arbitrary encryption method. Instead, we specifically design our method to generate keys that are used for *one-time pad* encryption. One-time pad encryption is an encryption method where a key bit is added (via the exclusive OR operation) to each message bit. This is known to provide the perfect secrecy discussed by Shannon in [1], but is often seen as idealistic and not usable in practice as it would require keys of the same size as the message itself. However, there are certain situations where we may be able to generate key bits at the same rate as new plaintext bits are generated.

One such situation is the wireless sensor network studied in [10], which inspired this paper. This type of sensor network, envisioned as an important part of the future Internet of Things (IoT), can be used to e.g. monitor systems or detect anomalies. In [10], sensors send short messages of only a few bits each infrequently to a fusion center, making the generation of key bits at the message rate feasible. In addition to providing perfect secrecy for the messages, the use of one-time pad encryption in this case would allow us to conceal the encryption itself (due to both input and output bits being uniformly distributed under nominal conditions), making it particularly suitable for this particular problem.

With the above in mind, the contributions of this paper are the following:

- A secret key generation scheme is proposed for generating keys to be used specifically for one-time pad encryption. By explicitly focusing on keys used in one-time pad encryption, we can trade off higher bit error probabilities in the keys for lower error correction communication requirements compared to existing generation schemes.
- The bit error of generated keys is characterized via an approximate upper bound, which is shown to be fairly tight for reasonable signal-to-noise ratios.

The rest of the paper is organized as follows. In Section II, the problem of secret key generation is formally described. In Section III, the proposed key generation scheme is presented. In Section IV, an upper bound on the bit error probability of the proposed scheme is derived, and in Section V this bound is compared to the true error probability. Finally, the paper is concluded in Section VI.

II. Problem Description

Alice and Bob want to establish a shared key $K \in \mathcal{K} = \{1, \dots, 2^k\}$, represented as a k-bit string. This key has to be kept secret from the (passive) eavesdropper Eve. In order to establish the secret key, Alice and Bob have access to a source of correlated random variables (X, Y, Z) as well as a noiseless public channel that they can use to communicate with each other. From the source (X, Y, Z), Alice receives X, Bob Y and Eve Z. Any communication over the public channel is assumed noise-free and can be heard by any of the three.

This work is supported by Academy of Finland WiFiUS grant "Secure Inference in the Internet of Things" and is based upon work partially supported by the National Science Foundation under grants CNS-1702555, and CNS-1702808.

978-1-7281-1151-3/19/$31.00 ©2019 IEEE

The secret key generation proceeds as follows. From her source variables X, Alice will generate messages $M_A = m_A(X)$, which she sends in the clear over the public channel to Bob. Correspondingly, Bob generates from his source variables Y messages $M_B = m_B(Y)$ to be sent over the public channel to Alice. Upon receiving each other's messages, Alice and Bob generate their keys $K_A = g_A(X, M_B)$ and $K_B = g_B(Y, M_A)$, respectively. Due to the broadcast nature of the communication channel, Eve will have access to messages M_A and M_B in addition to her source variable Z.

Our goal is to design the message functions m_A and m_B, and the key generator functions g_A and g_B such that

$$\Pr(K_A[i] = K_B[i]) \geq 1 - \epsilon,\ \forall i = 1, \ldots, k, \tag{1}$$

$$I(K_A; Z, M_A, M_B) \leq \epsilon, \tag{2}$$

$$H(K_A) \geq \log |\mathcal{K}| - \epsilon, \tag{3}$$

where $H(X)$ denotes the entropy of X, $I(X; Y)$ the mutual information of X and Y, and $K[i]$ the ith bit in the key K. In the above, condition (1) ensures that Alice and Bob generate the same key bits with a high enough probability; condition (2) ensures that the key generated by Alice is kept secret from Eve; and condition (3) ensures that the key bits generated by Alice are uniformly distributed. Uniformity of the key is necessary for one-time pad encryption to produce cipher texts that satisfy Shannon's secrecy condition in [1].

A. *Using the Fading Wireless Channel as Source*

So far, we have described the problem from a general point of view, without specifying the source (X, Y, Z) providing the input to our key generation protocol. Finding an appropriate real-world representation of this source is paramount to generating keys at a high enough rate. In particular, we would like to find a source where X and Y are highly correlated, while Z is as uncorrelated as possible with either.

It turns out that the wireless medium provides a good framework for establishing such a source. Namely, the wireless channel is reciprocal, meaning that the channel gain from a terminal A to a terminal B is the same as the channel gain from B to A (excluding hardware noise). Therefore, A and B can estimate the channel gain between them independently from each other, and the resulting estimates will be highly correlated. Furthermore, if the environment between the terminals is rich enough in scatterers and/or its scatterers are highly mobile (which is often the case in urban environments), the instantaneous channel fading coefficient will be highly variable depending on e.g. the location of the terminals. In such circumstances, an eavesdropper that is far enough apart from either terminal will have a very different channel fading coefficient between itself and either of the two terminals. Thus, the channel fading coefficient h makes for a suitable source.

In this paper, we will assume that the channel is a Rayleigh fading channel. This is the most common type of channel considered in the literature, as it is a good approximation of an urban environment where we have a high amount of scatterers and/or mobility in the scatterers and with no line of sight (LoS) between transmitter and receiver. In a Rayleigh fading channel the channel fading coefficient $h \in \mathbb{C}$ is complex normally distributed with mean 0 and variance P, denoted $h \sim \mathcal{CN}(0, P)$. That is, both the real and imaginary components of h are normally distributed with mean 0 and variance P and independent of each other. Then, for a transmitted signal $s \in \mathbb{C}$, the received signal $r \in \mathbb{C}$ is $r = hs + v$, where $v \in \mathbb{C}$ is some independent additive noise. Both Alice and Bob will try to estimate the channel fading coefficient h. Alice's estimate h_A, and Bob's corresponding estimate h_B are assumed to be

$$h_A = h + w_A, \quad h_B = h + w_B \tag{4}$$

That is, the only difference in their estimates comes from the additive noise terms w_A and w_B. For the rest of this paper, we will assume that $w_A \sim \mathcal{CN}(0, N)$ and $w_B \sim \mathcal{CN}(0, N)$ for some constant noise variance N.

Meanwhile, Eve's channel fading coefficient h_{AE} between herself and Alice, and h_{BE} between herself and Bob are both assumed independent from h (in a fast fading channel this is the case when she is at least half a wavelength away from either of the two). Thus, regardless of which channel Eve chooses to estimate, this estimate h_E will be completely independent from h_A and h_B. We then choose (h_A, h_B, h_E) as our source (X, Y, Z).

Notice that because $Z = h_E$ is independent from h_A and h_B, condition (2) can be written as

$$I(K_A; M_A, M_B) \leq \epsilon. \tag{5}$$

That is, we only need to ensure that the messages sent over the public channel are independent from the generated key K_A.

III. Secret Key Generation

The secret key generation protocol proposed in this paper consists of the following two steps:

1) **Quantization:** The continuous channel fading coefficient estimates h_A and h_B are quantized into one of a finite number of possible discrete symbols $S \in \mathcal{S}$ and $S' \in \mathcal{S}$, respectively. This is done via the quantization function $q : \mathbb{C} \mapsto \mathcal{S}$.
2) **Information reconciliation:** Alice and Bob exchange messages $M_A = m(h_A)$ and $M_B = m(h_B)$ over the public channel that, together with the quantization function q from the previous step and their channel estimates h_A and h_B allows them to determine keys $K_A = g(h_A, M_B)$ and $K_B = g(h_B, M_A)$ with a low enough bit error probability.

The above differs slightly from how we defined the key generation in the previous section. First of all, the quantization was not discussed previously, and is a necessary first step to convert the continuous channel estimates to discrete symbols that can be used as keys. In addition, we do not use separate message and key generator functions at Alice and Bob.

In the following subsections we will describe these two steps in detail.

A. Quantization

For the quantization, our goal is to design a quantization function $q : \mathbb{C} \mapsto \mathcal{S}$ that maps the (continuous) complex channel fading coefficient estimate h_A (or h_B) to a discrete symbol $S \in \mathcal{S} = \{1, \ldots, 2^k\}$. Note that k is a system parameter, and should be chosen carefully. Because the final output $K \in \mathcal{K}$ of the key generator function g must satisfy the uniformity condition (3), we will find it helpful to make sure that the output of q is also uniform.

The channel fading coefficient h is a complex number, and therefore has an amplitude and a phase. In particular, we notice that, since $h \sim \mathcal{CN}(0, P)$ in the Rayleigh fading channel considered, the phase of h is uniformly distributed. Thus, we can construct the quantization function q to have a uniform output with 2^k possible values by simply partitioning the phase space of h into 2^k equally sized intervals. Each of these intervals will then correspond to one symbol $S \in \mathcal{S}$.

More precisely, the quantization can be described via the quantization parameter k using the following definition:

Definition 1. *Given k, define*

$$\theta = 2\pi/2^k \tag{6}$$

to be the size (angle) of a quantization interval. Then, for each $n = 1, 2, \ldots, 2^k$,

$$g_n = (n-1)\theta \tag{7}$$

is the angle of quantization border n.

Now let ϕ_A be the phase of h_A. Then $q(h_A) = n$ for the $n \in \mathcal{S}$ that satisfies:

$$g_n \leq \phi_A \leq g_{n+1}, \tag{8}$$

where $g_{2^k+1} = 2\pi$. Because ϕ_A is uniformly distributed, and $g_{n+1} - g_n = \theta$ for all n, so is the output S as desired.

B. Information Reconciliation

For the information reconciliation, our goal is to design a message function $m : \mathbb{C} \mapsto \mathcal{M}$ that maps channel estimates h_A (or h_B) to the set of messages $\mathcal{M}$, and a key generator function $g : \mathbb{C} \times \mathcal{M} \mapsto \mathcal{K}$ that maps the symbols acquired in the previous phase together with the received messages to the final key space $\mathcal{K}$. Recall that m and g must be designed such that conditions (1), (5), and (3) can be satisfied for a given ϵ. We will achieve this by introducing *guard zones* around the quantization borders g_n.

In particular, notice that if ϕ_A is close to one of the borders g_n, $n = 1, 2, \ldots, 2^k$, then it is likely that ϕ_B is on the other side of the border. In such a case the two quantized symbols $S = q(h_A)$ and $S' = q(h_B)$ will be different, which is undesirable. To guard against this, we add guard zones around the borders, which allow us to disregard any estimates h_A and h_B that fall within them.

The guard zones are rectangular areas around each border g_n. For this, it will be convenient to treat the complex space $\mathbb{C}$ as the 2-dimensional real space $\mathbb{R}^2$. Therefore, let $\mathbf{vec} : \mathbb{C} \mapsto \mathbb{R}^2$ denote an operator that maps a complex number to a vector in 2D-space. That is $\mathbf{vec}(x+iy) = (x, y)$. Then we can define the guard zones via the guard parameter δ as follows:

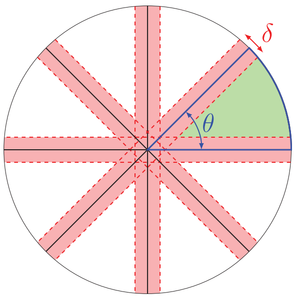


Fig. 1: Example of quantization borders (black lines) and their corresponding guard zones (red areas) for $k = 3$. The area enclosed in blue corresponds to one quantization sector, within which any estimates h_A (or h_B) are quantized to the same symbol S.

Definition 2. *The guard zone $G_n \subset \mathbb{C}$ corresponding to quantization border g_n, $n = 1, \ldots, 2^k$ and with parameter δ, is*

$$G_n = \{h \mid -\delta/2 \leq \mathbf{vec}(h)^T \mathbf{vec}(e^{i(g_n+\pi/2)}) \leq \delta/2\} \tag{9}$$

The union of all guard zones is denoted G:

$$G = \bigcup_{n=1}^{2^k} G_n. \tag{10}$$

Of key importance will be the guard parameter δ, which defines the width of the guard zones. As we will see later on, adjusting the size of δ will allow us to achieve the desired bit error probability ϵ. Also notice that, because the quantization borders are symmetric around the origin, the guard zones G_n for $n = 2^{k-1}+1, \ldots, 2^k$ are simply duplicates of the first 2^{k-1} guard zones, and can therefore be ignored when appropriate. Fig. 1 illustrates the quantization borders and their corresponding guard zones for $k = 3$.

Our message space $\mathcal{M}$ will now be $\mathcal{M} = \{0, 1\}$ and the message function is defined as

$$m(h_A) = \begin{cases} 1 & \text{if } h_A \in G \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

That is, if the estimate is in a guard zone, Alice will send the message 1, and otherwise she will send the message 0. Bob performs the corresponding operations based on his estimate h_B.

The key space $\mathcal{K}$ is $\mathcal{K} = \mathcal{S} \cup \{\perp\}$, where $\perp$ denotes an empty key (i.e. a 0-bit key). After receiving Bob's message M_B, Alice generates her key K_A via the following function

$$g(h_A, M_B) = \begin{cases} q(h_A) & \text{if } M_B = 0 \text{ and } h_A \notin G \\ \perp & \text{otherwise.} \end{cases} \tag{12}$$

Correspondingly, Bob generates his key as $K_B = g(h_B, M_A)$.

It is worth noting that the key K_A is represented by a k-bit string. In order to ensure that adjacent quantization sectors lead to bit strings that only differ by 1 bit, we can use e.g. gray coding when generating the actual bit strings for the keys.

Finally, let us verify that the fuctions m and g satisfy conditions (1), (5), and (3). It suffices to consider the reduced key space $\mathcal{K}' = \mathcal{K} \setminus \{\perp\} = \mathcal{S}$, as the case $K = \perp$ means we do not have a key that needs to be kept secret.

- **Reliability:** The bit error probability $\Pr(K_A[i] \neq K_B[i])$ depends on the choice of δ. In particular, the larger δ is, the lower we would expect the bit error to be. In other words, for a given ϵ, we can satisfy condition (1) by choosing a large enough δ.
- **Secrecy:** The message M_A (or M_B) only describes the relative position of the channel estimate h_A (or h_B) within the quantization sector corresponding to symbol S, but contains no information about the symbol S. Therefore, M_A and M_B are independent from the key $K_A \in \mathcal{K}'$, and condition (5) holds for all ϵ.
- **Uniformity:** Due to the circular symmetricity of the guard zones and the fact that the output of q is uniformly distributed, the keys $K_A \in \mathcal{K}'$ will also be uniformly distributed. Thus, the keys satisfy condition (3) for all ϵ.

Notice that because conditions (5) and (3) are satisfied for all ϵ, we need only concern ourselves with the ϵ in condition (1), which therefore corresponds to a desired bit error probability.

Unlike a lot of related research, we do not use error-correcting codes to further improve $\Pr(K_A[i] = K_B[i])$. This is justified by the fact that we do not require $\Pr(K_A[i] = K_B[i])$ to be arbitrarily close to 1, as the use of one-time pad encryption allows for some bit errors in the keys. Furthermore, using error correction codes would mean that more bits need to be communicated between Alice and Bob when the SNR of estimates h_A and h_B is smaller, or if we want to increase $\Pr(K_A[i] = K_B[i])$. In the scheme proposed in this paper, the amount of communication stays constant (1 bit per sample), and a worse SNR or higher $\Pr(K_A[i] = K_B[i])$ can be handled by simply increasing the guard size δ.

IV. Analytical results

In this section, we will investigate how $\Pr(K_A[i] \neq K_B[i])$ depends on the guard parameter δ. First of all, because K_A can be empty, $\perp$, and this case does not lead to any key bits being generated, we will condition $\Pr(K_A[i] \neq K_B[i])$ on the event that $h_A \notin G$ and $h_B \notin G$. Furthermore, we will find it easier to work with the key error probability $\Pr(K_A \neq K_B)$ instead of the individual bit errors. Therefore, the error probability of interest is

$$\Pr(K_A \neq K_B \mid h_A \notin G, h_B \notin G). \tag{13}$$

We will also slightly alter our model of h_A and h_B by assuming $h_A = h$ and $h_B = h_A + w_B - w_A$. This is not entirely equivalent to the original model discussed in Section II, but will yield results that are easier to read and will end up being usable in the original problem formulation.

The following proposition captures the sought after error probability as the quotient of two integrals which we will subsequently try to find good approximate bounds for.

Proposition 1. *The key error probability* $\Pr(K_A \neq K_B \mid h_A \notin G, h_B \notin G)$ *is equal to*

$$\frac{\int_{r_0}^{\infty} \int_{\phi_0(r)}^{\theta/2} P_{A\neq B, B\notin G}(h(r,\phi)) p_{R,\Phi}(r,\phi) d\phi dr}{\int_{r_0}^{\infty} \int_{\phi_0(r)}^{\theta/2} P_{B\notin G}(h(r,\phi)) p_{R,\Phi}(r,\phi) d\phi dr} \tag{14}$$

where $h = h(r,\phi)$ *is the channel coefficient* h *described by amplitude* r *and phase* ϕ,

$$P_{A\neq B, B\notin G}(h) = \Pr(K_A \neq K_B, h_B \notin G \mid h_A = h)$$
$$P_{B\notin G}(h) = \Pr(h_B \notin G \mid h_A = h)$$

$p_{R,\Phi}(r,\phi)$ *is the probability density function of the channel coefficient* h*, and*

$$r_0 = \frac{\delta}{2\sin(\theta/2)}, \quad \phi_0(r) = \arcsin(\frac{\delta}{2r}).$$

Proof: Skipped. ■

In the above result, we note first of all that the probability density function $p_{R,\Phi}(r,\phi)$ of $h(r,\phi)$ is

$$p_{R,\Phi}(r,\phi) = \frac{r}{2\pi P} e^{-\frac{r^2}{2P}} \tag{15}$$

which follows directly from our earlier established model for the channel coefficient h. Although this density function does not strictly speaking depend on the phase ϕ, we keep the notation $p_{R,\Phi}(r,\phi)$ to emphasize that it is the joint pdf of both the amplitude r and phase ϕ of $h = h(r,\phi)$.

For the two probabilities $P_{A\neq B, B\notin G}(h)$ and $P_{B\notin G}(h)$, it is not feasible to construct exact expressions. Instead, we will establish easier to compute upper and lower bounds. Before doing so, we will need the following definitions.

Definition 3. *Let* $x \in \mathbb{C}$ *be a point in the complex space. Then* $[x]_n \in \mathbb{R}$ *denotes the component of* x *that points in the direction of the line perpendicular to quantization border* g_n*. That is,*

$$[x]_n = \mathbf{vec}(x)^T \mathbf{vec}(e^{i(g_n \pm \pi/2)}). \tag{16}$$

where the sign of $\pi/2$ *is chosen such that* $[x]_n \geq 0$

Definition 4. *Given a point* $h = h(r,\phi)$*, we denote its minimum distance to each quantization border* g_n *by* $d_n(h)$*. As illustrated in Fig. 2, these distances can be calculated as*

$$d_n(h) = |r\sin((n-1)\theta - \phi)|. \tag{17}$$

Now we are ready to establish bounds for the probabilities $P_{A\neq B, B\notin G}(h)$ and $P_{B\notin G}(h)$ via the following two lemmas.

Lemma 1. *The probability* $P_{A\neq B, B\notin G}(h_A)$ *is upper bounded by*

$$\Pr([h_A - h_B]_1 > d_1(h_A) + \delta/2)$$
$$+ \Pr([h_A - h_B]_2 > d_2(h_A) + \delta/2) \tag{18}$$

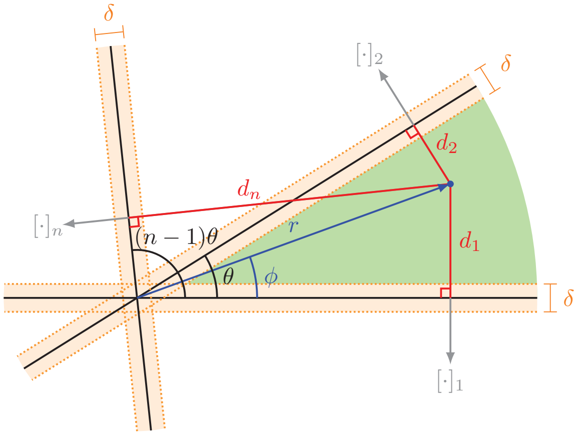


Fig. 2: The minimum distances d_n from a point $h = h(r, \phi)$ drawn as red lines.

Proof: First, let us define the following two events E_1 and E_2 w.r.t. a fixed h_A that belongs to quantization sector $g_1 \leq \phi_A \leq g_2$:

$$E_1(h_B|h_A) = \{h_B \text{ is on the other side of border } g_1.\}$$
$$E_2(h_B|h_A) = \{h_B \text{ is on the other side of border } g_2.\}$$

In the above, the *border* g_n is understood to mean the line that goes through the ray defined by angle g_n. Now we see that

$$\begin{aligned}
&P_{A\neq B, B\notin G}(h_A) \\
&= \Pr(K_A \neq K_B, h_B \notin G \mid h_A) \\
&= \Pr(E_1(h_B|h_A) \cup E_2(h_B|h_A), h_B \notin G \mid h_A) \\
&\leq \Pr(E_1(h_B|h_A), h_B \notin G \mid h_A) \\
&\quad + \Pr(E_2(h_B|h_A), h_B \notin G \mid h_A) \\
&\leq \Pr(E_1(h_B|h_A), h_B \notin G_1 \mid h_A) \\
&\quad + \Pr(E_2(h_B|h_A), h_B \notin G_2 \mid h_A) \\
&= \Pr([h_A - h_B]_1 > d_1(h_A) + \delta/2) \\
&\quad + \Pr([h_A - h_B]_2 > d_2(h_A) + \delta/2)
\end{aligned}$$

■

Lemma 2. *The probability $P_{B\notin G}(h_A)$ is lower bounded by*

$$1 - \sum_{n=1}^{2^{k-1}} \Pr(d_n(h_A) - \delta/2 \leq [h_A - h_B]_n \leq d_n(h_A) + \delta/2) \tag{19}$$

Proof: Following the same reasoning as in the previous Lemma, we can see that

$$\begin{aligned}
&P_{B\notin G}(h_A) \\
&= \Pr(h_B \notin G \mid h_A) \\
&= 1 - \Pr(h_B \in G \mid h_A) \\
&= 1 - \Pr(h_B \in \bigcup_{n=1}^{2^{k-1}} G_n \mid h_A) \\
&\geq 1 - \sum_{n=1}^{2^{k-1}} \Pr(h_B \in G_n|h_A) \\
&= 1 - \sum_{n=1}^{2^{k-1}} \Pr(d_n(h_A) - \delta/2 \leq [h_A - h_B]_n \leq d_n(h_A) + \delta/2).
\end{aligned}$$

■

The above Lemmas give bounds that hold for any arbitrary distribution of the random variable $[h_A - h_B]_n$. In our case, the random variable $[h_A - h_B]_n$ is normally distributed with mean 0 and variance $2N$. Hence, we can compute the probabilities $\Pr([h_A - h_B]_n > d_n(h_A) + \delta/2)$ and $\Pr(d_n(h_A) - \delta/2 \leq [h_A - h_B]_n \leq d_n(h_A) + \delta/2)$ in Lemmas 1 and 2 from the cumulative distribution function (CDF) of $\mathcal{N}(0, 2N)$.

By using the upper bound in Lemma 1 and the lower bound in Lemma 2 we can compute an upper bound for the error probability in Proposition 1. Notice that in the case of normally distributed noise as considered in this paper, we will not have closed form expressions for either of the two bounds, and hence we will not have a closed form bound for the integral in Proposition 1 either. Therefore, in the simulations in the next section, the upper bound of the error probability will simply be computed via numeric integration and table lookup.

Finally, recall that the probability we were originally interested in was the bit error probability. From the above results, we get an *approximate* upper bound by simply dividing our upper bound of $\Pr(K_A \neq K_B \mid h_A \notin G, h_B \notin G)$ by the number of bits k. As we will see in the next section, this will give us an upper bound on the bit error probability for sufficiently large δ.

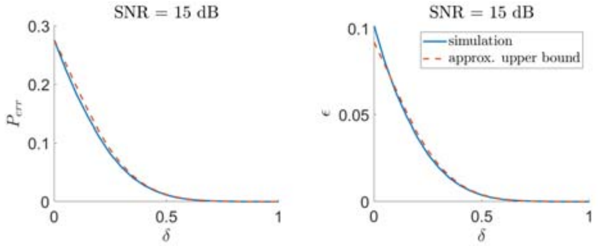
V. Simulation Results

In this section, we will analyze the proposed method via simulations in terms of (a) the bit error probability and (b) the bit generation rate. Note that the former is closely tied to our choice of the guard parameter δ, and the latter to the quantization parameter k. Hence, we will try to characterize the bit error probability in terms of δ, and the bit generation rate in terms of k.

For the bit error probability, we wish to compare our previously derived bit error probability bound to the true bit error probability. The analytical bound is computed as described in the previous section, utilizing numeric integration and table lookup where necessary. The true error is found via simulation, by performing a high number of key generation trials for randomly generated channel estimates h_A and h_B.

The bit error depends on both δ and k, as well as the variances P and N of the true channel h and its estimates h_A and h_B, respectively. In this case, we are only interested in the bit error as a function of δ, so we keep k, P and N constant. Equivalently, we can consider $\sqrt{P/N}$ constant, and we will refer to $\sqrt{P/N}$ as the *signal-to-noise ratio* (SNR) of the channel estimates. The SNR is expressed in the dB scale.

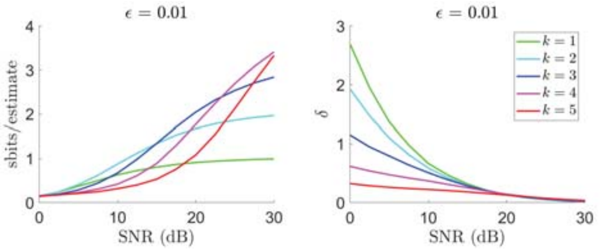
Fig. 3 shows the true error probability and corresponding error bound when $k = 3$ and SNR = 15 dB. Fig. 3a shows the key error probability, and Fig. 3b the bit error probability.



(a) key error

(b) bit error

Fig. 3: The error probability for $k = 3$ and SNR=15 dB. Solid blue line shows the true error probability, and red dashed line the approximate upper bound derived previously. (a) shows the key error probability (i.e. the probability that any bit in the keys differs), and (b) the bit error probability.



(a) sbits/estimate

(b) δ

Fig. 4: (a) The number of key bits generated on average, and (b) the corresponding δ as a function of SNR for desired bit error probability $\epsilon = 0.01$. Different colors correspond to different choices of k.

Recall that the model we had used to derive the upper bound of the key error was slightly different from the true model. Hence, this upper bound is not a true upper bound but merely an approximate upper bound. Nonetheless, Fig 3a suggests that, for this particular choice of parameters the key error bound acts as a true upper bound, and a fairly tight one at that. As for the bit error, we notice that for large enough δ ($\delta > 0.2$), the bit error bound appears to be a fairly tight upper bound on the bit error as desired. Recall that the bit error bound was simply the key error bound divided by k, i.e. we assume that the keys K_A and K_B will differ in at most 1 bit. The results suggest that this is indeed the case for sufficiently large δ.

Next, let us find out how many secret bits we can expect to generate per channel estimate using the proposed method. Because this will depend on the bit error probability, we first set a *desired* bit error probability ϵ, which uniquely defines δ. This δ is found experimentally. Note that for each channel coefficient h we estimate, we will generate either k or 0 bits. The latter occurs when $h_A \in G$ or $h_B \in G$. Therefore, the number of bits we can generate on average is simply $k \cdot \Pr(h_A \notin G, h_B \notin G)$. Clearly, the number of bits we can generate is upper bounded by k, and we increase the potential number of bits we can generate by increasing k. On the other hand, a larger k will decrease $\Pr(h_A \notin G, h_B \notin G)$.

Fig. 4a shows the number of key bits generated on average for different k, and a desired bit error probability of $\epsilon = 0.01$. Fig. 4b shows the corresponding experimentally derived values of δ. As one might expect, the optimal choice of k will depend on the SNR, with e.g. $k = 3$ being a good choice for SNRs in the range 10-20 dB. Generally, the higher the SNR is, the higher we should choose k.

VI. Conclusion

In this paper, we have proposed a secret key generation scheme that exploits the phase of the channel fading coefficient in a Rayleigh fading channel. This scheme was explicitly designed to generate keys to be used in one-time pad encryption. An example of where such encryption can be used is in distributed inference applications in IoT where sensors send short messages infrequently to a fusion center. By focusing on this specific type of keys, we were able to trade off a somewhat higher bit error for fewer bits needed for communication compared to other similar methods.

The proposed method has two adjustable parameters: the quantization parameter k, and the guard parameter δ. The former is closely tied to the bit generation rate, and the latter to the bit error probability. Given a desired bit error probability ϵ and working environment with known SNR $\sqrt{P/N}$, there is an optimal k and corresponding δ.

An approximate upper bound for the bit error probability was derived. This bound appears to be a (fairly tight) upper bound for δ large enough. Unfortunately, the bound does not have a closed form expression, making it hard to analyze exactly how it depends on the parameters δ, k, P, and N. Future work could include finding an alternative upper bound on the bit error that can be expressed in closed form.

References

[1] C. E. Shannon, ”Communication theory of secrecy systems,” *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656-715, Oct. 1949.

[2] U. Maurer, ”Secret key agreement by public discussion from common information,” *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[3] R. Ahlswede and I. Csiszár, ”Common randomness in information theory and cryptography. I. Secret sharing,” *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, July 1993.

[4] J. E. Hershey, A. A. Hassan, and R. Yarlagadda, ”Unconventional cryptographic keying variable management,” *IEEE Trans. Commun.*, vol. 43, pp. 3–6, Jan. 1995.

[5] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu, ”Cryptographic key agreement for mobile radio,” *Digital Signal Process.*, vol. 6, pp. 207–212, 1996.

[6] M. Bloch, J. Barros, M. R. D. Rodrigues, S. W. McLaughlin, ”Wireless Information-Theoretic Security,” *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2515-2534, June 2008.

[7] A. Sayeed, A. Perrig, ”Secure wireless communications: Secret keys through multipath,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[8] Y. Liu, S. C. Draper, A. M. Sayeed, ”Exploiting Channel Diversity in Secret Key Generation from Multipath Fading Randomness,” *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1484-1497, October 2012.

[9] H. Liu, Y. Wang, J. Yang, Y. Chen, ”Fast and Practical Secret Key Extraction by Exploiting Channel Response,” in *2013 Proceedings IEEE INFOCOM*, pp. 3048-3056, 2013.

[10] T. Halme, V. Koivunen, H. V. Poor, ”Nonparametric distributed detection using bootstrapping and Fisher's method,” in *52nd Annual Conference on Information Sciences and Systems (CISS)*, March 2018.