Secret Key Sharing and Rateless Coding for Practical Secure Wireless Transmission

Wei Liu†1, Chunjie Duan‡, Yige Wang‡, Toshiaki Koike-Akino‡, Ramesh Annavajjala‡, and Jinyun Zhang‡
† Department of EECS, Syracuse University, Syracuse, NY 13244, USA
‡ Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
Email: wliu28@syr.edu, {duan, yigewang, koike, annavajjala, zhang}@merl.com

***Abstract*—We discuss a secure wireless communication scheme, focusing on designing two major components: the key generation and the coding scheme. To achieve high key matching rate, we propose a feed-forward and feed-back quantization. The proposed scheme offers 1 dB improvement over the best known schemes. We also propose a universal quantization scheme with feed-forward/feed-back and show that its performance is the same as, or better than the other schemes which require prior distribution information. For rate-adaptive coding, we propose the use of rateless codes. Our evaluations show that the rateless code can offer significant performance gain over a low-density parity-check (LDPC) code. Moreover, we implement a soft input rateless decoder which offers additional gains. The overall security performance of our design based on these two components significantly outperforms existing designs.**

Keywords - Secret key generation; Wireless communications; Quantization; Rateless codes; Rate compatible codes.

I. Introduction

A security system is only as good as its weakest part, frequently, the Key Management System (KMS), which consists of the key management, key derivation, storage and distribution [20]. Unfortunately, designing a good KMS is an extremely hard problem and not all designers agree on how to construct it. This problem is exacerbated with the current trend of moving into the “Internet of Things” and ”cyber physical systems”, where large scale, complex, ad-hoc, often infrastructure-less wireless sensor networks and the broadcast nature of the channels makes the problem much harder to tackle.

Most commonly used key management techniques are based on public key cryptography and requires a Public key Infrastructure (PKI). PKI is generally not suitable for ad-hoc systems where a) infrastructure is not guaranteed; b) key derivation functions are computation intensive and therefore cannot be carried out in low power and low cost devices such as wireless sensor nodes; and c) designing a key storage scheme that survives frequent system and node reboot is also challenging.

Physical-Layer Security (PLS) has been proposed [23, 24] with the hope to address this problem. It is envisioned that by establishing security at the physical or link layer, a network wide KMS is no longer needed, or can be greatly simplified and we can potentially remove the requirement of having a PKI in the network, that would result in a simple and cost/power efficient yet secure network.

Most of the PLS schemes proposed so far fall into two categories: a) generating secret keys from correlated sources based on wireless channel reciprocity, and then applying traditional cryptography and b) adapting transmission rate to the information-theoretic channel capacity to achieve a positive secrecy capacity.

In the former category, where Alice and Bob try to generate a pair of secret keys based on correlated observations X^n and Y^n respectively, as illustrated in Fig. 1. Assume that Alice and Bob can communicate through an unauthenticated public channel which might be observed by an eavesdropper, Eve, who might have side information Z^n which can be correlated with X^n and Y^n. The information-theoretical study of this problem was provided by Maurer [1] and Ahlswede and Csiszár [2], where the secrecy capacity defined as the maximum key generation rate is given for the special case when Eve has no side information Z^n, and lower bounds and upper bounds on the secrecy capacity are provided for general cases. While the theoretical aspects of this problem are well understood, there is a growing interest in designing practical secret key generation algorithms to approach the secrecy capacity of the key generation rate.

One way to design such algorithms is to exploit the inherent randomness in the wireless channel between two nodes as the source for extracting secret key sequences [3–12]. The security of these schemes relies on the *reciprocity principle* of the radio wave propagation which states that the multipath properties of the radio channel such as channel gains, phase shifts and delays at any point in time are identical in both direction of the communication link [8]. In addition, these properties are intrinsically *spatially specific* in a multipath radio environment due to the scatter effects. An eavesdropper at a third location more than a few wavelengths away from the two legitimated users will observe a different and uncorrelated radio channel [13]. As a result, the two legitimated users can generate a secret key based on the shared common randomness which is unavailable to the eavesdropper. Among these schemes, channel gain information [4, 5, 7–12] is the most commonly used. Most of these existing algorithms on key generation from channel gain measurements consist three steps: quantization, information reconciliation [14], and privacy amplification [15]. The purpose of quantization is to convert the real channel measurements into binary bit strings. Information reconciliation is aimed at generating an identical random sequence between the two legitimated users by communicating through the public channel and privacy amplification is used to extract a perfect secure key from the identical random sequence agreed in the information reconciliation stage. As the foundation of the whole process,

[1] This work is done during the author’s internship at MERL

designing a good quantization algorithm is crucial. In this paper, we focus on the quantizer design. The purpose is to generate two binary sequences at two legitimated users with bit mismatch rate as small as possible.

Traditional quantization schemes such as equiprobable quantization [4] and level crossing quantization [7] have the limitation of high bit mismatch probability or low key generation rate. Moreover, these quantization schemes do not take advantage of the public channel. Recently, Wallace *et al.* [16] proposed a new quantization algorithm utilizing a one-way communication over the public channel and achieved a significant improvement in the bit mismatch probability. The over-quantization algorithm proposed in [12] used a similar idea to achieve higher key generation rate within 1.1 bits from the secret key capacity.

Schemes in the latter category are all based on the principle that data transmitted at a rate exceeding the channel capacity cannot be decoded by the receiving end. Many papers derive the secrecy capacity for various channels including wire-tap channels [21], broadcast channels [22], multiple access channels [26], relay broadcast channels [24], fading channels [23], *etc*. Fundamentally, to achieve a positive secrecy capacity, it is required that the channel capacity of the eavesdroppers be lower than that of the intended receivers. This limits the practicality of these types of schemes, as it is nearly impossible to guarantee such a condition in wireless environment. Some attempts have been made to address this. For instance, Pinto et al. showed in [27] that if physical security can be guaranteed (e.g., a small area around the intended receiver is secured from eavesdroppers), secrecy capacity is improved. Advanced transmission schemes are also proposed to improve the secrecy capacity, e.g., the secrecy capacity improvement with MIMO precoding is analyzed in [25]. Beamforming and artificial jamming noise injection are shown to improve the secrecy capacity [28], with the assumption that certain knowledge of the eavesdropper's channel is available to the transmitter.

To the best of our knowledge, however, there is yet a practical way of guarantees that legitimate receivers have favorable channels, and the research in this area suffers from being limited to mostly just theoretical analysis.

In this paper, we propose a secure transmission scheme by combining both ideas. The nodes first carry out key generation using the quantization schemes that offers high bit matching rate (BMR), and then carry out secure transmissions by scrambling data to create an artificial channel that is favorable to the legitimate node pairs, and then apply rate adaptive coding with a rate tightly coupled to instantaneous capacity.

The first focus of the design is the key generation process, in particular, quantization schemes that allow an independently generated keys pair to have high BMR. Motivated by the theoretical analysis in [1, 2], we propose a quantization algorithm by exploiting the public channel through two-way communications; more specifically, feed-forward and feed-back quantization. For the case that the channel measurements are Gaussian distributed, the proposed algorithm improves around $1\,\mathrm{dB}$ at high signal-to-noise ratio (SNR) over the best known scheme [16]. Moreover, we propose a universal quantization scheme with feed-forward and feed-back for the general case when the two end nodes do not know the prior distribution. It is demonstrated through simulations that the proposed universal scheme works asymptotically close to the case with known prior information.

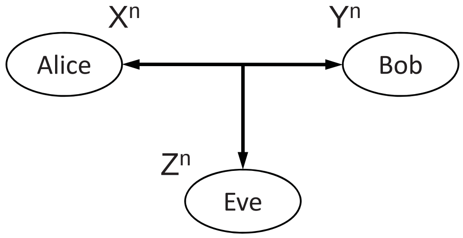

Fig. 1. Secret key sharing between Alice and Bob.

The second focus of the design is the transmission scheme, where we propose applying channel scrambling and rateless coding to achieve the maximum achievable secrecy capacity.

The key management of PLS scheme proposed in this paper is simple, as the link keys are generated locally between two communicating nodes and are used locally. A node only need to store and update the keys to its immediate (one hop) neighbors. PLS schemes such as the one discussed in this paper can be seamlessly integrated with existing security mechanisms in the upper layer to enhance the overall security level of wireless systems.

The rest of the paper is organized as follows. Section II gives an overview of the transmission scheme and the node design. In Section III, we introduce the system model for key generation. Section IV introduces the new quantization algorithm which uses the feed-forward and feed-back scheme. The universal quantization scheme is detailed in Section V. Coding with rateless codes is discussed in Section VI. We conclude this paper in Section VII.

II. Secure Transmission with Mismatching Keys

The transmitter and the receiver under consideration have the structure shown in Fig. 2. Both legitimate nodes, Alice and Bob, implement a *Key Generator* block. Each key generator produces a secret key string, i.e., K_A and K_B. The transmitter encodes a message M with an inner code and then scrambles the coded message C with its secret key, K_A. The scrambled data S will be transmitted directly, or alternatively, coded with an outer code before being transmitted over the wireless channel.

Correspondingly, a decoder implemented in the receiver to decode the outer code is necessary. The receiver descrambles the output of the outer decoder R_B with the secret key generated by its own key generator, K_B, and feed the descrambled message C_B to its inner code decoder. The receiver's inner decoder decodes the message. We also assume Eve has full knowledge of the transmission scheme and is equipped with the same functional block as in a legitimate receiver. Eve may be able to produce a key, K_E, descramble and decode the receiver message.

In the proposed receiver, rate adaptive codes are used as the inner code to maximize the secrecy capacity. We focus on the performance of rateless codes in this paper. During

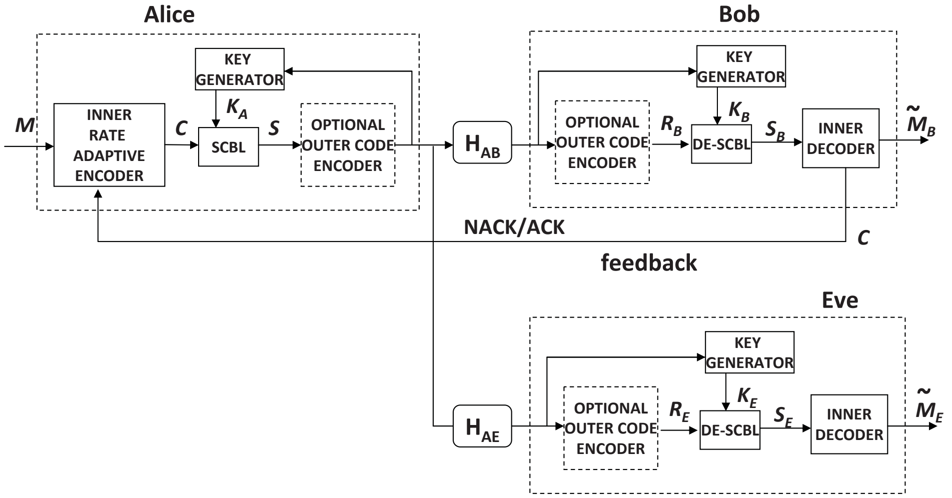

Fig. 2. Transmitter and receiver structures.

the transmission of a message M, the transmitter's encoder continues sending out the coded stream S until the receiver has successfully decoded the message, $\tilde{M}$. A feedback channel is needed for the receiver to send an acknowledgement (ACK) or a negative ACK (NACK) back to the transmitter. The mechanism of rateless codes guarantees that the rate of the transmission matches the instantaneous capacity.

The process of key generation needs to be carried out before any secure data transmissions. Alice and Bob first perform key generations. Both nodes transmit sounding signals alternatively to enable the other side to perform channel estimations. Key pairs need to be updated periodically but not necessary at every data transmission. The update frequency is determined by the overhead. In most cases, the data packets are preceded by preambles so that the receiver can estimate the channel for coherent detections. Therefore, this allows the key generation/update be integrated into the transmission period and further reduces the overhead.

The overall secrecy capacity of such a scheme is determined by two factors:

1) The difference of the key matching rates
2) The closeness of the coding rate

The former item dictates the *maximum* secrecy capacity while the latter item affects the *achievable* secrecy capacity. They both need to be optimized such that overall security can be maximized. The following sections describe how they can be optimized separately.

III. Secret Key Generation

A. Secret Key Generation Based on Channel Measurements

Alice and Bob want to share a secret key by measuring the channel response between them. With a channel reciprocity, the channel estimates of Alice and Bob are highly correlated. We can model Alice and Bob's observations, X^n and Y^n, as independent and identically distributed (*i.i.d.*) n repetitions of dependent random variables, X and Y, from a joint distribution $f(X, Y)$. Based on those correlated observations X^n and Y^n, Alice and Bob generate a secret key by communicating over a public channel, while the messages transmitted through the public channel may be observed by eavesdroppers. We assume that the eavesdropper, Eve, can only observe the channel without any message modifications, i.e., a passive eavesdropper. The messages transmitted through the channel (possibly two way) are denoted as $\mathbf{V}$.

In this paper, we focus on the case when X and Y are jointly Gaussian; more specifically,

$$X = G + W_A, \qquad Y = G + W_B, \tag{1}$$

where $G \sim \mathcal{N}(0, P)$, $W_A \sim \mathcal{N}(0, N_A)$ and $W_B \sim \mathcal{N}(0, N_B)$ denote the channel response, the estimation error at Alice and the estimation error at Bob, respectively. For simplicity, we consider the case of $N_A = N_B = N$.

IV. Quantization with Feed-forward and Feed-back

In this section, we discuss a quantization-based key generation algorithm. We use a core quantization module based on a scalar equiprobable quantizer because of its lightweight and its property of maximum entropy.

A. Algorithm Description

Our secure key sharing algorithm consists of the following five steps.

1) Initialization phase: Alice and Bob agree on the quantization level L and the feed-forwarding level m. Both Alice and Bob use the same equiprobable scalar quantizer, which is designed for a zero-mean and unit-variance Gaussian random variable. For the L-level quantizer, the quantization boundary q_i to indicate L intervals, $(q_0, q_1], (q_1, q_2], \cdots, (q_{L-1}, q_L)$, is chosen such that [4]

$$Q(q_i) \triangleq \int_{q_i}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{x^2}{2}\Big) \mathrm{d}x = \frac{L-i}{L}, \tag{2}$$

for any $i \in \mathbb{Z}_L$, where $Q(\cdot)$ denotes the Gaussian tail function and $\mathbb{Z}_L \triangleq \{0, 1, \cdots, L-1\}$ denotes an integer set. Here, we set $q_0 = -\infty$ and $q_L = \infty$. Gray coding is used for mapping the quantizer indices to bits.

For generating feed-forwarding information, each quantization interval $(q_{i-1}, q_i]$ is further split into m sub-intervals, $(t_{i-1,0}, t_{i-1,1}], (t_{i-1,1}, t_{i-1,2}], \cdots, (t_{i-1,m-1}, t_{i-1,m}]$, where $t_{i-1,0} = q_{i-1}$ and $t_{i-1,m} = q_i$, such that each sub-interval $(t_{i-1,k}, t_{i-1,k+1}]$ for $k \in \mathbb{Z}_m$ has an identical probability of

$$Q(t_{i-1,k}) - Q(t_{i-1,k+1}) = \frac{1}{mL}. \qquad (3)$$

For each sub-interval, we index them by $0, 1, \cdots, m-1$ in an ascending order.

2) Channel sounding phase: Alice and Bob sequentially send known training signals to each other in order to measure the channel between Alice and Bob. Given each channel estimate X_i and Y_i for any $i \in \mathbb{N}_n$ (where n denotes the total number of measurement times), Alice and Bob individually quantize them into $\log_2(L)$-bit indices $K_A(i)$ and $K_B(i)$, using the L-level equiprobable scalar quantizer. Note that the quantization is done with a power normalization of $\beta = \frac{1}{\sqrt{P+N}}$ to have unity variance for quantizing data. The quantized data at $i \in \mathbb{N}_n$ is then given as

$$\begin{aligned} K_A(i) &= \big\{ j : \beta x_i \in (q_j, q_{j+1}] \big\}, \\ K_B(i) &= \big\{ j : \beta y_i \in (q_j, q_{j+1}] \big\}, \end{aligned} \qquad (4)$$

where $\mathbb{N}_n \triangleq \{1, 2, \ldots, n\}$ is a positive integer set. After n observations, Alice and Bob obtain $n \log_2(L)$-bit quantized information, $\mathbf{K}_A = [K_A(1), K_A(2), \cdots, K_A(n)]$ and $\mathbf{K}_B = [K_B(1), K_B(2), \cdots, K_B(n)]$, respectively.

3) Feed-forward phase: Alice generates a $\log_2(m)$-bit feed-forwarding data $V_a(i)$ from X_i such that $V_a(i)$ is the sub-interval index of the interval $K_A(i)$, more specifically, we can write $V_a(i)$ for any $i \in \mathbb{N}_n$ as follows:

$$V_a(i) = \big\{ j : \beta x_i \in (t_{K_A(i),j}, t_{K_A(i),j+1}] \big\}. \qquad (5)$$

Through the public channel, Alice then sends an $n \log_2(m)$-bit message, $\mathbf{V}_a = [V_a(1), V_a(2), \cdots, V_a(n)]$ towards Bob where each sub-interval index $V_a(i)$ can be expressed by its binary natural code representation with $\log_2(m)$ bits. We propose one additional step termed feed-back phase to further improve the bit mismatch rate.

4) Feed-back phase: Upon receiving the feed-forward information $\mathbf{V}_a$ from Alice and his own observation Y^n, Bob employs a maximum *a posteriori* probability (MAP) estimation of $\mathbf{K}_A$. In the MAP estimation, for each $Y_i = y_i$ and $V_{ai} = v_{ai}$ (for any $i \in \mathbb{N}_n$), Bob searches for the index j_i such that

$$j_i = \operatorname*{argmax}_{j \in \mathbb{Z}_L} \Pr\big(\beta X_i \in (q_j, q_{j+1}] | Y_i = y_i, V_{ai} = v_{ai}\big). \qquad (6)$$

With the MAP estimate j_i and the original quantized data $K_B(i)$, Bob generates feed-back information $V_b(i)$ which is set to be one if $j_i \neq K_B(i)$ and zero otherwise, for each $i \in \mathbb{N}_n$. The feed-back message $\mathbf{V}_b = [V_b(1), V_b(2), \cdots, V_b(n)]$ of n bits is sent to Alice through the public channel.

5) Key generation phase: Based on the feed-back message $\mathbf{V}_b$, Alice skips the corresponding $K_A(i)$ if $V_b(i) = 1$, for $i \in \mathbb{N}_n$, and sets the remaining as her secret key $\mathbf{K}_A$. Similarly, for each $i \in \mathbb{N}_n$, Bob also skips the corresponding indices $K_B(i)$ if $V_b(i) = 1$ and produce his secret key using the remaining bits.

B. Discussions

1) MAP estimation: We call the algorithm described above *MAP with feed-back* key generation algorithm. For the case that X^n and Y^n are jointly Gaussian as in (1), by noticing that $\Pr(V_{ai} = v_{ai} | Y_i = y_i)$ is a constant for given y_i and v_{ai}, we have

$$\begin{aligned} &\Pr(\beta X_i \in (q_{j-1}, q_j] | Y_i = y_i, V_a(i) = v_{ai}) \\ &\propto \Pr(\beta X_i \in (q_{j-1}, q_j], V_a(i) = v_{ai} | Y_i = y_i) \\ &= \Pr(\beta X_i \in (t_{j-1,v_{ai}}, t_{j-1,v_{ai}+1}] | Y_i = y_i) \\ &= Q\left(\frac{\frac{1}{\beta} t_{j-1,v_{ai}} - \mu_i}{\sigma_i} \right) - Q\left(\frac{\frac{1}{\beta} t_{j-1,v_{ai}+1} - \mu_i}{\sigma_i} \right), \end{aligned} \qquad (7)$$

where $\mu_i = \rho y_i$, $\rho = \sqrt{\frac{P}{P+N}}$ and $\sigma_i = \sqrt{(P+N)(1-\rho^2)}$. The last equality comes from the fact that $(X_i | Y_i = y_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

Hence, the MAP estimation in (6) is equivalent to find the index $j \in \mathbb{Z}_L$ such that interval $(t_{j,v_{ai}}, t_{j,v_{ai}+1}]$ is the closest one to $\mu_i = \rho y_i$ in the sense of Euclidean distance. Or equivalently, Bob will decide $j_i = j \in \mathbb{Z}_L$ if $y_i \in (\bar{q}_j, \bar{q}_{j+1}] = \frac{1}{\rho}(\tilde{q}_j, \tilde{q}_{j+1}]$, where

$$\tilde{q}_j = \begin{cases} \frac{1}{2}(t_{j-1,v_{ai}+1} + t_{j,v_{ai}}), & 1 \leq j \leq L-2, \\ -\infty, & j = 0, \\ \infty, & j = L-1. \end{cases} \qquad (8)$$

We remark here that the process of finding new quantization regions for Bob according to feed-forward bits is slightly different from the one described in [16]. In [16], a middle point as in (8) is chosen as the new quantization region while the optimal one should multiply a coefficient of $\frac{1}{\rho}$.

The procedure of finding j_i is illustrated in Fig. 3 for $L = 4$ and $m = 2$. Initial quantization codebook for both Alice and Bob is $(-\infty, q_1], (q_1, q_2], (q_2, q_3], (q_3, \infty)$. The observations of Alice and Bob are X and Y, respectively. In this example, Alice will quantize her observation into a two-bit information $K_a = [0, 1]$ and send a feed-forwarding data $V_a = 0$ towards Bob. By observing $V_a = 0$, Bob calculates his MAP codebook $(-\infty, \bar{q}_1], (\bar{q}_1, \bar{q}_2], (\bar{q}_2, \bar{q}_3], (\bar{q}_3, \infty)$ using (8), and finds $j_i = 1$ to generate the feed-back information.

2) Security: Transmitting the messages $\mathbf{V}_a$ and $\mathbf{V}_b$ through the public channel does not result in release of information about the secret key. This is because $V_a(i)$ is equiprobable, namely $\Pr(V_a(i) = v_{ai}) = 1/L$, and $V_b(i)$ indicates only the positions of the skipped bits.

3) Simulation results: Fig. 4 shows the performance of the algorithm described above (with 10^6 runs). As we can see from this figure, interaction between two nodes significantly improves the key agreement performance in bit mismatch rate at high SNR (higher than 10 dB), where SNR is defined as P/N. The introduction of feed-back phase brings approximately 1 dB gain over the scheme without feed-back. To compare the performance with the result by Wallace *et al.* [16], we also plot MAP without feed-back scheme in which the feed-back step is dropped in the algorithm. The 1-bit feed-forwarding in [16], though slightly different from the MAP without feed-back scheme as discussed above, achieves almost the same performance as the optimal MAP scheme. It is shown by

ISBN: 978-1-61208-144-1

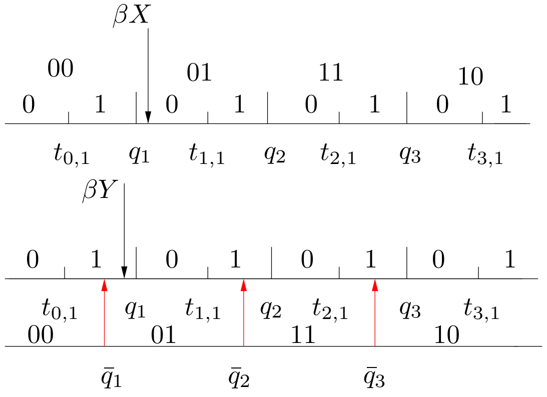

Fig. 3. MAP with feed-back quantizer.

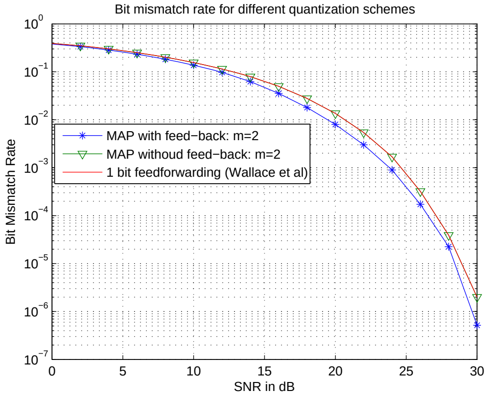

Fig. 4. Bit mismatch rate for 2-bits quantization.

simulation that the average number of skipped bits is relatively small (less than 15%).

V. Universal Key Generation Algorithm

The algorithm described in the previous section assumes prior knowledge of the joint distribution of X^n and Y^n is available to the nodes. However, in most of the practical applications, we might not have such information. In this section, we deal with this issue by introducing a universal key generation scheme in which no prior knowledge about the underlying joint distribution is required. The core quantization module is a universal equiprobable quantizer described below.

A universal equiprobable scalar quantizer for an arbitrarily distributed random sequence X^n is a set of intervals $[q_0, q_1), [q_1, q_2), \cdots, [q_{L-1}, q_L]$, and the boundary q_i (for $i \in \mathbb{Z}_L$) can be calculated as follows. Let us sort $X^n = [X_1, X_2, \cdots, X_n]$ into an ascending order $[X_{(1)}, X_{(2)}, \cdots, X_{(n)}]$ and dividing it into L intervals with each interval containing equal number of X_i's, and then pick the boundary points in an ascending order as $q_0, q_1, \cdots, q_L$, that is, $q_0 = X_{(1)}, q_1 = X_{(\lfloor n/L \rfloor)}, \cdots, q_L = X_{(n)}$.

A. Algorithm Description

The universal key generation algorithm is described in the following steps. We suppose that Alice and Bob pre-agree on the quantization level L and the feed-forward data level m.

1) Channel sounding phase: For each X_i and Y_i ($i \in \mathbb{N}_n$), Alice and Bob quantize their estimated channel information into $\log_2(L)$-bit data $K_A(i)$ and $K_B(i)$ using the universal equiprobable scalar quantizer described above. Gray coding is used for mapping the quantizer indices to bits. Notice that, Alice and Bob's quantization codebook might not be the same. After n channel measurements, Alice and Bob obtain an $n\log_2(L)$-bit data $\mathbf{K}_A = [K_A(1), K_A(2), \cdots, K_A(n)]$ and $\mathbf{K}_B = [K_B(1), K_B(2), \cdots, K_B(n)]$.

2) Feed-forwarding phase: Alice generates feed-fowarding data by using the universal equiprobable quantizer to indicate m-level sub-intervals $[q_{i-1}, t_{i-1,1}), [t_{i-1,1}, t_{i-1,2}), \cdots, [t_{i-1,m-1}, q_i)$. Let us index the sub-intervals by $0, 1, \cdots, m-1$ for each sub-interval in an ascending order. Alice then sends an $n\log_2(m)$-bit message, $\mathbf{V}_a = [V_a(1), V_a(2), \cdots, V_a(n)]$ towards Bob through the public channel.

3) Feed-back phase: For each $i \in \mathbb{N}_n$, using the feed-forward data $V_a(i) = v_{ai}$, Bob decides his estimation of Alice's index by $j_i = j \in \mathbb{Z}_L$ if $y_i \in (\bar{q}_j, \bar{q}_{j+1})$, where

$$\bar{q}_j = \begin{cases} \text{median}[t_{j-1,v_{ai}+1}, t_{j,v_{ai}}), & 1 \le j \le L-2, \\ -\infty, & j = 0, \\ \infty, & j = L-1. \end{cases} \qquad (9)$$

Here, the median operation of an ordered set $(t_{j-1,v_{ai}+1}, t_{j,v_{ai}})$ as in (9) can be implemented by looking into the ordered sequences $[X_{(1)}, X_{(2)}, \cdots, X_{(n)}]$ and picking the middle point such that it divides interval $(t_{j-1,v_{ai}+1}, t_{j,v_{ai}})$ into two subregions with equal number of sequences. Notice that this is a little different from (8) where a Euclidean middle point of two boundaries of $(t_{j-1,v_{ai}+1}, t_{j,v_{ai}})$ is chosen. Bob then sets $V_b(i) = 1$ if $j_i \neq K_B(i)$ and $V_b(i) = 0$ otherwise, for each $i \in \mathbb{N}_n$, and sends an n-bit feed-back message $\mathbf{V}_b = [V_b(1), V_b(2), \cdots, V_b(n)]$ to Alice through the public channel.

4) Key generation phase: Using the feed-back message $\mathbf{V}_b$, Alice skips the corresponding $K_A(i)$ if $V_b(i) = 1$ (for each $i \in \mathbb{N}_n$), and sets the remaining as her secret key. As for Bob, for each $i \in \mathbb{N}_n$, he skips the indices $K_B(i)$ if $V_b(i) = 1$ and sets it as his secret key.

B. Simulation results

Fig. 5 plots the performance of the universal key generation algorithm described above with 1-bit (i.e., $m = 2$) feed-forwarding (with 10^5 runs). Here, we assume the underlying joint distribution of X and Y is given by (1). To compare the performance, we also present the result of MAP with feed-back algorithm with $m = 2$ as in the previous section. As we can see from the figure, the universal scheme works almost as well as the MAP with feed-back scheme. Fig. 6 shows the performance of the universal key generation algorithm when the underlying source is uniformly distributed and the estimation noise is assumed Gaussian. For this case, the universal scheme outperforms the MAP with feed-back algorithm designed for Gaussian priors as in Section III.

VI. Secure Coding with Rateless Codes

If a positive secrecy capacity can be established, it is necessary that the transmitter shall transmit at a rate R that

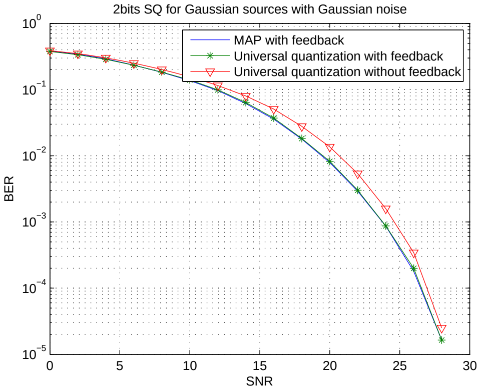

Fig. 5. BER for 2-bits universal quantization for Gaussian source.

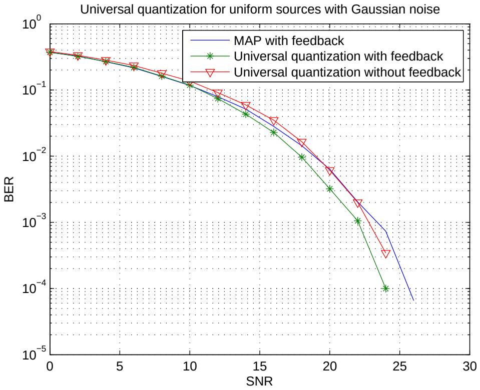

Fig. 6. BER for 2-bits universal quantization for uniform source.

equals the C_{AB} to achieve the allowed maximum secrecy capacity. Unfortunately, it is difficult to estimate accurately C_{AB}, and moreover, the instantaneous capacity $C_{AB}(t)$ varies.

Automatic Repeat Request (ARQ) is generally used to deal with instantaneous channel capacity variation. However, it is inefficient as information from previous transmission is thrown away. Hybrid ARQ (HARQ) using rate-compatible codes combines the original packet and the retransmitted packet(s) and is slightly more efficient. The issue with typical rate compatible code is the large granularity of the effective rates, For instance, a convolutional code may be punctured to obtain rates of $R = 7/8$, $R = 3/4$ and $R = 1/2$. Certain degree of secrecy capacity is sacrificed because of the coarse granularity.

Rateless codes, on the other hand, allows the transmitter to gradually reduce the rate. Combining it with HARQ, we can theoretically match the transmission rate to the exact instantaneous capacity as the number of bits transmitted can be arbitrarily small and we can adjust the rate at a much finer granularity.

A. Channel Scrambling for Positive Secrecy Capacity

To obtain positive secrecy capacity, the transmitter simply scrambles the coded data stream with the key it produced using the scheme discussed above. The received stream is descrambled with the receiver's key. This is similar to a one-time pad encryption. Our scheme, however, does NOT require keys to be exactly the same.

Note that in the case of a perfectly matched key pairs, the output after descrambling is the original received message. Mismatching bits in the keys will result in erroneous bits and the scrambling-descrambling process is equivalent to a binary symmetric channel (BSC), and the key mismatch rate is the crossover rate of the BSC channel. Because the eavesdropper's channel has zero, or low correlation with the channel between Alice and Bob, the crossover rate is much higher, and hence a much worse channel for the eavesdropper.

The artificial BSC channels created by scrambling-descrambling, can be combined with the wireless channels between nodes, or treated separated if we insert an outer channel code. For clarity, we consider the wireless channel noise free in the following discussion.

B. Overview of Rateless Codes

Rateless codes, also known as Fountain codes, are a class of codes with no fixed coding rate. The transmitter can generate a potentially limitless sequence of encoding symbols from a given set of information symbols. The receiver starts to decode after receiving a sufficient number of symbols from the channel. If it fails decoding, it will collect more symbols and start decoding again. The process is repeated until decoding succeeds. The transmitter keeps sending more symbols until receiving an ACK of successful decoding from the receiver. The number of symbols required for successful decoding depends on the quality of the channel. Rateless codes are especially useful for those cases where channel statistics are not known and fixed-rate codes do not work well.

The first class of practical rateless codes, Luby Transform (LT) codes [17], is invented in 1998. LT codes can be represented by Tanner graph and decoded using the belief propagation algorithm, like low-density parity-check (LDPC) codes. Given k information symbols, an encoding symbol of LT codes can be generated by first picking a degree d at random according to some distribution. Then select d distinct information symbols uniformly and XOR them to form an encoding symbol. The transmitter and the receiver should share the same random seed so that the decoder could construct the same code graph as the encoder. Luby shows the Robust Soliton distribution has excellent performance on erasure channel, but the disadvantage is that the decoding complexity is high, i.e., $O(k \ln k)$.

One problem with LT codes is that they exhibit an error floor phenomenon. This can be fixed by Raptor codes proposed by Shokrollahi [18], which combine LT codes with outer LDPC codes. Raptor codes show no noticeable error floors, however their rate is slightly bounded away from capacity.

C. Code Parameters and Decoding Scheme

In this paper, we focus on LT codes with the following degree distribution [18].

$$\begin{aligned}\mu(x) &= 0.007969x + 0.493570x^2 + 0.166220x^3 + 0.072646x^4 \\ &\quad + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 \\ &\quad + 0.055590x^{19} + 0.025023x^{65} + 0.0003135x^{66}. \qquad (10)\end{aligned}$$

The main advantage of this distribution is that its decoding complexity grows only as $O(k)$. However this leads to a small fraction of information symbols that are not involved in any encoding symbols. Then the bit error rate does not go to zero even when k goes to infinity. Nevertheless in practice, LT codes with the degree distribution in (10) still work well.

Decoding LT codes is similar to LDPC codes. Since information symbols are not transmitted, at the decoder side, they do not have any observation, thereby no *a priori* information. Different from information symbols, encoding symbols have *a priori* information, which is denoted by log likelihood ratios (LLRs) [29]:

$$L = \ln \frac{\Pr(s=1|r_c, r_k)}{\Pr(s=0|r_c, r_k)}, \tag{11}$$

where r_c is the observation from the channel and r_k is the observation from the key. An *a priori* information is used for the initialization of decoding. Let the LLRs of received data from the channel be L_c and the LLRs of the key be L_k. Then, we have [29]

$$L = 2\tanh^{-1}\left(\tanh\left(\frac{L_c}{2}\right)\tanh\left(\frac{L_k}{2}\right)\right). \tag{12}$$

If the reliability of key is unknown, then K_b can be treated as error free, i.e., L_k is infinity. For this case, (12) can be rewritten as

$$L = (1-2K_b)L_c, \tag{13}$$

where K_b is the corresponding bit in the key string of the receiver.

D. Simulation Results

We consider a time-variant channel and compare the performance of a fixed-rate LDPC code and an LT code with the degree distribution in (10). In this simulation, we assume the channel between Alice and Bob is a binary symmetric channel (BSC) with crossover probability p. To consider the time-variant property in wireless channels, we assume that p is uniformly distributed over an interval of $[0.01, 0.06]$, and is fixed for every 100 blocks while keep changing during the entire simulation (10^5 blocks).

In each block, 1000 bits are transmitted using either an LT code with or a fixed-rate LDPC code, whose degree distribution is optimized according to [19]. To make fair comparison, the parameters of the time-variant channel are the same for both codes. This can be achieved by saving the values of p in the first round of simulation and use it directly in the second round.

If decoding fails, the scheme using LT codes can request more bits from the transmitter to achieve an arbitrary small probability of decoding error. Hence, a reasonable procedure is to set a retransmission threshold n_t at the decoder side. If the total number of received bits exceeds this threshold and there are still bit errors, the decoder will report a block error.

Fig. 7 shows the simulation result of the two codes, wherein the y-axis is the probability of block error and the x-axis is the inverse of the code rate. For LDPC codes, we set a rate of 0.65, 0.6 or 0.55, with an optimization degree [19]. For LT codes, the rate is calculated using the average code-length in the entire 10^5 blocks. Note that, if a block error is reported, n_t is chosen as the code-length in that block and a block error is collected. In this figure, the red curve corresponds to the case where the LLRs of the key are known while the blue one corresponds to the case without key reliability information. As we can see from the figure, the LT code achieves better performance. The probability of block error converges to zero much faster than the fixed-rate LDPC code. Moreover, key reliability information helps decoding and hence achieves better performance. The same observation can be made in another simulation where p is assumed to be uniformly distributed in $[0.06, 0.11]$ as shown in Fig. 8.

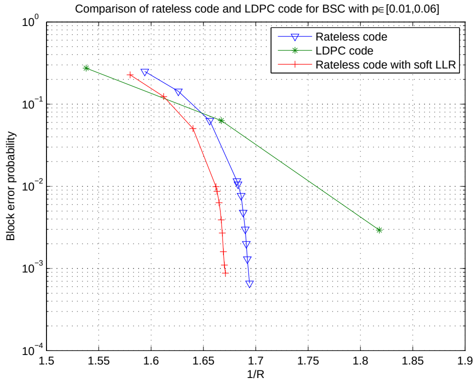

Fig. 7. Comparison of LT code and LDPC code for $p \in [0.01, 0.06]$.

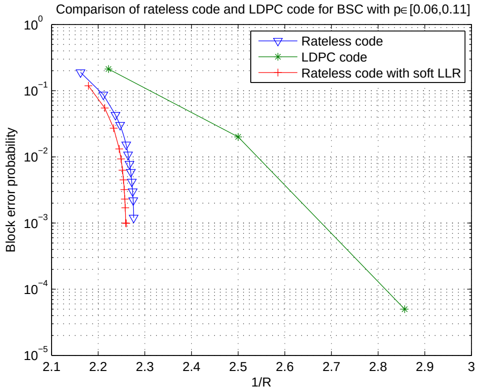

Fig. 8. Comparison of LT code and LDPC code for $p \in [0.06, 0.11]$.

VII. Conclusion

We described a transceiver design which can improve physical-layer security for practical wireless communications. The design combines the channel reciprocity based key generation and rate-adaptive coding. The design does not require a perfectly matched key pair. By scrambling the transmitted data and de-scrambling the received data with the independently generated keys, we guarantee the legitimate receiver to have a preferable channel compared to eavesdroppers, therefore a positive secrecy capacity.

To improve the overall security, we optimized the key generation and secure transmissions separately. The secrecy capacity is maximized by minimizing key mismatch rate between two legitimate nodes. We introduced the notion of feed-forward and feed-back techniques into our quantizer design and proposed an MAP estimator with feed-back quantization scheme that achieves 1 dB improvement over the best known scheme in the high SNR regimes. Moreover, for the cases that the two legitimated nodes do not have any prior information about the underling statistical distribution, we proposed a universal quantization scheme with feed-forward and feed-back information. It has been verified by simulations that the proposed universal scheme can achieve the same performance as algorithms which require prior information.

We then investigated the scheme of applying rateless codes for secure transmissions. Rateless codes can potentially overcome the uncertainty from the key generation as well as channel variation. The performance of the proposed scheme is compared with a fixed-rate LDPC code. It is demonstrated in the simulation that the proposed scheme achieves a significant improvement over LDPC codes.

REFERENCES

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

[2] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography–part I: secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121—1132, 1993.

[3] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *Elsevier Digital Signal Processing Magazine*, vol. 6, pp. 207–212, 1996.

[4] C. Ye, A. Reznik, and Y. Shah, "Extracting secrecy from jointly Gaussian random variables," *IEEE Int'l Symp. Inf. Theory*, pp. 2593–2597, July 2006.

[5] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: secret sharing using reciprocity in UWB channels," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, Sept. 2007.

[6] A. Sayeed and A. Perrig, "Secure wireless communications: secret keys through multipath," *IEEE Int'l Conf. Acoustic, Speech & Signal Processing (ICASSP)*, pp. 3013–3016, Apr. 2008.

[7] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radiotelepathy: extracting a secret key from an unauthenticated wireless channel," *ACM SigMobile Int'l Conf. Mobile Computing and Networking (Mobicom)*, Sept. 2008.

[8] S. Jana, S. P. Nandha, M. Clark, S. K. Kasera, N. Patwari, and S. Krishnamurty, "On the effectiveness of secret key extraction using wireless signal strength in real environments," *ACM SigMobile Int'l Conf. Mobile Computing and Networking (Mobicom)*, Sept. 2009.

[9] 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 Propagation*, no. 53, pp. 3776–3784, Nov. 2005.

[10] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," *ACM Conf. Computer and Communications Security*, pp. 401–410, Nov. 2007.

[11] M. A. Tope and J. C. McEachen, "Unconditionally secure communications over fading channels," *Military Communications Conference (MILCOM)*, pp. 54–58, Oct. 2001.

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

[13] G. D. Durgin, "*Space-Time Wireless Channels*," Prentice Hall, Upper Saddle River, NJ, 2002.

[14] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," *Advances in Cryptology–EUROCRYPT*, pp. 410–423, 1994.

[15] C. H. Bennett, G. Brassard, C. Crepeau, and U. Maurer, "Generalized privacy amplification," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 1915–1923, Nov. 1995.

[16] J. W. Wallace, C. Chen, and M. A. Jensen, "Key generation exploiting MIMO channel evolution: algorithms and theoretical limits," *European Conf. Antennas and Propagation (EuCAP)*, Berlin, Germany, March 2009.

[17] M. Luby, "LT Codes," *43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[18] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52 , no. 6, pp. 2551–2567, 2006.

[19] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[20] R. Moskowitz, "Key management protocols-value, cost, and future proofing", *doc.: 15-10-0877-00-0hip-Turorial-KeyManagementProtocols*, Nov. 8, 2010. Last access Nov., 2010.

[21] A. D. Wyner " The wire-tap channel", *The Bell System Technical Journal*, vol. 54, pp. 1355-1387, Oct. 1975.

[22] I. Csiszár and J. Körner, "Broadcast channels with confidential messages", *IEEE Trans. Inform. Theory*, IT-24, no. 3, pp. 339-348, May 1978.

[23] Y. Liang, H. V. Poor, and S. Shamai (Shitz). "Secure communication over fading channels", *IEEE Transactions on Information Theory*, Special Issue on Information Theoretic Security, 54(6), 2470-2492, June 2008.

[24] Y. Liang and G. Kramer. "Rate regions for relay broadcast channels", *IEEE Transactions on Information Theory*, Special Issue on Models, Theory and Codes for Relaying and Cooperation in Communication Networks, 53(10), 3517-3535, Oct. 2007.

[25] T. Koike-Akino, A. F. Molisch, C. Duan, Z. Tao, and P. Orlik, "Capacity, MSE and secrecy analysis of linear block precoding for distributed antenna systems in multi-user frequency-selective fading channels", *IEEE Transactions on Communications.*, Mar. 2011.

[26] R. Liu, I. Maric, R. D. Yates, and P. Spasojevic, "The discrete memoryless multiple access channel with confidential messages", *IEEE ISIT2006*, Seattle, July 2006.

[27] Pedro C. Pinto, Joao Barros, and Moe Z. Win, "Secure Communication in Stochastic Wireless Networks", *http://arxiv.org/abs/1001.3697*, Jan. 2010. Last access date Oct., 2010.

[28] S. Goel and R. Negi, "Guaranteeing Secrecy using Artificial Noise", IEEE Transactions on Wireless Communications, vol. 7, no. 6, pp. 2180-2189, June 2008

[29] John G. Proakis, "Digital Communications", McGraw-Hill Science/Engineering/Math; 4th edition, 2000