entropy

MDPI

Article

A Hybrid Information Reconciliation Method for Physical Layer Key Generation

Guyue Li [1,*], Zheying Zhang [1], Yi Yu [2,3] and Aiqun Hu [1]

[1] School of Cyber Science and Engineer, Southeast University, Nanjing 210096, China
[2] Institut Supérieur d'Electronique de Paris, 75006 Paris, France
[3] Conservatoire National des Arts et Métiers, 75003 Paris, France
* Correspondence: guyuelee@seu.edu.cn

Received: 31 May 2019; Accepted: 12 July 2019; Published: 14 July 2019

Abstract: Physical layer key generation (PKG) has become a research focus as it solves the key distribution problem, which is difficult in traditional cryptographic mechanisms. Information reconciliation is a critical process in PKG to obtain symmetric keys. Various reconciliation schemes have been proposed, including the error detection protocol-based approach (EDPA) and error correction code-based approach (ECCA). Both EDPA and ECCA have advantages and drawbacks, regarding information leakage, interaction delay, and computation complexity. In this paper, we choose the BBBSS protocol from EDPA and BCH code from ECCA as a case study, analyzing their comprehensive efficiency performance versus pass number and bit disagreement ratio (BDR), respectively. Next, we integrate the strength of the two to design a new hybrid information reconciliation protocol (HIRP). The design of HIRP consists of three main phases, i.e., training, table lookup, and testing. To comprehensively evaluate the reconciliation schemes, we propose a novel efficiency metric to achieve a balance of corrected bits, information leakage, time delay, and computation time, which represents the effectively corrected bits per unit time. The simulation results show that our proposed method outperforms other reconciliation schemes to improve the comprehensive reconciliation efficiency. The average improvement in efficiency is 2.48 and 22.36 times over the BBBSS and BCH code, respectively, when the range of the BDR is from 0.5% to 11.5%. Compared to the BBBSS protocol and the BCH code, HIRP lies at a mid-level in terms of information leakage and computation time cost. Besides, with the lowest time delay cost, HIRP reaches the highest reconciliation efficiency.

Keywords: information reconciliation; comprehensive reconciliation efficiency; physical layer security; secret key generation

1. Introduction

Wireless communication is ubiquitous in our daily life, and it is expected to support extremely high data rates and radically new applications in the foreseeable future. Meanwhile, wireless transmission is vulnerable to eavesdropping attacks due to the broadcast nature of the wireless medium. Therefore, safeguarding data transmission is given the top priority in the development of next-generation wireless networks [1,2]. A paradigmatic problem of securing data transmission is the key distribution. Traditional public key cryptography techniques are widely used in existing communication networks. However, they require a public key infrastructure and are computationally intense, and thus encounter key distribution and management difficulties in the limited-resource mobile networks. Furthermore, with the advent of quantum computers capable of rapidly performing a complex and massive factorization, the traditional cryptography mechanism based on computation complexity is no longer reliable.

Recently, physical layer key generation (PKG) has been emerging as a supplement to the upper layer key distribution method [2]. The underlying idea of it lies in the use of channel reciprocity and the uncertainty of multipath characteristics to encrypt the transmitted information in order to solve the problem of symmetric secret key distribution [3,4]. Besides, the spatial variation prevents eavesdroppers from observing the same randomness as legitimate users, for a sufficiently large distance between them.

Although the uplink and downlink channels are reciprocal, measurements of radio channels are not the same, due to the differences originating from additive noise, transceiver hardware, and time delay in time division duplex (TDD) systems [5]. However, the objective of PKG is to generate a pair of strict identical symmetric keys. Even one bit of difference would lead to decryption failure due to the avalanche effect. To address this issue, information reconciliation is exploited to detect and correct errors in the preliminary key material between the two communicating parties [6].

Several information reconciliation approaches have been proposed in previous work. Generally, these approaches can be classified into two categories, i.e., error detection protocol-based approaches (EDPAs) and error correction code-based approaches (ECCAs). EDPAs, such as BBBSS [7], Cascade [8], and Winnow [9], use multiround interactive parity checks to eliminate mismatches. In EDPAs, Alice divides the preliminary key material into small blocks and sends the parity information of each block to Bob. Bob divides his key material in the same way, computes parity check bits, and checks for mismatches. For each mismatch, Bob performs a binary search on the block to find a correction vector, which may fix the errors. These steps are iterated a number of times to ensure a high probability of success. Bennett et al. proposed BBBSS on the permute-and-bisect method in the first implementation of quantum key distribution (QKD) [7]. To reduce information leakage, Brassard and Salvail proposed an improved scheme called Cascade in [8]. Cascade uses the information in the preceding passes to correct errors in the parity check of the current pass. The parity check in bisect is also replaced with an MD5 hash [10] and by a cyclic redundancy check (CRC) [11] to further reduce the information leakage of Cascade. A more efficient implementation of Cascade, using some inherent information already available in the protocol, exactly known bits, and already known parities, was proposed in [12]. However, these approaches lead to more interactions since parallelization becomes much more challenging. Winnow is another interactive reconciliation protocol introduced by Buttler et al. in [9]. Different from BBBSS and Cascade, Winnow resolves the errors in the block using a syndrome calculation with Hamming codes. Parity bits and syndromes can be calculated and exchanged in parallel, making the protocol less interactive than Cascade. However, Winnow can introduce new errors if the error count per block is more than two. A modified one-way error reconciliation protocol that employed a mixed Hamming code concatenation scheme was proposed to study the relationship between the error correction capability and the key generation efficiency in [13].

On the other hand, it has been shown that reconciliation can be deemed as a special case of channel coding [6]. Therefore, existing coded modulation techniques can be adapted for reconciliation. ECCA corrects the mismatches using error correction codes such as Hamming, BCH codes, and low density parity check (LDPC) codes [6,13–15]. Alice and Bob divide the preliminary key material into vectors. By utilizing the error correction code, Alice calculates and sends the parity check bits to Bob. Bob applies the corresponding decoder, whereby the required code word is composed of Bob's information vector and the received parity bits. If the number of bit disagreements is smaller than the error correcting capability, synchronized key material is guaranteed. Only one-round interaction is required in ECCA. In [13], a reconciliation protocol is proposed that is based on a mixed combination of Hamming syndrome concatenation. A reconciliation scheme, which uses syndromes of a BCH code for error correction and a one-bit feedback to report a successful decoding, is studied in [16]. In [17], the authors proposed using Turbo codes for reconciliation purposes. An information reconciliation protocol based on a rate-compatible construction of LDPC codes was proposed in [18]. However, the information leakage and computation complexity are generally increased in ECCA.

In existing work, reconciliation efficiency is the most commonly used evaluation metric, which is inversely proportional to leakage bit rate (LBR). However, rare work takes into account the interaction delay and computation complexity. But these factors may affect reconciliation efficiency greatly in some specific scenarios. For example, heavy interaction is unbearable in long-distance satellite communications. In Internet of Things (IoT) networks with limited resources, computation complexity has to be considered. Furthermore, both EDPA and ECCA have their pros and cons. There is still a gap regarding how to integrate their strengths to improve the reconciliation efficiency. To address these problems, this paper carries out a comprehensive and theoretical study on the information reconciliation schemes to establish highly efficient identical secret keys. Our contributions are as follows.

- A comprehensive reconciliation evaluation metric is proposed, taking consideration of LBR, interaction delay, and computation overhead. The metric represents the effective corrected bit number per unit time. The calculation expression of the metric is derived in this paper.
- The characteristics of BBBSS and BCH are analyzed from the perspective of the proposed new metric. Combining advantages of the two together, a new hybrid information reconciliation protocol (HIRP) is proposed. The detailed realization steps of HIRP are presented, including training, table lookup, and testing.
- The simulation results verify the theoretical analysis of both BBBSS and BCH. Monte Carlo simulations validate that the proposed HIRP outperforms the other two approaches to provide a more efficient information reconciliation in PKG.

The rest of the paper is organized as follows. Section 2 introduces the system model, the secret key generation process, the general information reconciliation model, and the problem studied in this paper. Section 4 provides a comprehensive reconciliation efficiency metric and presents the calculation expression for each factor. Section 5 proposes a new hybrid information reconciliation protocol (HIRP) and designs the realization algorithms. Section 6 presents the simulation results, and Section 7 concludes the paper.

2. System Model

2.1. General System Model

We consider a general Single Input single output single eavesdropper (SISOSE) model. All the users are equipped with a single antenna. Alice and Bob are two distinct legitimate users with a distance between each other of d meters. The communication system works at a frequency of f_c GHz with a bandwidth of B Hz. The data transmission rate is then B bits per second (bps). Alice and Bob intend to extract secret keys from their channel characteristics to protect the data transmission. Key generation requires a temporally dynamic channel, and the channel variation can be introduced by the movement of users and/or surrounding objects [19].

Eve is a passive eavesdropper located more than the coherence distance from both Alice and Bob. According to the definition of coherence distance, the coherence distance at a carrier frequency of 2.4 GHz is 6.25 cm. Therefore, we assume that Eve experiences a fading channel independent of that of Alice and Bob. Despite this, Eve knows the whole communication protocols, the pilots, and all the information transmitted over the public channels between Alice and Bob.

The notations used in this paper, and their definitions, are summarized in Table 1.

Table 1. Notations used throughout the paper.

Notation	Definition
B	System bandwidth
d	Distance between Alice and Bob
t_i	The time round index of secret key generation
T	Period of each secret key generation round
u	User Alice, Bob, and Eve
H_u	Channel characteristics estimation of user u
L_H	Length of H_u
Q_u	Quantized bit sequence of H of user u
L_Q	Length of Q_u
R_u	Corrected bit sequence of Q of user u
L_R	Length of R_u
P_u	Bit sequence of R after privacy amplification of user u
V_u	Verification hash value of user u
M	Interactive information leaked during reconciliation
L_M	Length of M
K	Interaction passes in reconciliation
$J(k)$	Number of back-and-forth interaction rounds for the k-th pass
$N_G(k)$	Number of groups for the k-th pass
$L_{RG}(k)$	Length of each group for the k-th pass
N_{aGE}	Average number of error bits in one group
$\mathbb{C}(n_c, k_c, t_c)$	Error correction code, message length k_c and error correcting bits t_c
ξ	Effective corrected bit number per unit time
η	Information leakage rate
T_{delay}	Time delay caused by interaction in reconciliation
T_{comp}	Computation time in reconciliation
N_{corr}	Corrected disagreement bit number
N_{eqAdd}	Number of "addition"
N_{EG}	Number of existing disagreement block detected by parity check

2.2. Secret Key Generation Process

Considering the $t_i \in \{1, 2, \cdots\}$-th round of secret key generation, Alice and Bob generate secret keys during a period of time T, as shown in Figure 1, which includes four main steps: channel sounding, quantization, information reconciliation, and privacy amplification. At first, Alice and Bob estimate their channel characteristics through channel sounding, i.e., sending pilots to each other. Eve may also estimate her channel to Alice or Bob. For simplicity, Eve's channel is referred to as that between Eve and Bob in this paper. Denote the channel characteristics estimated during T for the user $u \in \{A, B, E\}$ as H_u with length L_H, where A, B, and E represent Alice, Bob, and Eve, respectively. Secondly, the user u maps the input values from H_u into output values in a bit sequence set through quantization, e.g., channel quantization with guardband (CQG) used in [20]. The quantized bit sequence is represented as Q_u with length L_Q.

Until now, there has existed unavoidable bit disagreements between Q_A and Q_B, caused by time delay in TDD systems, hardware differences, and noise [21]. Although some preprocessing approaches, e.g, principal component analysis (PCA) [20], are applied, the bit disagreements are not fully eliminated. However, even a bit difference in a secret key will trigger an avalanche effect, leading to complete decryption failure. To deal with this problem, Alice and Bob correct the bit disagreements of Q_u through information reconciliation, and the corrected bit sequence is denoted by R_u with length $L_R = L_Q$. Totally, the L_M bits of parity information M are transmitted during the reconciliation process. The dashed line in Figure 1 shows that the communication may either be bidirectional or one way. Unfortunately, M is also leaked to Eve as she knows all the information transmitted through public channels. According to the leftover hash lemma, L_M bits arbitrarily chosen for R_u are discarded to guarantee key security during the privacy amplification step. For example, when $L_M = 40$ and

$L_R = 168$, a simple realization method is to map the 168-bit corrected sequence R_u to a 128-bit random sequence P_u through an MD5 hash function. Finally, the key consistency is verified by sending a simple hash value V_u of P_u from one to another. When the hash value is identical, the t_i-th round of secret key generation is successful. Otherwise, the t_i-th round of secret key generation fails, and the P_u is discarded.

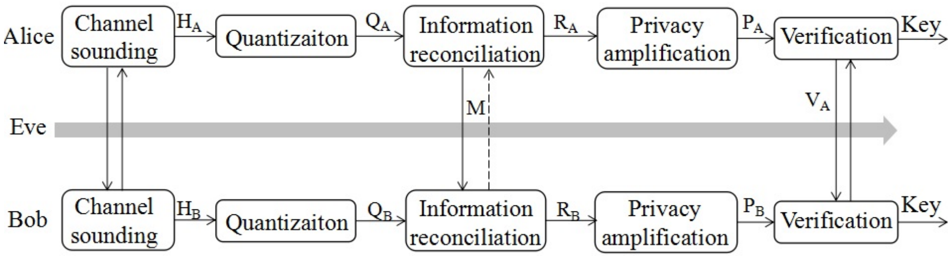


Figure 1. Physical-layer-based secret key generation process.

2.3. *A General Model for Information Reconciliation*

Various approaches, including EDPA and ECCA, are proposed for information reconciliation. In this section, we establish a general model for them. During the information reconciliation step, Alice communicates with Bob over public channels for K passes. All information transmitted over public channels is assumed to be error-free.

An EDPA, such as BBBSS, has $J(k), k = 1, 2, \cdots, K$ rounds of back-and-forth interactions for the k-th pass. In each round of interaction, Alice first sends the parity information to Bob, then Bob feeds back the information about error position to Alice. On the other hand, an ECCA, such as BCH codes, generally only has one pass and one round of communication, i.e., $K = 1$ and $J = 1$. It is because the error correct code has error propagation when the error number is beyond its error correcting capability. Therefore, it is inefficient for ECCA to gradually reduce bit error numbers through multiple passes or rounds. Besides, ECCA is a one-way communication in which Alice sends a syndrome to Bob but Bob does not provide feedback. Bob uses the syndrome to correct his channel observation through decoding algorithms, e.g., the Viterbi algorithm.

Figure 2 illustrates the information reconciliation process for both EDPA and ECCA. During the k-th pass of communication, Alice and Bob divide Q_u into $N_G(k)$ groups with group length $L_{RG}(k)$. Denote N_{aGE} as the estimated average number of error bits in one group. For EDPA, group length $L_{RG}(k)$ is designed to guarantee that each group has about one error, i.e., $N_{aGE} = 1$. During the first round of communication ($J(k) = 1$), Alice sends the parity of each group to Bob, and Bob feeds back the indexes of wrong groups. A group is defined as an error group if the parity information of Alice and Bob is different. Then, for each wrong group, $J(k) - 1$ rounds of bisect error-correcting are applied to find the position of error bit. As for ECCA, group length L_{RG} depends mainly on the affordable decoding complexity of Bob. In the affordable range, the larger the L_{RG}, the more accurate the N_{aGE}. Therefore, L_{RG} is usually set as the largest affordable length. Each group may have more than one error in this case. According to the signal-to-noise ratio (SNR), it is estimated that the ratio of N_{aGE} to L_{RG} matches the coarse bit disagreement ratio (BDR) estimation of Q_u. Then, ECCA chooses the error correction code $\mathbb{C}(n_c, k_c, t_c)$, where n_c, k_c and t_c are the code length, message length, and error correcting number, respectively. Code $\mathbb{C}(n_c, k_c, t_c)$ satisfies that the message bit length $k_c = L_{RG}$ and the correction error number $t_c \geq N_{aGE}$. Next, Alice divides Q_A into groups and sends all groups of syndromes to Bob. According to the syndromes, Bob corrects the inconsistent bits in Q_B using decoding algorithms.

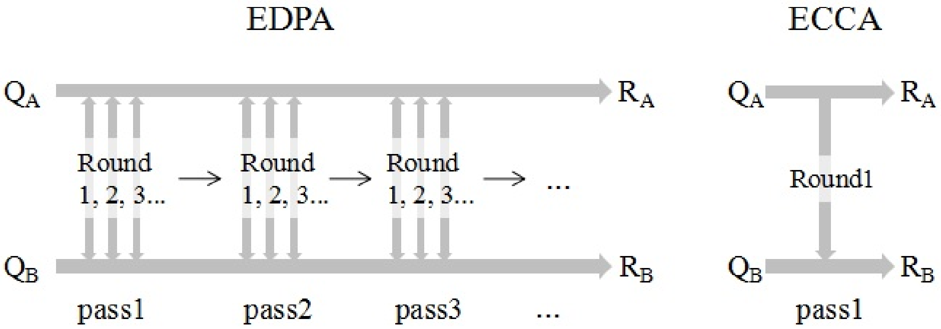


Figure 2. Illustration of information reconciliation process for the error detection protocol-based approach (EDPA) and error correction code-based approach (ECCA).

3. Problem Statement

Information reconciliation approaches are mainly categorized as EDPA or ECCA, and each has its advantages and drawbacks. The downside to EDPA is that it needs multiple passes and multiple rounds of back-and-forth interactions, as shown in Figure 2. When Alice is far away from Bob, it causes a very large interaction delay and communication overhead. Furthermore, the efficiency of EDPA decreases with the increase in pass number. The proof is provided in Section 3. On the plus side, EDPA just uses bisect error-correcting, which consumes less computation and leaves less leakage of information.

Conversely, ECCA only has one pass, one round, and one-way communication. Obviously, the interaction delay and communication overhead are significantly reduced. The negative side of ECCA is that it has expensive computation overhead and large information leakage, especially for low SNR scenarios. If L_{RG} is small, the estimate of N_{aGE} is inaccurate, which may lead to error propagation. Instead, if L_{RG} is large, the decoding complexity is high. Even worse, information leakage increases rapidly with the rise of N_{aGE} for ECCA. Table 2 summarizes the features of EDPA and ECCA.

Table 2. Features of EDPA and ECCA.

Features	Interaction	Complexity	Leakage
EDPA	High	Low	low
ECCA	Low	High	High

Since both EDPA and ECCA have their pros and cons, this raises a natural question: "How to comprehensively evaluate the performance of an information reconciliation approach?" Existing work only considers one or two indicators of performance, e.g., information leakage, leaking the evaluations of interaction time, complexity, etc. The subsequent problem is: "Is it possible to integrate the strengths of both EDPA and ECCA to design a new reconciliation approach that makes a trade-off of all these performance indicators?" To address this problem, we first propose a comprehensive metric to evaluate the efficiency of reconciliation approaches. Next, we discuss the performance of EDPA and ECCA, respectively. In this paper, we choose BBBSS of EDPA and BCH code of ECCA as a case study. Under the guidance of the new metric, we then design a new approach named HIRP to achieve good efficiency.

4. A Comprehensive Information Reconciliation Evaluation Metric

In this section, we propose a comprehensive reconciliation efficiency metric, taking consideration of corrected bits, information leakage, interaction delay, and computation time.

4.1. Information Leakage

Information reconciliation poses a security threat as eavesdroppers can infer keys from the interacted information. Therefore, the information leakage should be considered when evaluating an information reconciliation scheme.

Definition 1. *Denote η as the information leakage ratio, which is defined by*

$$\eta = \frac{I(R_A, M)}{L_R}, \tag{1}$$

where R_A is the reconciled key with length L_R, M is information disclosed during interaction, and $I(R_A, M)$ is the mutual information between them, which represents the information that eavesdroppers can obtain about the key. To guarantee the security of the final key, at least η proportion of the reconciled keys should be wiped off in the privacy amplification step.

Remark 1. *Denote L_M as the length of M and ε as the BDR between Q_A and Q_B, then the lower bound and upper bound of η are derived as*

$$h(\varepsilon)\frac{L_Q}{L_R} \le \eta \le \frac{L_M}{L_R}, \tag{2}$$

where $h(\varepsilon)$ is the entropy of ε with

$$h(\varepsilon) = -\varepsilon \cdot \log \varepsilon - (1-\varepsilon) \cdot \log(1-\varepsilon). \tag{3}$$

The lower bound of η represents the minimum amount of interaction information per bit for Q_A and Q_B to obtain identical keys. Since L_M is the length of M, then the maximum disclosed bits is L_M. When M has a linear relationship with Q_A, the disclosed bits is L_M. Otherwise, it is less than L_M due to the increased ambiguity caused by nonlinearity. Therefore, the upper bound of η is L_M/L_R.

In this paper, we calculate the information leakage ratio through its upper bound with $\eta = \frac{L_M}{L_R}$ for security purposes.

4.2. Interaction Delay

The interaction delay represents the time spent on exchanging information M. It can become significant in EDPA, which has multiround interactions. Denote T_{delay} as the interaction delay, which includes two parts, i.e., the data transmission time and the propagation time. Then, T_{delay} is calculated as

$$T_{delay} = T_{data} + T_{prop} = \frac{L_M}{B} + 2\sum_{k=1}^{K} J(k)(\frac{d}{c} + T_0), \tag{4}$$

where B is the system bandwidth, $\sum\limits_{k=1}^{K} J(k)$ is the number of back-and-forth interactions, d is the transmission distance, c is the velocity of light, and T_0 is the communication overhead in every back-and-forth interaction.

As derived in (1), $L_M \ge h(\varepsilon)L_Q$, and thus

$$T_{delay} \ge \frac{h(\varepsilon)L_Q}{B} + 2\sum_{k=1}^{K} J(k)(\frac{d}{c} + T_0). \tag{5}$$

The latter term rises with the increase of $\sum\limits_{k=1}^{K} J(k)$. Besides, in long-distance communications, such as satellite communications, the latter term becomes the dominant factor for long interaction delays. Therefore, the number of information interactions should be lowered to reduce the delay and communication overhead.

4.3. Computation Time

In some resource-constrained systems, the performance of error-correcting schemes may be constrained since decoding algorithms require multiple round iterations. Therefore, computation complexity, which is characterized by the computation time T_c, should be taken into account. Denote T_c as

$$T_{comp} = t_c \cdot N_{eqAdd}, \tag{6}$$

where t_c is the time cost of an "equivalent addition" and N_{eqAdd} represents the number of equivalent additions. The required mathematical and logical operations can be viewed as multiples of "equivalent addition" due to current digital signal processor (DSP) specifications in [22]. In Table 3, computation operations are normalized to 5. T_{comp} is determined by the BDR of initial keys, group size, and decoding complexity. The higher the BDR is, the longer the computation time is. Generally, ECCA has a much heavier computation cost than EDPA.

Table 3. Equivalent addition conversion table.

Addition/Subtraction:	1	Division by 2:	1	Max/Min(2 arguments):	2
(± 1)·Multiplication:	1	Table lookups:	6	Compare:	1

4.4. Effective Reconciliation Rate ξ

To achieve a balance in the above factors, we propose a novel comprehensive metric ξ, which is called the effective reconciliation rate, to evaluate the performance. The definition of ξ is given by

$$\xi = \frac{(1-\eta)N_{corr}}{T_{delay} + T_{comp}}, \tag{7}$$

where N_{corr} denotes the number of corrected inconsistent bits. Actually, ξ represents the effective corrected bit number per unit time. Therefore, it reflects the efficiency of an information reconciliation approach. There is a negative correlation between ξ and information leakage η, and interaction delay T_{delay} and computation time T_{comp}. Reducing the value of η, T_{delay}, and T_{comp} contributes to the improvement of ξ. The higher the ξ is, the more efficient the information reconciliation approach is.

5. A Hybrid Information Reconciliation Protocol

In this section, we first review the characteristics of BBBSS and BCH from the perspective of the new metric ξ. Combining the advantages of both, we propose a new approach named HIRP, which aims to improve the comprehensive reconciliation efficiency.

5.1. BBBSS

The BBBSS protocol uses permutation-and-bisect block to remove the discrepancies [23]. Define one pass of bisect and permutation block correction as one pass of BP. Figure 3 illustrates the flow chart of BBBSS, in which solid blocks contain information interactions. Permutation distributes disagreements randomly and then groups key strings into blocks using estimated BDR. The block length is recommended as $L_{RG}(k) = 0.73/\varepsilon(k)$, where $\varepsilon(k)$ is the BDR for the k-th pass. Then Alice and Bob interact the parity check of each block to find out error blocks and apply bisect error correcting to correct disagreements. Since this method couldn't detect the block that has an even number of disagreements, multiple passes of BPs are required. The pass iteration terminates when the parity check of all the blocks are identical.

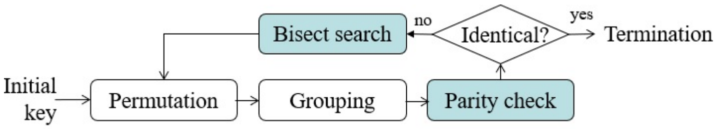


Figure 3. The flow chart of BBBSS.

We further define the efficiency metric in the k-th pass as

$$\xi(k) = \frac{(1-\eta^{(k)})N_{corr}^{(k)}}{T_{delay}^{(k)} + T_{comp}^{(k)}}. \tag{8}$$

The information leakage satisfies that

$$\eta^{(k)} = \frac{L_M(k)}{L_R} = \frac{N_G(k) + N_{EG}^{(k)}[J(k)-1]}{L_R}, \tag{9}$$

where $N_{EG}^{(k)} = N_{corr}^{(k)}$ indicates the number of error groups for the k-th pass and $J(k) = \lceil \log_2 L_{RG}(k) \rceil + 1$ is the number of back-and-forth interactions for the k-th pass. Except for the first round of finding the error groups, it needs additional $\lceil \log_2 L_{RG}(k) \rceil$ rounds to find the error position.

The time delay satisfies that

$$T_{delay}^{(k)} = \frac{L_M(k)}{B} + 2\sum_{k=1}^{K} J(k)(\frac{d}{c} + T_0). \tag{10}$$

The computation time T_{comp} has a linear growth with $L_M(k)$.

In general, with the increase of pass number k, the group number $N_G(k)$ and corrected bits $N_{corr}^{(k)}$ decline. However, the group length $L_{RG}(k)$ increases, and thus the interaction number $J(k)$ increases. As stated in Section 4.2, the latter term in the time delay plays a dominant role. When k is small, one round of interaction is more efficient as it processes parity information for multiple groups in parallel. However, when k is large, even one error group may need a round of interaction, which causes low efficiency. With $N_{corr}^{(k)}$ and $T_{delay}^{(k)}$ playing dominant roles, ξ decreases with the increase in pass number, which is also verified in the simulations of Section 6. To sum up, BBBSS has high efficiency at the first several passes and then becomes less efficient in subsequent passes.

5.2. BCH

BCH code with $\mathbb{C}(n_c, k_c, t_c)$ has only one pass of interaction. Since each group has the same code, the ξ in one group is equal to the whole ξ. Then the leakage rate satisfies that

$$\eta = \frac{n_c - k_c}{k_c}. \tag{11}$$

When $\mathbb{C}(n_c, k_c, t_c)$ is capable of correcting all of the errors, then $N_{corr} = t_c$. The time delay is

$$T_{delay} = \frac{n_c - k_c}{B} + \frac{d}{c} + T_0. \tag{12}$$

At last, T_{comp} rises with the increase of t_c.

The metric ξ is

$$\xi = \frac{(1 - \frac{n_c - k_c}{k_c})t_c}{(\frac{n_c - k_c}{B} + \frac{d}{c} + T_0) + T_{comp}(t_c)}. \tag{13}$$

To correct t_c errors, it has to be satisfied that $n_c - k_c \geq 2t_c + 1$. Assume that $n \approx k_c + (2t_c + 1)$, then Equation (13) is approximated as

$$\xi \approx \frac{(1 - \frac{2t_c+1}{k_c})t_c}{(\frac{2t_c+1}{B} + \frac{d}{c} + T_0) + T_{comp}(t_c)}. \tag{14}$$

In one group, $\mathbb{C}(n_c, k_c, t_c)$ satisfies that the message bit length $k_c = L_{RG}$ and the correction error number $t_c \geq N_{aGE}$. Thus, $\varepsilon \approx t_c/k_c$, and k_c is a constant that mainly depends on the affordable decoding complexity of Bob. Thus, Equation (15) is further written as

$$\xi \approx \frac{a_1\varepsilon^2 + a_2\varepsilon}{a_3\varepsilon + T_{comp}(\varepsilon) + a_4}, \tag{15}$$

where $a_1 = -2k_c$, $a_2 = k_c - 1$, $a_3 = \frac{2k_c}{B}$, $a_4 = \frac{1}{B} + \frac{d}{c} + T_0$. $T_{comp}(\varepsilon)$ decreases monotonously along with increasing ε. Because $a_1 < 0$ and $a_3 > 0$, ξ decreases along with increasing ε generally. In summary, BCH is more efficient in low BDR regions.

5.3. The Algorithm of the Proposed HIRP

In the previous analysis, both BBBSS and BCH are efficient in some specific conditions. On the one hand, BBBSS is effective for the first few passes of BP, and its BDR is reduced down (about threefold) after each pass. On the other hand, BCH shows better efficiency at low BDR regions. Inspired by these, we propose a hybrid approach named HIRP to further improve the reconciliation efficiency, combining the virtues of both BBBSS and BCH.

Figure 4 illustrates the flow chart of HIRP. The core idea is that when the BDR is high, several passes are firstly exploited to reduce it to a low value, and then the few residual errors are further corrected by BCH, which is efficient in low BDR regions. Algorithm 1 gives the details of the realization steps of HIRP, which contains three main phases, i.e., training, table lookup, and testing.

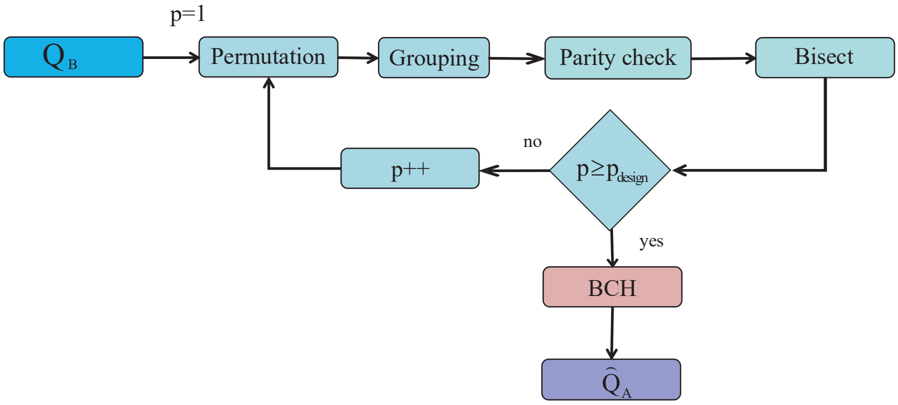


Figure 4. The flow chart of the hybrid information reconciliation protocol (HIRP).

Algorithm 1 Algorithm of HIRP

```
Input: training data: $Q_A^{Train}$ and testing data: $Q_A^{Test}, Q_B^{Test}$    Output: Estimated testing data $\hat{Q}_A$
Training phase:
  1. Add noise to $Q_A^{Train}$ to generate $Q_B^{Train}$ with different BDR $\varepsilon$.
  2. Traverse all possible BPs for different $\varepsilon$ in range and calculate their efficiency respectively.
  3. Find the optimal pass number to maximize the efficiency $\xi$ and draw Table 4.
Table lookup phase:
  1. Mark the locations of $p_{optimal}$ in Table 5.
  2. Find the threshold BDR $\varepsilon_{th}$.
  3. Calculate the desgined pass number $p_{designed}$ and draw Table 6.
Testing phase:
  1. Estimate the $\varepsilon$ of $Q_A^{Test}$ and $Q_B^{Test}$.
  2. Select $p_{designed}$ from Table 5 with the estimated BDR.
  3. Use the $H_p$ algorithm for reconciliation, which applies $p_{designed}$ passes of BPs firstly and then
     eliminates remaining disagreements by BCH codes.
```

Table 4. Optimal p for different bit disagreement ratios (BDRs) .

$\varepsilon(\%)$	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5
$p_{optimal}$	1	2	2	2	3	3	3	3	3	3	3	3

Table 5. Output BDR after p passes of bisect and permutation (BP).

p $\varepsilon(\%)$	**1**	**2**	**3**	**4**	**5**	**6**
0.5	0.184	0.060	0.016	0.003	0.000	0.000
1.5	0.564	0.186	0.049	0.009	0.001	0.000
2.5	0.944	0.312	0.082	0.015	0.002	0.000
3.5	1.304	**0.425**	0.108	0.018	0.002	0.000
4.5	1.699	0.559	0.142	0.024	0.002	0.000
5.5	2.065	0.672	0.170	0.028	0.002	0.000
6.5	2.438	0.789	0.196	0.030	0.002	0.000
7.5	2.721	0.862	0.205	0.030	0.002	0.000
8.5	3.093	0.988	0.240	0.035	0.002	0.000
9.5	3.405	1.068	0.249	0.034	0.002	0.000
10.5	3.635	1.084	0.236	0.029	0.001	0.000
11.5	4.187	1.293	0.303	0.042	0.002	0.000

Define H_p as the HIRP approach with p passes of BP. When $p = 0$, HIRP turns into BCH, and when p gets large enough, HIRP is equal to BBBSS. The parameter selection of p is critical to our proposed approach. In the training phase, we first collect the optimal p values that achieve the maximal ξ for a group of BDRs. Figure 5 shows a realization framework to find $p_{optimal}$. By adding artificial noise to training data Q_A^{Train}, we get the desired Q_B^{Train} with BDR ε ranging from 0.5% to 11.5%. The collected results of $p_{optimal}$ versus ε are shown in Table 4.

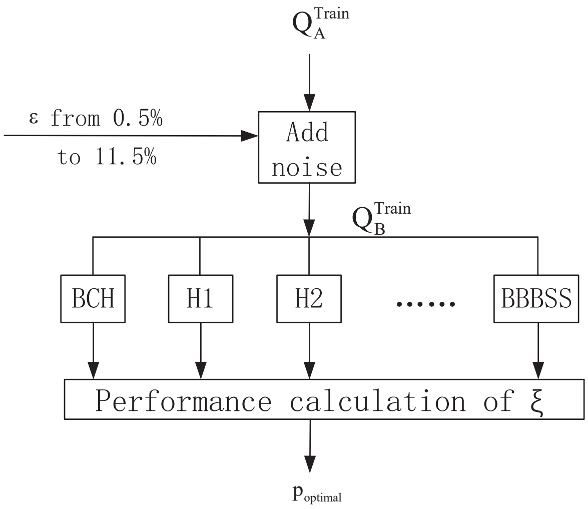


Figure 5. Block diagram of training phase.

Although when $p = p_{optimal}$, H_p can achieve the highest ξ, the traversing method is complicated, and the cost is huge in practical applications. Besides, it is challenging to go through all the possible H_ps for every possible ε. To deal with the problem, we design a new table of p_{design} versus ε with the combination of both Tables 4 and 5. The element $(\varepsilon_{in}, p, \varepsilon_{out})$ in Table 5 represents the input BDR, pass number, and the corresponding output BDR. From Table 5, the BDR is reduced to roughly one third after every pass. After p passes, the BDR of output signals ε_{out} satisfies that

$$\varepsilon_{out} \approx \varepsilon(\frac{1}{3})^p. \tag{16}$$

To simplify the process, $p_{designed}$ is calculated as the minimum value of p, that satisfies

$$\varepsilon_{out} \approx \varepsilon(\frac{1}{3})^p \leq \varepsilon_{th}, \tag{17}$$

where the threshold ε_{th} is set as the largest value of ε_{out} among the marked elements. From Table 5, the threshold of our simulation is $\varepsilon_{th} = 0.425$. According to the above rules, Table 6 gives the value of $p_{designed}$ versus different ε.

Table 6. Designed p for different BDR.

ε(%)	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5
$p_{designed}$	1	2	2	2	3	3	3	3	3	3	3	4

In the testing phase, Bob first estimates the ε. The coarse BDR estimation can be calculated according to the channel signal-to-noise ratio. After that, Q_A and Q_B are grouped with $len = 0.73/\varepsilon_{coarse}$, which satisfies $N_{aGE} = 1$. Then A and B interact the parity check of each block for a fine BDR estimation. Next, Bob selects the corresponding pass number $p_{designed}$ from Table 6. Finally, Bob conducts the algorithm of H_p for information reconciliation and recovers the bit sequence of $\hat{Q}_A$. The block diagram of the testing phase is illustrated in Figure 6. The testing phase does not need sophisticated communication or a heavy computation cost. The additional operation is the table lookup, which is easy to realize in practice.

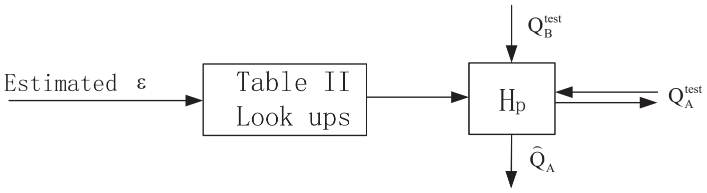


Figure 6. Block diagram of testing phase.

6. Simulations

In this section, we give some simulation results of the BBBSS, BCH, and our proposed HIRP scheme with $p_{optimal}$ and p_{design} for comparison. The communication distance is set as $d = 5$ KM, the communication bandwidth is $B = 4$ MHz. The communication overhead in one interactive is set as $T_0 = 50$ ms for the consideration of packet loss.

First, we simulate the efficiency metrics of BBBSS in every individual pass. The results are given in Figure 7. Both the corrected bit number and information leakage ratio reduce with the increase in pass number. The interaction time delay rises at first and then goes down after the 4-th pass. This is caused by the fact that the group length L_{RG} increases, while the number of error groups is not reduced significantly. Comprehensively, as shown in Figure 8, the metric ξ decreases with the increase in pass number, which means that BBBSS has a high efficiency at the first several passes. The simulation results coincide with the theoretical analysis in Section 5.1.

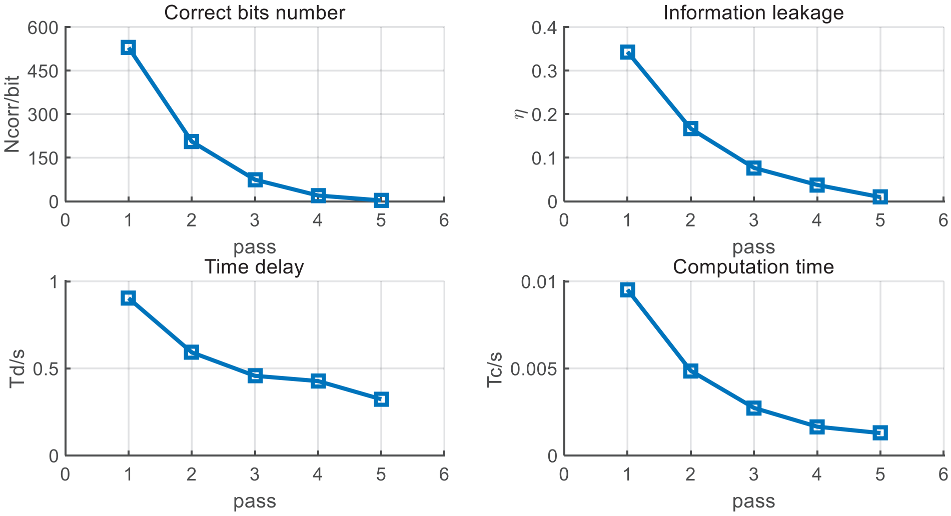


Figure 7. Individual performance of BBBSS per pass with $\varepsilon = 11\%$, $d = 5$ KM, $B = 4$ MHz, and $T_0 = 50$ ms.

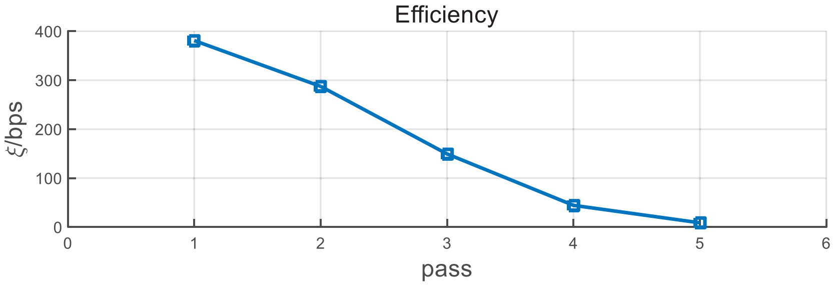


Figure 8. Efficiency performance of BBBSS per pass with $\varepsilon = 11\%$, $d = 5$ KM, $B = 4$ MHz, and $T_0 = 50$ ms.

We also simulate the individual performance of BCH for different BDRs in Figure 9. With the increase in the BDR, the information leakage ratio, the time delay, and the computation time show an upward trend. Therefore, the performance of ξ presents a general falling tendency, as shown in

Figure 10. When the BDR is lower than 1.5%, the ξ has a slight increase. This is because the correct bit number has a significant rise with the increase in BDR.

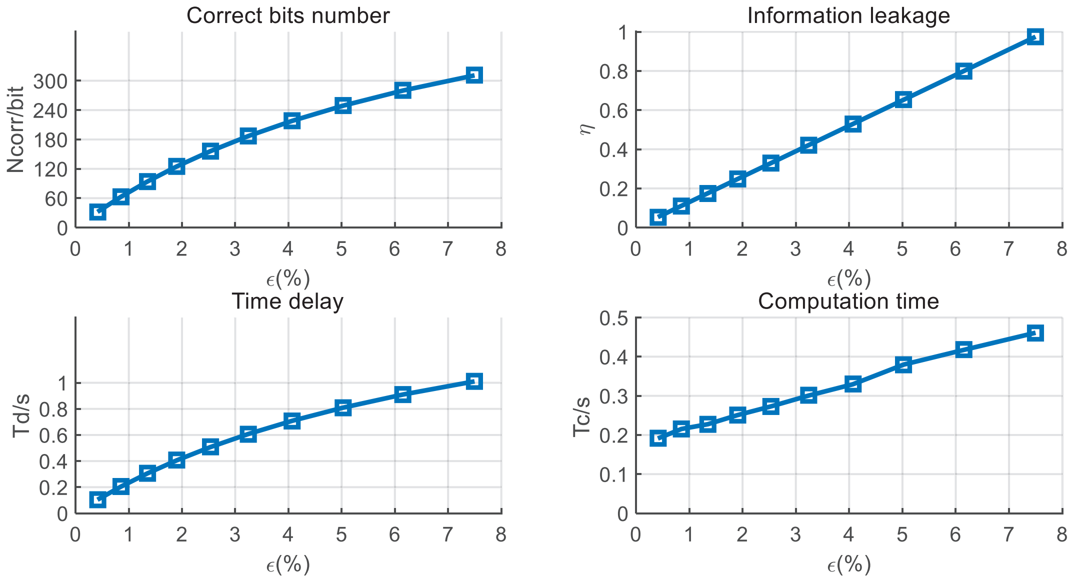


Figure 9. Individual performance of BCH versus BDR with $d = 5$ KM, $B = 4$ MHz, and $T_0 = 50$ ms.

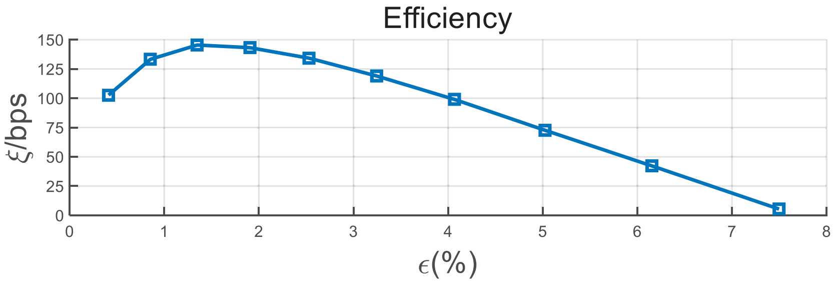
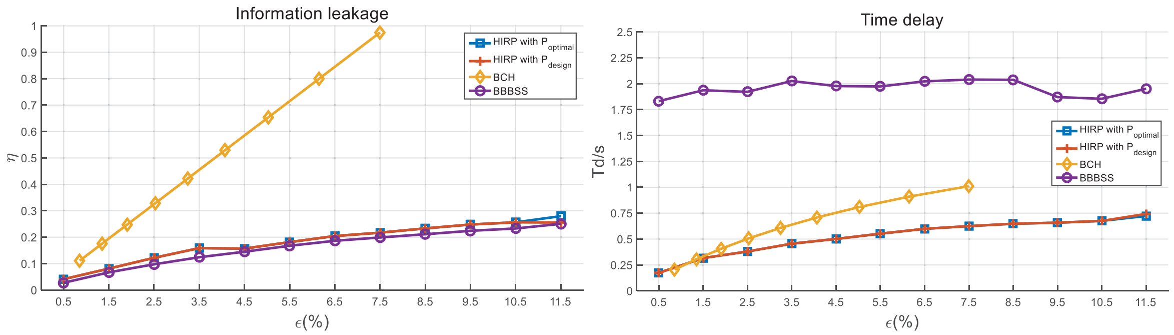


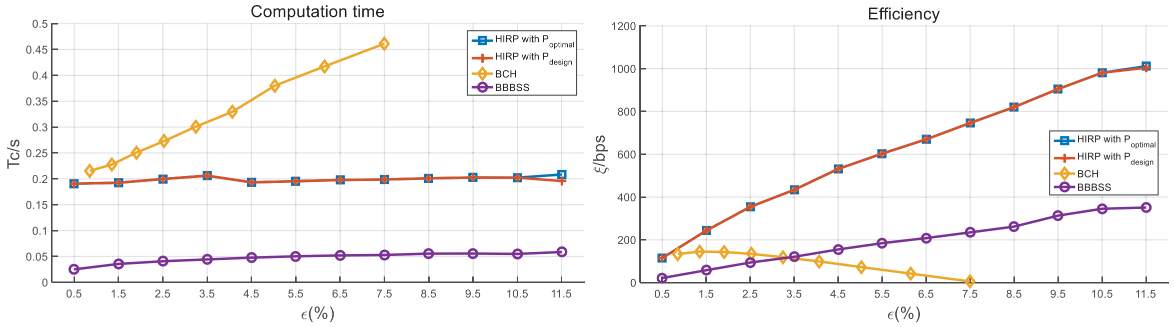
Figure 10. Efficiency performance of BCH versus BDR with $d = 5$ KM, $B = 4$ MHz, and $T_0 = 50$ ms.

Next, we compare the performance in terms of the information leakage, the time delay, the computation time, and the comprehensive efficiency for various information reconciliation approaches including BBBSS, BCH, and HIRPs with optimal and designed pass numbers. Figure 11a shows the information leakage ratio versus BDRs. BCH has the highest information leakage ratio, which rises significantly with the increase in BDR. When the BDR is 7.5%, the BCH code is chosen as $\mathbb{C}(8191, 4148, 311)$, and the leakage ratio reaches 1. Therefore, we do not represent the BCH performance results for BDRs larger than 7.5%. The leakage ratio of the HIRPs is almost identical to that of BBBSS, and their growth is slow with BDR. Figure 11b represents the interaction time delay as a function of the BDRs. BBBSS has a longer interaction time compared with others. HIRPs have the shortest time delay and the slowest growth for BDRs above 1.5%. Figure 12a describes the computation time with respect to BDRs. The computation complexity rises significantly with the ramp-up of BDR. BCH has the longest computation time, which becomes significant in high BDR regions. BBBSS has the shortest computation time, and HIRPs have the middle one. In addition, the computation times of BBBSS and HIRPs rise slowly with the increase in BDR.



(**a**) (**b**)

Figure 11. The comparison of (**a**) information leakage and (**b**) time delay. $d = 5$ KM, $B = 4$ MHz, and $T_0 = 50$ ms.



(**a**) (**b**)

Figure 12. The comparison of (**a**) computation time and (**b**) efficiency. $d = 5$ KM, $B = 4$ MHz, and $T_0 = 50$ ms.

Figure 12b compares the efficiency ξ of different information reconciliation approaches. In low BDR regions, BCH has a better performance than BBBSS, while in high BDR regions, the opposite is true. It is observed that the ξ of HIRP outperforms that of both BBBSS and BCH along all BDR regions. It should be noted that when we only consider information leakage and computation time, HIRP seems to have no advantage compared to BBBSS. However, the time delay in Figure 11b shows that HIRP has a much lower time delay than BBBSS. The multipass interaction in the BBBSS protocol increases its time delay seriously. Therefore, the final comprehensive efficiency of HIRP is higher than that of BBBSS. In addition, HIRP with designed p has almost the same performance as HIRP with optimal p. The results verify the effectiveness of our proposed approach.

Table 7 shows the numerical improvement results of HIRP against BBBSS and BCH. According to Equation (7), the effective reconciliation rate ξ is inversely proportional to information leakage η, time delay T_{delay}, and computation time T_{comp}. Compared to BBBSS, the comprehensive efficiency ξ is improved 2.48 times, mainly due to the fact that HIRP declines T_{delay} by 73% on average. Compared to BCH codes, HIRP declines η, T_{delay}, and T_{comp}, which results in the improvement of HIRP efficiency by an average of 22.36.

Table 7. Improvement of HIRP reconciliation factors.

	Compared to BBBSS				**Compared to BCH Codes**			
BDR	ξ	η	T_{delay}	T_{comp}	ξ	η	T_{delay}	T_{comp}
0.50%	4.35	0.50	−0.90	6.61	0.07	−0.37	0.40	−0.16
6.50%	2.22	0.10	−0.70	2.80	19.41	−0.76	−0.36	−0.53
Average	2.48	0.17	−0.73	3.38	22.36	−0.66	−0.20	−0.37

7. Conclusions

This paper examined the efficiency of information reconciliation approaches. We introduced a comprehensive reconciliation efficiency metric that considers the corrected bits, the interaction delay, and the computation time synthetically. Furthermore, we analyzed the characteristics of both BBBSS and BCH from the perspective of the metric. The efficiency of BBBSS decreases along with pass number, and BCH has low efficiency in high BDR regions. Inspired by this, we proposed a HIRP method that exploits certain passes of BP and then corrects the residual errors by BCH. The design of HIRP contains training, table lookup, and testing phases. The simulation results verified the effectiveness of our proposed HIRP approach. HIRP improves the comprehensive reconciliation efficiency 2.48 and 22.36 times compared with BBBSS and BCH, respectively. It makes a trade-off between individual performance indicators by achieving a median value of information leakage, interaction delay, and computation time. In the future, we plan to study the parameter design of HIRP from the theoretical point of view in some specific scenarios. In addition, we chose the BBBSS protocol of EDPA and the BCH code of ECCA as a case study in this paper. We plan to expand HIRP to a more general hybrid method considering more protocols and codes in EDPA and ECCA in our next step.

Author Contributions: Conceptualization, G.L.; methodology, G.L.; validation, Z.Z.; formal analysis, G.L. and Z.Z.; investigation, Z.Z.; writing—original draft preparation, G.L. and Z.Z.; writing—review and editing, G.L., A.H. and Y.Y.

Funding: This research was supported by the China Scholarship Council, National Natural Science Foundation of China (61801115, 61571110), the Purple Mountain Network Communications and Security Laboratory, and the Joint Laboratory for Mobile Key Management, State Key Laboratory of Cryptology, School of Cyber Science and Engineering, Southeast University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kai, Z. Physical layer key generation in wireless networks: Challenges and opportunities. *IEEE Commun. Mag.* **2015**, *53*, 33–39.
2. Li, G.; Sun, C.; Zhang, J.; Jorswieck, E.; Xiao, B.; Hu, A. Physical Layer Key Generation in 5G and Beyond Wireless Communications: Challenges and Opportunities. *Entropy* **2019**, *21*, 497. [CrossRef]
3. Mathur, S.; Trappe, W.; Mandayam, N.; Ye, C.; Reznik, A. Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel. In Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, San Francisco, CA, USA, 14–19 September 2008; pp. 128–139.
4. Jana, S.; Premnath, S.N.; Clark, M.; Kasera, S.K.; Patwari, N.; Krishnamurthy, S.V. On the effectiveness of secret key extraction from wireless signal strength in real environments. In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, Beijing, China, 20–25 September 2009; ACM: New York, NY, USA, 2009; pp. 321–332.
5. Zhang, J.; Woods, R.; Duong, T.Q.; Marshall, A.; Ding, Y. Experimental Study on Channel Reciprocity in Wireless Key Generation. In Proceedings of the 2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Edinburgh, UK, 3–6 July 2016; pp. 1–5.
6. Huth, C.; Guillaume, R.; Strohm, T.; Duplys, P.; Samuel, I.A.; Güneysu, T. Information reconciliation schemes in physical-layer security: A survey. *Comput. Netw.* **2016**, *109*, 84–104. [CrossRef]
7. Bennett, C.H.; Bessette, F.; Brassard, G.; Salvail, L.; Smolin, J. Experimental quantum cryptography. *J. Cryptol.* **1992**, *5*, 3–28. [CrossRef]
8. Brassard, G.; Salvail, L. Secret-Key Reconciliation by Public Discussion. In *Advances of Cryptology-Eurocrypt '93, Proceedings of the 1993 Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lofthus, Norway, 23–27 May 1993;* Springer: Berlin/Heidelberg, Germany, 1993; Volume 765, pp. 410–423.

9. Buttler, W.T.; Lamoreaux, S.K.; Torgerson, J.R.; Nickel, G.H.; Donahue, C.H.; Peterson, C.G. Fast, efficient error reconciliation for quantum cryptography. *Phys. Rev. A* **2003**, *67*, 125–128. [CrossRef]
10. Gong, C.Q.; Zhou, H.Y.; Feng, J.L. An Improvement of Protocol Binary in Reconciliation of Quantum Key Distribution. In Proceedings of the 2009 International Conference on Management and Service Science, Wuhan, China, 20–22 September 2009; pp. 1–4.
11. Gong, C.Q.; Zhou, H.Y.; Feng, J.L. Research on Reconciliation Algorithm in Quantum Key Distribution. In Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems, Shenyang, China, 12–14 August 2009; pp. 1–3.
12. Toyran, M. More efficient implementations of CASCADE information reconciliation protocol. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 16–19 May 2016; pp. 161–164. [CrossRef]
13. Zhao, F.; Li, J. Performance of an Improved One-Way Error Reconciliation Protocol Based on Key Redistribution. *China Commun.* **2014**, *11*, 63–70.
14. Tian, T.; Jones, C.R. Construction of Rate-Compatible LDPC Codes Utilizing Information Shortening and Parity Puncturing. *Eurasip J. Wirel. Commun. Netw.* **2005**, *2005*, 789–795. [CrossRef]
15. Niuniu, S.; Shuilian, Z.; Gang, X.; Wenbing, C. The information reconciliation protocol basing on error codes. In Proceedings of the 2013 IEEE 4th International Conference on Software Engineering and Service Science, Beijing, China, 23–25 May 2013; pp. 693–696. [CrossRef]
16. Treeviriyanupab, P.; Sangwongngam, P.; Sripimanwat, K.; Sangaroon, O. BCH-based Slepian-Wolf coding with feedback syndrome decoding for quantum key reconciliation. In Proceedings of the 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Phetchaburi, Thailand, 16–18 May 2012; pp. 1–4.
17. Nguyen, K.C.; Assche, G.V.; Cerf, N.J. Side-Information Coding with Turbo Codes and its Application to Quantum Key Distribution. *arXiv* **2004**, arXiv:cs/0406001.
18. Elkouss, D.; Martinez, J.; Lancho, D.; Martin, V. Rate compatible protocol for information reconciliation: An application to QKD. In Proceedings of the 2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo), Cairo, Egypt, 6–8 January 2010; pp. 1–5. [CrossRef]
19. Li, G.; Hu, A.; Zhang, J.; Xiao, B. Security Analysis of a Novel Artificial Randomness Approach for Fast Key Generation. In Proceedings of the GLOBECOM 2017–2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
20. Li, G.; Hu, A.; Zhang, J.; Peng, L.; Sun, C.; Cao, D. High-Agreement Uncorrelated Secret Key Generation Based on Principal Component Analysis Preprocessing. *IEEE Trans. Commun.* **2018**, *66*, 3022–3034. [CrossRef]
21. Gopinath, S.; Guillaume, R.; Duplys, P.; Czylwik, A. Reciprocity enhancement and decorrelation schemes for PHY-based key generation. In Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps), Austin, TX, USA, 8–12 December 2014; pp. 1367–1372.
22. Wu, H.Y. On the complexity of turbo decoding algorithms. In Proceedings of the EEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202), Rhodes, Greece, 6–9 May 2001; Volume 2, pp. 1439–1443.
23. Bennett, C.H.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.* **2014**, *560*, 7–11. [CrossRef]