RESEARCH

Open Access

Polar code-based secure transmission with higher message rate combining channel entropy and computational entropy

Chen An[1,2], Mengjie Huang[1,2], Xianhui Lu[1,2*], Lei Bi[1,2] and Weijie Li[1,2]

Abstract

The existing physical layer security schemes, which are based on the key generation model and the wire-tap channel model, achieve security by utilizing channel reciprocity entropy and noise entropy, respectively. In contrast, we propose a novel secure transmission framework that combines noise entropy with reciprocity entropy, achieved by inserting reciprocity entropy into the frozen bits of polar codes. Note that in real-world scenarios, when eavesdroppers employ polynomial-time attacks, the bit error rate (BER) increases due to the introduction of computational entropy. To achieve indistinguishability security, we convert the practical physical layer security metric, BER, into the average min-entropy, a widely accepted concept in cryptography. The simulation results demonstrate that the eavesdropper's BER can be significantly increased without compromising the communication performance of the legitimate receiver. Under concrete parameters we selected, when compared to the joint scheme of physical layer key generation and one time pad, the modular semantically-secure scheme based on the wire-tap channel model, and the simple channel entropy combination scheme, our scheme achieves a message rate approximately 1.2 times, 3.8 times, and 1.4 times better, respectively. Experimental testing validates the feasibility of our scheme.

Keywords Secure transmission framework, Entropy combination, Polar codes, Physical layer security

Introduction

With the imminent arrival of the 6th generation wireless systems (6 G), secure transmission has become one of the core technologies in the field of cybersecurity. While upper-layer cryptographic algorithms are currently the most widely adopted solution, they often lack protective mechanisms for the physical layer itself (Sanenga et al. 2020). In response to this scenario, physical layer security has been introduced as a supplement of protection to conventional encryption techniques by making use of the random nature of wireless transmission media for ensuring communication secrecy (Liu et al. 2016). By introducing additional protective measures at the physical layer, overall security is enhanced, posing greater challenges for potential attackers attempting to breach the security defenses. According to the security resources used, existing physical layer secure transmission schemes can be divided into schemes based on the key generation model and schemes based on the wire-tap channel model (Hong 2020).

The key generation model was proposed by Maurer (1993). In Hershey et al. (1995) proposed to use the characteristics of wireless channels to generate the key. Wireless channels with short-term reciprocity, time variability and spatial decorrelation can provide the channel reciprocity entropy required for key generation. The typical process of the key generation model includes four steps: channel measurement, quantization, information

*Correspondence:
Xianhui Lu
luxianhui@iie.ac.cn
[1] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, CAS, No.19 Shucun Road, Haidian District, Beijing 100085, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, No.19 Yuquan Road, Shijingshan District, Beijing 100049, China

reconciliation, and privacy amplification (Guyue et al. 2014). In Li et al. (2020) provided an investigation on secure transmission achieved by one time pad and key generation from wireless channels.

The wire-tap channel model was constructed by Wyner (1975) in 1975, which assumes that the main channel between legitimate communication parties (Alice and Bob) is less noisy than the wire-tap channel to limit the amount of information obtained by the eavesdropper (Eve) (Hu and Li2014). When the quality of the main channel is better than that of the wire-tap channel, semantic security can be achieved by properly encoding the information sent by Alice. In modular semantically-secure schemes based on the wire-tap channel model (Bellare et al. 2012a, 2012b; Sharifian et al. 2017), the secure coding module guarantees security, and the error correction coding module ensures reliability.

The above schemes achieve security with large loss of message rate. It is an interesting problem whether security resources of the channel can be used more systematically and comprehensively to improve the message rate.

Related work

Error-correcting codes can be employed for the design of encryption schemes. The first code-based encryption scheme was pioneered by McEliece (1978). Subsequent research has seen several improvements based on this scheme (An et al. 2021). These schemes primarily emphasize the protection of message confidentiality while sacrificing the capability to correct errors in the communication channel. Conversely, our scheme achieves a dual objective of error correction and security.

In addition to modular semantically-secure schemes, another category of secure schemes based on the wire-tap channel model is code-based schemes (Wu et al. 2018). In practical applications, commonly used codes for physical layer security include low-density parity-check codes, polar codes, and lattice codes. Unlike our approach, these error-correcting codes require specific design considerations to efficiently leverage channel capacity, ultimately achieving the security goals of the scheme.

As of the year 2022, existing wire-tap coding schemes are generally information-theoretically secure. In Ishai et al. (2022) proposed a wire-tap coding scheme achieving computational security based on Ideal Obfuscation in the Oracle model. However, this scheme is merely a theoretical construction and lacks practical applicability. In Ishai et al. (2023) further developed a wire-tap coding scheme achieving computational security based on Indistinguishability Obfuscation in the plain model. Nevertheless, the practical usability of the scheme remains a challenge.

Based on the key generation model, Lu et al. (2018) designed a physical layer encryption algorithm based on the frozen bits of polar codes and chaotic sequences in 2018. In Lu et al. (2019) further studied the influence of different numbers of encrypted frozen bits and proposed a physical layer encryption algorithm based on partial frozen bits of polar codes and AES encrypter. These schemes measure security using bit error rate (BER), which lacks formal security assessment. Furthermore, they do not employ an extractor to extract channel noise entropy after inserting the key into the frozen bits, thus missing the opportunity to utilize noise entropy for ensuring secure message transmission.

In Kim et al. (2014) proposed a secure information transmission scheme with a secret key based on polar coding. Our proposal differs from theirs in the following aspects.

- *Design concept* Our goal is to leverage the security gains provided by polar codes. The security of Kim et al. (2014) actually comes from pre-processing messages with the key. Not inserting the key into the frozen bits already guarantees security, while inserting the fixed key into the frozen bits would reduce security. This is because the polarization of polar codes is not perfect in the scenario of finite code length, the attacker can obtain some information about the key by decoding the codeword, and the key can be recovered when enough packets are accumulated.
- *Security resources* Our security resources are derived from physical channels and represented by entropy. We achieve indistinguishability (IND) security by establishing a connection between BER and security parameters while demonstrating efficiency through the message rate. The security resources in Kim et al. (2014) are derived from upper-layer symmetric cryptography, and the security of polar codes and preprocessing are considered separately.

Our contributions

In this paper, we design a computational secure transmission framework by utilizing the structure of polar codes (Arikan 2009). Our contributions are summarized as follows.

Combine Channel Reciprocity Entropy and Noise Entropy We use the reciprocity entropy provided by the inherent randomness of the transmission channel to generate the key, and insert the key after information reconciliation into the frozen bits of polar codes. The polar coding module can combine channel reciprocity entropy and noise entropy to solve the current situation that the amount of reciprocity entropy extraction is insufficient or the assumption of noise entropy is not satisfied in some scenarios.

Enhance Channel Entropy by Introducing Computational Entropy In this paper, we take into account adversaries with polynomial-time attack capabilities. Assuming the adversary employs the best attack algorithm, such as first performing an exhaustive search on some key bits, and then using the successive cancellation list (SCL) decoding algorithm with the assistance of known key bits. The adversary reduces the BER at the expense of computational complexity, thereby decreasing the average min-entropy. In this process, compared to computationally unbounded adversaries, the introduction of computationally bounded adversaries' computational entropy enhances the reciprocity entropy and noise entropy within the channel.

From BER to Security Parameters We seamlessly connect the physical layer security metric BER, the information theory metric average min-entropy, and the entropy-based secure module, forming a cohesive link between BER and security parameters. Furthermore, we propose a BER-influence model after the secret key is inserted into the frozen bits of polar codes. In the analysis of simulation results, a set of heuristic rules is derived, which is subsequently employed to determine the selection of scheme parameters required to achieve the target security level.

Compact Information Reconciliation Method We propose a compact information reconciliation method that utilizes secure transmission to reduce the amount of information leakage caused by the information reconciliation step. Specifically, we securely encode some bits of the syndrome of the key required by the next message packet together with the current message packet to reduce the amount of leakage caused by the direct transmission of the syndrome.

Preliminaries

Polar codes

Polar codes, introduced by Arikan (2009), are the first error-correcting codes that provably achieve the capacity for any discrete memoryless channel. In order to better mine the security characteristics of polar codes, the related knowledge of polar codes, including polarization, encoding and decoding, is introduced.

Channel polarization

A polar code with a code length of N employs N independent copies of channel W to perform channel combining and channel splitting operations, resulting in the creation of N bit channels. As the code length increases, the bit channels exhibit two distinct extremes: some bit channels evolve into noiseless and reliable channels with a channel capacity approaching 1, while the remaining bit channels transform into entirely noisy and unreliable channels with a channel capacity nearing 0. Furthermore, as the code length N tends towards infinity, the proportion of reliable channels aligns with the channel capacity of channel W.

Polar encoding

Code constructions in this paper will be carried out in vector spaces over the binary field. The vector $\boldsymbol{u}$ of length N is denoted as $\boldsymbol{u} = [u_1, u_2, \cdots, u_N]$, and the matrix $\boldsymbol{G}_N = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n}$, where $N = 2^n$, with $\otimes$ representing the Kronecker product. The vector $\boldsymbol{u}$ is polar transformed to obtain the codeword $\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G}_N$.

In non-systematic polar encoding with a code length of N and a message length of K, the vector $\boldsymbol{u} = \{\boldsymbol{u}_A, \boldsymbol{u}_{A^c}\}$ where $A \subset \{1, \cdots, N\}$, $\boldsymbol{u}_A$ represents the message $\boldsymbol{m}$ of length K, and $\boldsymbol{u}_{A^c}$ represents the frozen bits (default $\mathbf{0}$) of length $(N - K)$ that have been pre-agreed upon by both legitimate parties. The elements in the set A correspond to the indices of K reliable bit channels following channel polarization, whereas the elements in the set A^c correspond to the indices of $(N - K)$ unreliable bit channels. The codeword $\boldsymbol{c}$ can be expressed as $\boldsymbol{c} = \boldsymbol{u}_A\boldsymbol{G}_A + \boldsymbol{u}_{A^c}\boldsymbol{G}_{A^c}$, where $\boldsymbol{G}_A$ and $\boldsymbol{G}_{A^c}$ denote the submatrices of $\boldsymbol{G}_N$ formed by the rows with indices in A and A^c, respectively.

Polar decoding

Typical polar decoding algorithms include the successive cancellation (SC) decoding algorithm (Arikan 2009) and the successive cancellation list (SCL) decoding algorithm (Tal and Vardy 2015; Balatsoukas-Stimming et al. 2015). The SCL decoding algorithm is an enhancement of the SC decoding algorithm, capable of storing up to L candidate paths during the decoding process, thereby reducing the probability of path errors. Tal and Vardy (2015) show that the decoding performance is very close to that of a ML decoder by setting an appropriate maximum path L. Their simulation results are reproduced in Fig. 1, but in a binary symmetric channel (BSC) with crossover probability p.

Key generation model

Physical layer key generation aims to achieve swift updates of wireless keys without relying on key derivation, accomplished through the utilization of shared random sources (Zhang et al. 2016). This research can be traced back to the key generation model initially proposed by Maurer (1993). Hershey et al. (1995) later proposed utilizing the characteristics of wireless channels for key generation.

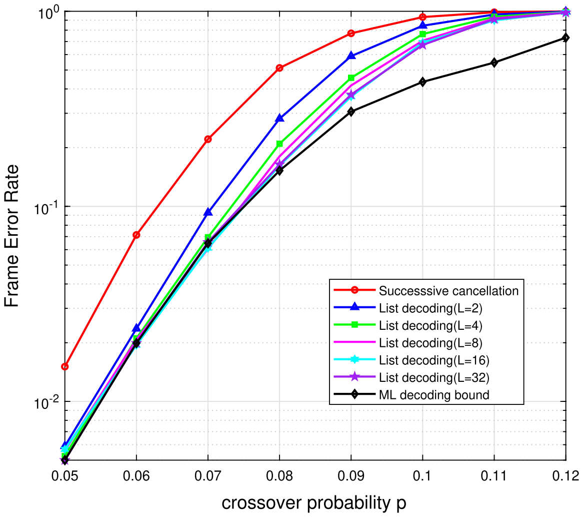

Fig. 1 Frame Error Rate of a length $N = 1024$, rate 1/2 polar code under various list sizes

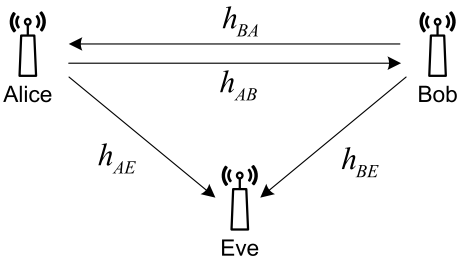

Fig. 2 Key generation model

The key generation model is illustrated in Fig. 2. Alice and Bob are legitimate users, and Eve represents the eavesdropper. In this model, Alice and Bob measure their common channel and will get noisy but correlated observations h_{AB} and h_{BA}. Key generation typically operates in time-division duplex mode, where all users operate on the same frequency. This ensures that the uplink and downlink channels are reciprocal, resulting in Alice and Bob obtaining roughly the same key. The time variability of the channel allows Alice and Bob to generate distinct keys during different time periods, facilitating timely key updates and ensuring randomness. Spatial decorrelation signifies that when Eve is located more than one half-wavelength away from either user, her measurements, denoted as h_{AE} and h_{BE}, exhibit no correlation with the measurements of Alice and Bob (Zhang et al. 2016).

The typical process of the key generation model includes four steps: channel measurement, quantization, information reconciliation and privacy amplification (Guyue et al. 2014).

- *Channel Measurement* Alice and Bob measure some characteristics of the channel by sending pilot signals to each other to obtain the time-varying value of the wireless channel between them. At present, common wireless channel characteristic parameters include received signal strength indication (RSSI), channel state information (CSI), channel phase (CP), and envelope.
- *Quantization* Convert the measured value into a string of key bits using different quantization methods. Quantization algorithms can generally be divided into two categories: lossy quantization and lossless quantization (Mathur et al. 2008; Nasrabadi and King 1988). It's worth noting that the key disagreement rate between Alice and Bob is approximately 0.09 (Zhang et al. 2016).
- *Information Reconciliation* Use the reconciliation protocol in a public noise-free channel to discard or correct inconsistencies in the key bits generated by Alice and Bob. The commonly used information reconciliation methods mainly include Cascade method (Brassard and Salvail 1993; Zhihua 2016) and error-correcting codes (Bloch et al. 2008; Ye et al. 2010).
- *Privacy Amplification* Discard partially consistent bits or perform some kind of bit transformation to strengthen the key, increase the entropy of the key and hide partial information that may be obtained by eavesdroppers during information reconciliation. The existing privacy amplification methods mainly include general hash functions and extractors (Bennett et al. 1995).

Wire-tap channel model

The main idea of the wire-tap channel model is that when the quality of the main channel is better than that of the wire-tap channel, information can be securely transmitted by designing a certain encoding method. In 1975, Wyner proposed the wire-tap channel model from the physical layer for the first time (Wyner 1975). Csiszár and Korner (1978) extend it to non-degraded discrete memoryless broadcast channels. As shown in Fig. 3, the main channel and the wire-tap channel are separated, and there is some noise interference in the main channel. When the noise entropy of the main channel is smaller than that of the wire-tap channel, semantic security can be achieved by properly encoding the message sent by Alice.

Modular semantically-secure transmission schemes based on the wire-tap channel model

The current modular semantically-secure transmission schemes (Bellare et al. 2012, 2012; Sharifian et al. 2017)

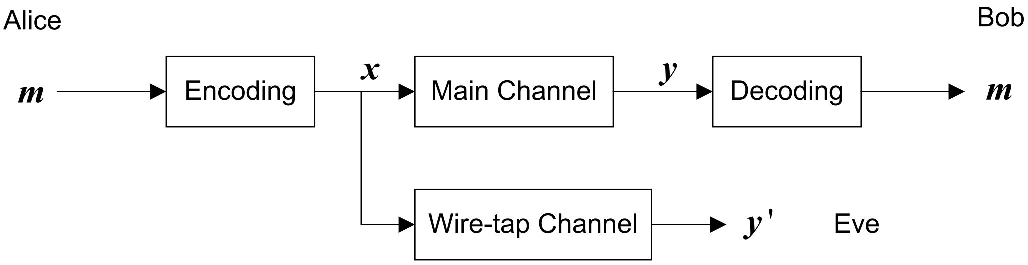

Fig. 3 Wire-tap channel model

based on the wire-tap channel model employ cryptographic primitives in the secure coding module to ensure security, and utilize error-correcting codes in the error correction coding module to guarantee reliability. These two modules operate independently.

Framework

The framework of modular semantically-secure transmission schemes based on the wire-tap channel model is shown in Fig. 4.

Alice sends a message packet $\boldsymbol{m}$, which is transformed into $\boldsymbol{x}$ through secure encoding. Following error correction encoding, $\boldsymbol{x}$ is further encoded into a codeword $\boldsymbol{c}$, which is subsequently transmitted over noisy channels.

Bob receives $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{e}$ represents the noise in Bob's channel. After error correction decoding and secure decoding, the correct message $\boldsymbol{m}$ is obtained.

Eve receives $\boldsymbol{y}' = \boldsymbol{c} + \boldsymbol{e}'$, where $\boldsymbol{e}'$ represents the noise in the wire-tap channel, i.e., $\boldsymbol{e}' > \boldsymbol{e} > 0$. Therefore, after Eve obtains $\boldsymbol{y}'$, $\boldsymbol{x}$ still retains some amount of entropy, which is measured by the average min-entropy $\widetilde{H}_\infty(\boldsymbol{x}|\boldsymbol{y}')$. Cryptographic primitives used in secure encoding can guarantee that the advantage of Eve getting any information about the message packet $\boldsymbol{m}$ is negligible.

RItE scheme

The RItE scheme proposed by Bellare et al. (2012) is a typical representative of modular semantically-secure schemes that can achieve secrecy capacity. For a single message packet, Bellare et al. define two functions *Ext* and *Inv*, and then illustrate that *Ext* is an extractor and *Inv* is its efficient inverse process.

Definition 1 (A function *Ext*) $\{0,1\}^K \times \{0,1\}^K \rightarrow \{0,1\}^{L_{sec}}$. For $\boldsymbol{s} \in \{0,1\}^K \setminus 0^K$, $\boldsymbol{x} \in \{0,1\}^K$, then $Ext(\boldsymbol{s},\boldsymbol{x}) = (\boldsymbol{s} \odot \boldsymbol{x})|_{L_{sec}}$, where K-bit strings can be interpreted as elements of the finite field $GF(2^K)$, $\odot$ represents multiplication operations over $GF(2^K)$, and $|_{L_{sec}}$ represents the first L_{sec} bits of the string.

Definition 2 (A function *Inv*) $\{0,1\}^K \times \{0,1\}^{K-L_{sec}} \times \{0,1\}^{L_{sec}} \rightarrow \{0,1\}^K$. For $\boldsymbol{s} \in \{0,1\}^K \setminus 0^K$, $\boldsymbol{r} \in \{0,1\}^{K-L_{sec}}$, $\boldsymbol{m} \in \{0,1\}^{L_{sec}}$, then $Inv(\boldsymbol{s},\boldsymbol{r},\boldsymbol{m}) = \boldsymbol{s}^{-1} \odot (\boldsymbol{m}\|\boldsymbol{r})$, where K-bit strings can be interpreted as elements of the finite field $GF(2^K)$, $\odot$ represents multiplication operations over $GF(2^K)$, and $\boldsymbol{s}^{-1}$ represents the multiplicative inverse of $\boldsymbol{s}$ in $GF(2^K)$.

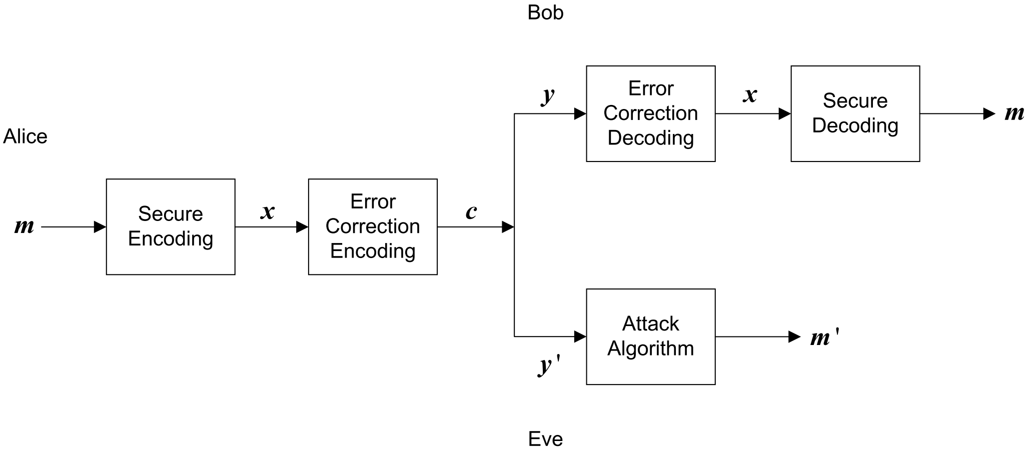

Fig. 4 Framework of modular semantically-secure transmission schemes based on the wire-tap channel model

Lemma 1 (Imported from Bellare et al. (2012), *Lemma* 10) *For all* $\alpha \in (0,1]$, *all* $L_{sec} \leq K - 2\log(1/\alpha) + 2$, *the function Ext is an* $(L_{sec} + 2\log(1/\alpha) - 2, \alpha)$*-extractor, and the function Inv is an efficient inverse process of Ext.*

Remark 1 A function $\boldsymbol{Ext} : \{0,1\}^K \times \{0,1\}^K \rightarrow \{0,1\}^{L_{sec}}$ is an $(\boldsymbol{L_{sec}} + 2\log(1/\alpha) - 2, \alpha)$-extractor if $SD((Ext(\boldsymbol{s},\boldsymbol{x}),\boldsymbol{y}',\boldsymbol{s}); (\boldsymbol{u},\boldsymbol{y}',\boldsymbol{s})) \leq \alpha$ for all pairs of (correlated) random variables $(\boldsymbol{x},\boldsymbol{y}')$ over $\{0,1\}^K \times \{0,1\}^N$ with $\widetilde{\boldsymbol{H}}_\infty(\boldsymbol{x}|\boldsymbol{y}') \geq \boldsymbol{L_{sec}} + 2\log(1/\alpha) - 2$, where *SD* represents the statistical distance between random variables, and additionally $\boldsymbol{s}$ and $\boldsymbol{u}$ are uniform on $\{0,1\}^K$ and $\{0,1\}^{L_{sec}}$, respectively.

The processes of the RItE scheme for the sender and receiver are shown in Algorithm 1 and 2, respectively. The notation $\xleftarrow{\$}$ signifies the operation of selecting an element at random from a set, while $\xleftarrow{L_{sec}}$ indicates that the bit string is grouped by L_{sec} bits. In this scheme, a message $\boldsymbol{M}$ is divided into L_{sec}-bit message packets, with a total of Q packets. $E : \{0,1\}^K \rightarrow \{0,1\}^N$ represents error correction encoding, and $D : \{0,1\}^N \rightarrow \{0,1\}^K$ represents error correction decoding.

Algorithm 1 $\mathcal{E}_{RItE}(\boldsymbol{M})$

```
$\boldsymbol{s} \xleftarrow{\$} \{0,1\}^K \setminus 0^K$
$\boldsymbol{m}[1], \ldots, \boldsymbol{m}[Q] \xleftarrow{L_{sec}} \boldsymbol{M}$
for $i = 1, \ldots, Q$ do
    $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^{K-L_{sec}}$
    $\boldsymbol{x}[i] \longleftarrow Inv(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{m}[i])$
    $\boldsymbol{c}[i] \longleftarrow E(\boldsymbol{x}[i])$
end for
return $E(\boldsymbol{s})\|\boldsymbol{c}[1]\| \cdots \|\boldsymbol{c}[Q]$
```

Algorithm 2 $\mathcal{D}_{RItE}(\boldsymbol{Y})$

```
$\boldsymbol{y}[0], \boldsymbol{y}[1], \ldots, \boldsymbol{y}[Q] \xleftarrow{N} \boldsymbol{Y}$
$\boldsymbol{s} \longleftarrow D(\boldsymbol{y}[0])$
for $i = 1, \ldots, Q$ do
    $\boldsymbol{x}[i] \longleftarrow D(\boldsymbol{y}[i])$
    $\boldsymbol{m}[i] \longleftarrow Ext(\boldsymbol{s}, \boldsymbol{x}[i])$
end for
return $\boldsymbol{m}[1]\|\boldsymbol{m}[2]\| \cdots \|\boldsymbol{m}[Q]$
```

Bellare et al. point out that the RItE scheme achieves semantic security if the channel is symmetric, *Inv* is the inverse process of an $(L_{sec} + 2\log(1/\alpha) - 2, \alpha)$-extractor, and the average min-entropy of a single message packet $\widetilde{H}_\infty(\boldsymbol{x}|\boldsymbol{y}') \geq L_{sec} + 2\log(1/\alpha) - 2$, where α is the security parameter such as 2^{-128}. According to Lemma 1, it can be known that the K-bit secure encoding output obtained through the inverse process of the extractor can ensure the security of the L_{sec}-bit message packet, where $L_{sec} = \lfloor \widetilde{H}_\infty(\boldsymbol{x}|\boldsymbol{y}') - 2\log(1/\alpha) + 2 \rfloor$. In other words, cryptographic primitives used in secure encoding can guarantee that the advantage of Eve getting any information about the message packet $\boldsymbol{m}$ is negligible.

In the RItE scheme, the average min-entropy of $\boldsymbol{x}$ in the wire-tap channel is guaranteed by the noise entropy based on the assumptions of the wire-tap channel model. It is essential to stress that the error-correcting code in this scheme is dedicated solely to error correction and lacks the capacity to offer security assurances.

Our framework

To address the current issue of inadequate reciprocity entropy or noise entropy, the most straightforward approach is to combine these two types of entropy. A simple channel entropy combination scheme we can think of is: taking the polar code-based secure transmission scheme (Wu et al. 2018) as an example, the message can be XORed with the reciprocity entropy and then inserted into bit channels with good reliability for both Bob and Eve. This increases the secrecy capacity, which was originally assured solely by noise entropy. This section specifically delves into the scenario of inserting reciprocity entropy into the frozen bits of polar codes.

An outline of our framework

Our secure transmission framework for a single message packet is shown in Fig. 5. We use polar codes to combine secure coding with error correction coding, achieving secure and reliable transmission between the sender and receiver.

The key extracted by Alice and Bob from the channel and after information reconciliation is denoted as $\boldsymbol{k}$, and it is inserted into the frozen bits of polar codes. Simultaneously, the key eavesdropped by Eve is represented as $\boldsymbol{k}'$.

Alice sends a message packet $\boldsymbol{m}$, which is transformed into $\boldsymbol{x}$ through secure encoding. Following polar encoding, $\boldsymbol{x}$ is further encoded into a codeword $\boldsymbol{c}$, which is subsequently transmitted over noisy channels.

Bob receives $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{e}$ represents the noise in Bob's channel. Since the key inserted into the frozen bits has no effect on Bobâ€™s decoding ability, Bob obtains the correct message after polar decoding and secure decoding. Polar codes play a role in correcting errors in the channel.

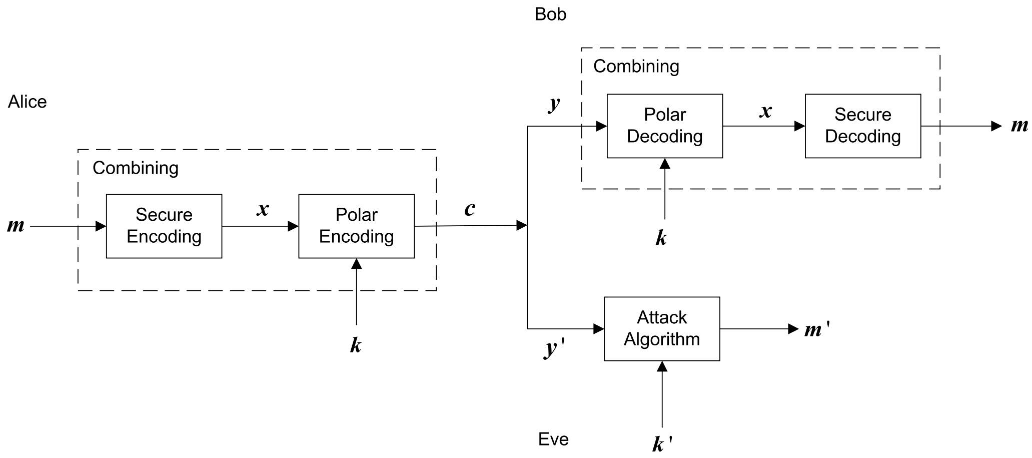

Fig. 5 Our secure transmission framework for a single message packet

Eve receives $\boldsymbol{y}' = \boldsymbol{c} + \boldsymbol{e}'$, where $\boldsymbol{e}'$ represents the noise in Eve's channel, i.e., $\boldsymbol{e}' > 0$. For Eve, the key bits in the frozen bits that it does not know increase its BER. Then, according to the relationship between the average min-entropy and BER given by Sason and Verdú (2017), we calculate the average min-entropy $\widetilde{H}_\infty(\boldsymbol{x}|\boldsymbol{y}') = \lfloor -K \log(1 - p_E) \rfloor$, where K represents the length of $\boldsymbol{x}$, and p_E represents Eve's BER. Same as the modular semantically-secure transmission based on the wire-tap channel model, cryptographic primitives used in secure encoding can guarantee that the advantage of Eve getting any information about the message packet $\boldsymbol{m}$ is negligible. Polar codes not only serve for error correction but also provide the required average min-entropy for secure coding.

In other words, with a given security parameter α and the (N, K)-polar code, the length of the frozen bits is $(N - K)$. Therefore, the length of the key that can be inserted into the frozen bits after information reconciliation is denoted as L_k, where $0 < L_k \leq N - K$. The key bits in the frozen bits that Eve does not know increase its BER. The average min-entropy is calculated based on Eve's BER. Then, the length of the message packet $\boldsymbol{m}$ that can achieve security is determined. Note that when Eve is computationally bounded, the calculated value based on BER is computational average min-entropy, and the achieved security is computational security.

Detailed description

In this subsection, we refer to the secure coding module in the RItE scheme to provide an instantiated scheme within our framework. This scheme comprises four main components: increase Eve's BER by combining channel entropy and computational entropy using polar codes, generate a consistent key between Alice and Bob after information reconciliation, implement secure coding based on Eve's BER and accomplish information reconciliation compactly. The processes for the sender and receiver are shown in Algorithm 3 and 4, respectively.

Algorithm 3 $\mathcal{E}(M)$

```
$L_{syn_s} := \max(H_0 - (L_k - L_{syn}), 0)$
$B := L_{sec} - L_{syn_s}$
$Q := \lceil L_M / B \rceil$
$\boldsymbol{s} \xleftarrow{\$} \{0,1\}^K \setminus 0^K$
$\boldsymbol{k}_A[1] \longleftarrow \boldsymbol{k}_0$
$\boldsymbol{m}[1], \ldots, \boldsymbol{m}[Q] \xleftarrow{B} \boldsymbol{M}$
for $i = 1, \ldots, Q$ do
    $\boldsymbol{k}_A[i+1] \longleftarrow KeyGen2$
    $\boldsymbol{r}[i] \xleftarrow{\$} \{0,1\}^{K - L_{sec}}$
    $\boldsymbol{x}[i] \longleftarrow Inv(\boldsymbol{s}, \boldsymbol{r}[i], \boldsymbol{m}[i] \| Syn(\boldsymbol{k}_A[i+1])|_{L_{syn_s}})$
    $\boldsymbol{syn}_d[i+1] \longleftarrow Syn(\boldsymbol{k}_A[i+1])|^{L_{syn} - L_{syn_s}}$
    $\boldsymbol{c}[i] \longleftarrow PE(\boldsymbol{x}[i], \boldsymbol{k}_A[i])$
end for
return $E(\boldsymbol{s}) \| \boldsymbol{c}[1] \| \cdots \| \boldsymbol{c}[Q] \| \boldsymbol{syn}_d[2] \| \cdots \| \boldsymbol{syn}_d[Q]$
```

Algorithm 4 $\mathcal{D}(\boldsymbol{Y})$

```
$L_{syn_s} := \max(H_0 - (L_k - L_{syn}), 0)$
$L_{syn_d} := L_{syn} - L_{syn_s}$
$B := L_{sec} - L_{syn_s}$
$\boldsymbol{Y}_1 \| \boldsymbol{Y}_2 \longleftarrow \boldsymbol{Y}$
$\boldsymbol{y}_1[0], \ldots, \boldsymbol{y}_1[Q] \xleftarrow{N} \boldsymbol{Y}_1$
$\boldsymbol{y}_2[2], \ldots, \boldsymbol{y}_2[Q] \xleftarrow{L_{syn_d}} \boldsymbol{Y}_2$
$\boldsymbol{k}_A[1] \longleftarrow \boldsymbol{k}_0$
$\boldsymbol{s} \longleftarrow D(\boldsymbol{y}_1[0])$
for $i = 1, \ldots, Q$ do
    $\boldsymbol{k}_B[i+1] \longleftarrow KeyGen2$
    $\boldsymbol{x}[i] \longleftarrow PD(\boldsymbol{y}_1[i], \boldsymbol{k}_A[i])$
    $\boldsymbol{m}[i] \longleftarrow Ext(\boldsymbol{s}, \boldsymbol{x}[i])|_B$
    $\boldsymbol{syn}_s[i+1] \longleftarrow Ext(\boldsymbol{s}, \boldsymbol{x}[i])|^{L_{syn_s}}$
    $\boldsymbol{k}_A[i+1] \longleftarrow Rec(\boldsymbol{k}_B[i+1], \boldsymbol{syn}_s[i+1] \| \boldsymbol{y}_2[i+1])$
end for
return $\boldsymbol{m}[1] \| \boldsymbol{m}[2] \| \cdots \| \boldsymbol{m}[Q]$
```

Increase Eve's BER by Combining Channel Entropy and Computational Entropy Using Polar Codes We denote $PE : \{0,1\}^K \times \{0,1\}^{L_k} \rightarrow \{0,1\}^N$ as the polar encoding for K-bit $\boldsymbol{x}$ with L_k-bit key inserted into the frozen bits and $PD : \{0,1\}^N \times \{0,1\}^{L_k} \rightarrow \{0,1\}^K$ as its decoding. Due to the structure of polar codes, Eve's BER will increase after executing *PE*. Configuring the parameters as follows: (N, K)-polar code, p_B (the BER of the legitimate channel), p_{E_0} (the original BER of the eavesdropping channel) and H_0 (the minimum value of the key's entropy for achieving security). These settings ensure that Bob can decode correctly while maintaining Eve's BER at p_E. The p_E is the combined result of the original noise in the eavesdropping channel, the randomness of the key inserted into the frozen bits and the computational entropy introduced by Eve's attack algorithm. In our scheme, Alice and Bob employ physical layer key generation to obtain a consistent key, insert the key into the frozen bits of polar codes, and realize the combination of channel entropy and computational entropy to increase Eve's BER.

Generate A Consistent Key Between Alice and Bob after Information Reconciliation In our scheme, the consistent key used by Alice and Bob is not necessarily uniformly random; thus, it can be obtained through the first three steps of physical layer key generation without the need for privacy amplification. For simplicity, we assume that Alice and Bob have generated a consistent key $\boldsymbol{k}_0$ with an entropy of H_0 bits and a length not exceeding $(N-K)$ bits. We can amortize its impact on the message rate to essentially zero by transmitting a large number of message packets. Note that the generated $\boldsymbol{k}_0$ by Alice and Bob must be consistent; otherwise, it is equivalent to inserting keys unknown to Bob into the frozen bits of polar codes. In such a scenario, Bob's position is comparable to Eve. After executing algorithm *PE*, Bob's BER increases, compromising the guarantee of correct message transmission. Skipping the specific steps, we denote *KeyGen*2 as the interface that provides Alice (Bob) with a L_k-bit long key, represented as $\boldsymbol{k}_A (\boldsymbol{k}_B)$ after quantization. Under the assumption that the distance between Eve and either legitimate user is greater than one half-wavelength, it can be considered that the L_k-bit $\boldsymbol{k}_A (\boldsymbol{k}_B)$ is uniformly random (Zhang et al. 2016), meaning it possesses an entropy of L_k bits. However, due to the imperfection of the channel reciprocity, the key disagreement rate between $\boldsymbol{k}_A$ and $\boldsymbol{k}_B$ is about 0.09 (Zhang et al. 2016). To ensure consistency, Alice generates a syndrome for $\boldsymbol{k}_A$ using the function $Syn : \{0,1\}^{L_k} \rightarrow \{0,1\}^{L_{syn}}$ and transmits it to Bob through a public noise-free channel. Bob, in turn, employs both $\boldsymbol{k}_B$ and the syndrome received from Alice to reconstruct $\boldsymbol{k}_A$ using the function $Rec : \{0,1\}^{L_k} \times \{0,1\}^{L_{syn}} \rightarrow \{0,1\}^{L_k}$. Without loss of generality, it can be assumed that the *Syn* is optimal, so that the length of the syndrome can be limited to $L_{syn} = \lceil L_k(1/C_{0.09} - 1) \rceil$ bits, where $C_{0.09}$ is the capacity of the BSC with crossover probability 0.09. Specifically, $C_{0.09}$ can be calculated as: $C_{0.09} = 1 + 0.09 \log 0.09 + (1-0.09) \log(1-0.09)$. Unfortunately, the syndrome can be eavesdropped by Eve. Considering that the upper bound of the leaked information is equal to the syndrome's length, the entropy of the key will be reduced to $(L_k - L_{syn})$ bits. As a result, the entropy of the consistent key may not be sufficient to increase Eve's BER to meet the average min-entropy required for secure transmission. We will propose a compact information reconciliation method to address this problem in the following.

Implement Secure Coding Based on Eve's BER The crossover probabilities of the legitimate channel and the eavesdropping channel are denoted as p_B and p_{E_0}, respectively. In this scenario, error-correcting codes are used to correct errors in the legitimate channel, and at the same time, the BER of Eve will be increased to p_E. We can state that the average min-entropy of the K-bit random string $\boldsymbol{x}$ for Eve after receiving $\boldsymbol{y}'$ is $\lfloor -K \log(1-p_E) \rfloor$ bits, and Bob can correctly recover $\boldsymbol{x}$. In the secure encoding module, the L_{sec}-bit secret string and the L_r-bit random string $\boldsymbol{r}$ are input into *Inv* to obtain the secure codeword $\boldsymbol{x}$, where $L_{sec} = \lfloor -K \log(1-p_E) \rfloor - 2\log(1/\alpha) + 2$ and $L_r = K - L_{sec}$. By inputting $\boldsymbol{x}$ into *Ext*, Bob can get the secret string. Meanwhile, Eve cannot obtain any information about it (with security parameter α).

In conclusion, L_{sec} bits can be securely transmitted between Alice and Bob using the K-bit $\boldsymbol{x}$.

Accomplish Information Reconciliation Compactly Our scheme reduces the amount of information leakage during the information reconciliation process. In order to ensure that the entropy of the consistent key obtained by Alice and Bob after information reconciliation is not less than H_0 bits, it may be necessary to securely transmit some bits of the syndrome. Specifically, we denote the first $\max(H_0 - (L_k - L_{syn}), 0)$ bits of the syndrome (where L_k is the length of $\boldsymbol{k}_A[i]$ and $\boldsymbol{k}_B[i]$, L_{syn} is the length of $\boldsymbol{k}_A[i]$'s syndrome) as $\boldsymbol{syn}_s$, which must be securely transmitted. The last $(L_{syn} - L_{syn_s})$ bits of the syndrome $Syn(\boldsymbol{k}_A[i])$, which we denote as $Syn(\boldsymbol{k}_A[i])\Big|^{L_{syn}-L_{syn_s}}$, will be transmitted directly. We set the size of message packet as $B = L_{sec} - L_{syn_s}$ bits and denote the concatenation of the i—th message packet and the first L_{syn_s} bits of $\boldsymbol{k}_A[i+1]$'s syndrome as $\boldsymbol{m}[i]\|Syn(\boldsymbol{k}_A[i+1])|_{L_{syn_s}}$. Then Alice uses Inv for secure encoding to obtain $\boldsymbol{x}[i]$ and executes $PE(\boldsymbol{x}[i], \boldsymbol{k}_A[i])$ to ensure the secure transmission of the message packet $\boldsymbol{m}[i]$ and the first L_{syn_s} bits of $\boldsymbol{k}_A[i+1]$'s syndrome. To accomplish the information reconciliation of $\boldsymbol{k}_A[i+1]$, Bob executes the $Rec(\boldsymbol{k}_B[i+1], \boldsymbol{syn}_s[i+1]\|\boldsymbol{y}_2[i+1])$. Therefore, legitimate users can obtain a consistent key $\boldsymbol{k}_A[i+1]$ with an entropy of no less than H_0 bits. It is worth noting that, to maintain key consistency, we propose two error correction mechanisms to ensure correct transmission without leaking redundant information to Eve. (1) Error Correction Encoding of Keyâ€™s Syndrome Bits: We apply error correction encoding to the key's syndrome bits before transmission. This method is suitable for scenarios with significant channel quality fluctuations, as the introduction of redundancy enhances resilience against channel noise. (2) Retransmission Mechanism: Since keys cannot be reused, retransmission requires the extraction of the key anew from the channel. We perform cyclic redundancy check (CRC) on the key's syndrome bits before transmission. Upon reception, if Bob identifies that the syndrome bits did not pass the check, both parties renegotiate the key. This approach is effective in scenarios with minor channel quality fluctuations.

Security analysis

In this section, we analyse the security of the key generation model and our scheme.

Security of the key generation model

The main idea of the key generation model is that legitimate communication parties leverage the physical characteristics of the channel, such as reciprocity, time variability, and spatial decorrelation, to generate consistent, random, and secure keys. The key generation process at the physical layer involves channel measurement, quantization, information reconciliation, and privacy amplification. In our scheme, we do not perform privacy amplification during the key generation phase but achieve a similar effect during information transmission using an information-theoretic extractor.

In the measurement step, according to the conclusion in reference (Zhang et al. 2016), when Eve is located more than one half-wavelength away from either user, the eavesdropping channel is considered uncorrelated with the legitimate channel, indicating that this process involves no information leakage. Quantization is performed locally by Alice and Bob, with no information leaked to Eve. The information reconciliation process typically involves the use of error correction codes. For example, Alice sends the syndrome bits of $\boldsymbol{k}_A$ to Bob. Upon reception, Bob corrects $\boldsymbol{k}_B$ based on these syndrome bits. Although this reconciliation process leaks information about the key, the disclosed amount does not exceed the information sent during the reconciliation process, such as the length of the syndrome bits. We also use the length of the syndrome bits as an upper bound for the leaked information. To reduce information leakage, we divide the syndrome bits into two parts. One part is transmitted directly, contributing to the information leakage equal to the length of this part. The other part is securely transmitted along with the previous message packet, causing no information leakage. This division serves to minimize the overall information leakage during the information reconciliation process, ensuring that both parties have sufficient entropy in the key used for inserting frozen bits. It is crucial to note that, in this process, we should employ appropriate error correction mechanisms to guarantee the correct transmission of both parts of the syndrome bits, thereby ensuring that Alice and Bob negotiate a consistent key.

In our scheme, after information reconciliation, keys with sufficient entropy are inserted into the frozen bits of polar codes. By leveraging the BER-influence model, we can deduce Eve's BER. Subsequently, we seamlessly connect the physical layer security metric (BER), the information theory metric (average min-entropy), and the entropy-based secure module, forming a cohesive link between BER and security parameters. Detailed security analysis is provided in the following subsection.

Security of our scheme

BER-Based Computational Average Min-Entropy

When inserting a key into the frozen bits of polar code, the adversary may employ the following three attack strategies:

(1) The information set decoding (ISD) algorithm is employed; however, this algorithm is designed for random codes and requires that the Hamming weight of the error vector is less than the minimum distance of the code.
(2) The currently best decoding algorithm for messages, the SCL algorithm, is employed, which has a complexity of $O(N \log N)$. However, its decoding performance decreases as the number of keys bits increases.
(3) An exhaustive search is performed on a portion of the key bits, followed by the combination with the SCL decoding algorithm. The complexity of this algorithm is $O(2^{H_{sk}} \cdot N \log N)$, where H_{sk} represents the key's entropy under exhaustive search. In other words, the attack complexity is mainly determined by the exhaustive search algorithm. Additionally, since only a portion of the key bits is being searched, it's necessary to combine the results with the SCL algorithm for attacking the message bits. Therefore, it's not feasible to leverage algorithms like meet-in-the-middle to accelerate the exhaustive search.

In this paper, we make the assumption that for computationally bounded adversaries, the BER after employing the best attack algorithm is denoted as p_E.

Definition 3 (BER-Based Computational Average Min-Entropy) The BER-based computational average min-entropy can be calculated as $\widetilde{H}^c_\infty(\boldsymbol{x}|\boldsymbol{y}') = \lfloor -K \log(1-p_E) \rfloor$, where K represents the length of $\boldsymbol{x}$. In other words, for any computationally bounded adversary, the uncertainty of $\boldsymbol{x}$ given $\boldsymbol{y}'$ is no less than $\widetilde{H}^c_\infty(\boldsymbol{x}|\boldsymbol{y}')$.

Secure Transmission Bellare et al. (2012) provided the definition of semantic security against unbounded adversaries. In cryptography, the common assumption of adversaries being computationally bounded should result in improved feasibility outcomes. We define the indistinguishability (IND) and IND for random messages (IND-R) of secure transmission against probabilistic polynomial-time (PPT) adversaries.

Definition 4 (IND of Secure Transmission) A transmission scheme $\mathcal{T} = (\mathcal{E}, \mathcal{D})$ achieves indistinguishability for a pair of noisy channels (*ChB*, *ChE*) and message space $\mathcal{M} = \{0,1\}^{L_{sec}}$ if there exists negligible functions $\epsilon(\lambda), \mu(\lambda)$ such that

(1) Correctness: for all messages $\boldsymbol{m} \in \mathcal{M}$, $\mathbf{Pr}[\mathcal{D}(1^\lambda, ChB(\mathcal{E}(1^\lambda, \boldsymbol{m}))) = \boldsymbol{m}] \geq 1 - \epsilon(\lambda)$

(2) Security: for all PPT adversaries $\mathcal{A}$ and all $\boldsymbol{m_0}, \boldsymbol{m_1} \in \mathcal{M}$,

$$\begin{aligned}\mathbf{Adv}^{\text{IND}}_{\mathcal{E}}(1^\lambda) = \underset{\mathcal{A}, \boldsymbol{m_0}, \boldsymbol{m_1}}{max} &|\mathbf{Pr}[\mathcal{A}(\mathcal{E}(\boldsymbol{m_0})) = 1] \\ &- \mathbf{Pr}[\mathcal{A}(\mathcal{E}(\boldsymbol{m_1})) = 1]| \\ &\leq \mu(\lambda).\end{aligned}$$

Definition 5 (IND-R of Secure Transmission) $\mathcal{T} = (\mathcal{E}, \mathcal{D})$ achieves indistinguishability for random messages on $\mathcal{M} = \{0,1\}^{L_{sec}}$ for a pair of noisy channels (*ChB*, *ChE*) if there exists negligible functions $\mu(\lambda)$ such that for all PPT adversaries $\mathcal{A}$,

$$\begin{aligned}\mathbf{Adv}^{\text{IND-R}}_{\mathcal{E}}(1^\lambda) = \underset{\mathcal{A}}{max} &|\mathbf{Pr}[\mathcal{A}(\mathcal{E}(\boldsymbol{u}_{L_{sec}}), \boldsymbol{u}_{L_{sec}}) = 1] \\ &- \mathbf{Pr}[\mathcal{A}(\mathcal{E}(\boldsymbol{u}'_{L_{sec}}), \boldsymbol{u}_{L_{sec}}) = 1]| \\ &\leq \mu(\lambda),\end{aligned}$$

where $\boldsymbol{u}_{L_{sec}}$ and $\boldsymbol{u}'_{L_{sec}}$ are uniformly distributed over $\mathcal{M}$.

IND-R of Our Scheme Let's first analyze the IND-R security of our scheme, specifically its security when messages come from a random distribution. Then, we can derive the security of the scheme for messages from any distribution based on the relationship between IND-R and IND security.

Lemma 2 *Let the BSC ChE ensure that the BER of guessing $\boldsymbol{x}$ from $ChE^N(PE(\boldsymbol{x}, \boldsymbol{k}))$ for any PPT adversary is not less than p_E, where $\boldsymbol{x}$ is uniformly distributed over $\{0,1\}^K$, and $ChE^N(\cdot)$ represents the independent use of ChE N times. Then for transmission scheme $\mathcal{E} : \{0,1\}^{L_{sec}} \to \{0,1\}^N$ defined in Algorithm 3, there is*

$$\mathbf{Adv}^{\text{IND-R}}_{\mathcal{E}}(1^\lambda) \leq 2^{-\frac{\lfloor -K \log(1-p_E) \rfloor - L_{sec}}{2} - 1}.$$

Proof The game sequences used in the proof are illustrated in Fig. 6. In Game 0, the challenger chooses a uniformly random message $\boldsymbol{m}$ and calculates the N-length codeword $\boldsymbol{c}$. The adversary $\mathcal{A}$ gets $\boldsymbol{y}'$, the seed $\boldsymbol{s}$, and $\boldsymbol{m}$. In Game 2, $\mathcal{A}$ gets $\boldsymbol{y}''$ which is independent of $\boldsymbol{m}$, the seed $\boldsymbol{s}$, and $\boldsymbol{m}$. We claim that $\mathbf{Adv}^{\text{IND-R}}_{\mathcal{E},\mathcal{A}}(1^\lambda) = |\mathbf{Pr}[(Game\ 0) = 1] - \mathbf{Pr}[(Game\ 2) = 1]|$. In Game 1, we first sample $\boldsymbol{x}$ uniformly at random from $\{0,1\}^K$ and then set $\boldsymbol{m}$ to $Ext(\boldsymbol{s}, \boldsymbol{x})$. The output distribution of Game 1 and Game 0 are identical because of the regularity of *Ext*.
The BER of guessing the uniformly random $\boldsymbol{x}$ from $\boldsymbol{y}'$ is p_E, then the computational average min-entropy of $\boldsymbol{x}$ based on $\boldsymbol{y}'$ is $\lfloor -K \log(1-p_E) \rfloor$ from Definition 3. According to Lemma 1, *Ext* is a $(\lfloor -K \log(1-p_E) \rfloor, 2^{-\frac{\lfloor -K \log(1-p_E) \rfloor - L_{sec}}{2} - 1})$-extractor, we conclude that

Game 0	Game 1	Game 2
$\boldsymbol{s} \xleftarrow{\$} \{0,1\}^K \setminus 0^K$ $\boldsymbol{m} \xleftarrow{\$} \{0,1\}^{L_{sec}}$ $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^{K-L_{sec}}$ $\boldsymbol{x} \leftarrow Inv(\boldsymbol{s},\boldsymbol{r},\boldsymbol{m})$ $\boldsymbol{c} \leftarrow PE(\boldsymbol{x},\boldsymbol{k})$ $\boldsymbol{y}' \leftarrow ChE(\boldsymbol{c})$ $b' \leftarrow \mathcal{A}(\boldsymbol{y}',\boldsymbol{m},\boldsymbol{s})$ **return** b'	$\boldsymbol{s} \xleftarrow{\$} \{0,1\}^K \setminus 0^K$ $\boldsymbol{x} \xleftarrow{\$} \{0,1\}^K$ $\boldsymbol{m} \leftarrow Ext(\boldsymbol{s},\boldsymbol{x})$ $\boldsymbol{c} \leftarrow PE(\boldsymbol{x},\boldsymbol{k})$ $\boldsymbol{y}' \leftarrow ChE(\boldsymbol{c})$ $b' \leftarrow \mathcal{A}(\boldsymbol{y}',\boldsymbol{m},\boldsymbol{s})$ **return** b'	$\boldsymbol{s} \xleftarrow{\$} \{0,1\}^K \setminus 0^K$ $\boldsymbol{m} \xleftarrow{\$} \{0,1\}^{L_{sec}}$ $\boldsymbol{m}' \xleftarrow{\$} \{0,1\}^{L_{sec}}$ $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^{K-L_{sec}}$ $\boldsymbol{x}' \leftarrow Inv(\boldsymbol{s},\boldsymbol{r},\boldsymbol{m}')$ $\boldsymbol{c}' \leftarrow PE(\boldsymbol{x}',\boldsymbol{k})$ $\boldsymbol{y}'' \leftarrow ChE(\boldsymbol{c}')$ $b' \leftarrow \mathcal{A}(\boldsymbol{y}'',\boldsymbol{m},\boldsymbol{s})$ **return** b'

Fig. 6 Games for the proof of Lemma 2

$$\begin{aligned}\mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND-R}}(1^\lambda) &= \max_{\mathcal{A}} |\mathbf{Pr}[(Game\ 1) = 1] \\ &\quad - \mathbf{Pr}[(Game\ 2) = 1]| \\ &\le 2^{-\frac{\lfloor -K\log(1-p_E)\rfloor - L_{sec}}{2} - 1}.\end{aligned}$$

□

IND-R Implies IND

Lemma 3 (Imported from Arikan (2009), *Proposition* 13 *and Bellare and Tessaro* (2012), *Lemma* 5.7) *Let ChE be a BSC, and let* $Ch_{\boldsymbol{s},\boldsymbol{k}} : \{0,1\}^{L_{sec}} \to \{0,1\}^N$ *be the channel that, given an input* $\boldsymbol{m} \in \{0,1\}^{L_{sec}}$, *outputs* $ChE^N(\mathcal{E}(\boldsymbol{m}))$ *for all* $\boldsymbol{k} \in \{0,1\}^{N-K}$ *and* $\boldsymbol{s} \in \{0,1\}^K \backslash 0^K$, *where* $\mathcal{E} : \{0,1\}^{L_{sec}} \to \{0,1\}^N$ *is defined in Algorithm* 3. *Then* $Ch_{\boldsymbol{s},\boldsymbol{k}}$ *is symmetric for all* $\boldsymbol{s}$ *and* $\boldsymbol{k}$.

Lemma 4 *For BSC ChE and transmission scheme* $\mathcal{E}$ *defined in Algorithm* 3, *there is* $\mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND}}(1^\lambda) \le 2 \cdot \mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND-R}}(1^\lambda)$.

Proof The IND-R advantage for transmission scheme $\mathcal{E}$ and any PPT adversary $\mathcal{A}$ is

$$\begin{aligned}\mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND-R}} &= \mathbb{E}_{\boldsymbol{s}} \max_{\mathcal{A}}(|\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}_{L_{sec}}), \boldsymbol{u}_{L_{sec}}) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}'_{L_{sec}}), \boldsymbol{u}_{L_{sec}}) = 1]|) \\ &= \frac{1}{2^{L_{sec}}}\mathbb{E}_{\boldsymbol{s}} \max_{\mathcal{A}}(\sum_{\boldsymbol{m}} |\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{m})) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}'_{L_{sec}})) = 1]|) \\ &= \mathbb{E}_{\boldsymbol{s}}[\delta].\end{aligned} \tag{1}$$

For the symmetry of $Ch_{\boldsymbol{s},\boldsymbol{k}}$ according to Lemma 3, there is a permutation $\pi_{\boldsymbol{i},\boldsymbol{j}} : \{0,1\}^N \to \{0,1\}^N$ for any $\boldsymbol{i},\boldsymbol{j} \in \{0,1\}^{L_{sec}}$ such that $\mathbf{Pr}[(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{i})) = \boldsymbol{v}] = \mathbf{Pr}[Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{j}) = \pi_{\boldsymbol{i},\boldsymbol{j}}(\boldsymbol{v})]$ for any $\boldsymbol{v} \in \{0,1\}^N$. It's clearly that there is a $\delta > 0$ such that

$$\begin{aligned}\delta &= \max_{\mathcal{A}}(|\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{i})) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}'_{L_{sec}})) = 1]|) \\ &= \max_{\mathcal{A}}(|\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{j})) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}'_{L_{sec}})) = 1]|)\end{aligned}$$

for any $\boldsymbol{i}$ and $\boldsymbol{j}$. Then we get Eq. (1).

$$\begin{aligned}\mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND}} &= \mathbb{E}_{\boldsymbol{s}} \max_{\mathcal{A},\boldsymbol{m_0},\boldsymbol{m_1}} ([|\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{m_0})) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{m_1})) = 1]) \\ &\le \mathbb{E}_{\boldsymbol{s}} \max_{\mathcal{A}}((|\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{m_0})) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}_{L_{sec}})) = 1]|) \\ &\quad + (|\mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{m_1})) = 1] \\ &\quad - \mathbf{Pr}[\mathcal{A}(Ch_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{u}_{L_{sec}}))) = 1]|)) \\ &= 2 \cdot \mathbb{E}_{\boldsymbol{s}}[\delta] \\ &= 2 \cdot \mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND-R}}.\end{aligned}$$

□

Theorem 1 is immediately derived by combining Lemma 2 and Lemma 4, providing the security analysis of our scheme.

Theorem 1 *Let the BSC ChE ensure that for any PPT adversary, the BER when guessing* $\boldsymbol{x}$ *from* $ChE^N(PE(\boldsymbol{x},\boldsymbol{k}))$ *is not less than* p_E, *where* $\boldsymbol{x}$ *is uniformly distributed over* $\{0,1\}^K$. *Then for the transmission scheme* $\mathcal{E} : \{0,1\}^{L_{sec}} \to \{0,1\}^N$ *defined in Algorithm* 3, *there is*

$$\mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND}}(1^\lambda) \le 2 \cdot 2^{-\frac{\lfloor -K\log(1-p_E)\rfloor - L_{sec}}{2} - 1}.$$

Simulation results

In this section, we provide a concrete example to illustrate the advantages of our scheme with real numerical values.

The polar coding module

The following analysis and explanation illustrate how the polar coding module combines channel entropy and computational entropy to ensure that Eve's BER is sufficiently high to meet the average min-entropy requirement of the secure coding module.

BER-influence model

We consider more realistic scenarios of imperfect polarization. Taking the BSC with crossover probability p as an example, we propose a BER-influence model after the key is inserted into the frozen bits of polar codes, which is mainly divided into four aspects: decoding strategy, positions of key bits in the frozen bits, the number of key bits in the frozen bits and key reuse in the frozen bits. It's important to note that Bob possesses knowledge of the key's value, whereas Eve does not. Hence, the key referred to in this subsection pertains to the secret key.

Decoding Strategy When the code length $N = 1024$ and the message length $K = 512$, the BER performance after Eve and Bob adopt different SCL decoding strategies with the key in the frozen bits is shown in Fig. 7.

For Eve, consider two decoding strategies, SCL-Eve-key-allzero means to decode key bits in the frozen bits as an all-zero string. SCL-Eve-key-allmsg represents that key bits in the frozen bits are decoded as random messages. It can be seen that the BER of the second decoding strategy is smaller, that is, Eve's attack capability is stronger, and Eve will adopt this decoding strategy.

For Bob, since Bob knows the value of the key in the frozen bits, the key has no effect on the decoding message strategy. Therefore, Bob can use the normal SCL decoding algorithm with the correct key in the frozen bits.

Positions of Key Bits in the Frozen Bits When the code length $N = 1024$ and the message length $K = 512$, the BER performance of Eve and Bob after inserting key bits into different positions of frozen bits is shown in Fig. 8.

For Eve, the BER is smaller when key bits are inserted into frozen bit channels with "good reliability" than when they are inserted into frozen bit channels with bad reliability. It should be note that "good reliability" refers to relatively good channels among frozen bit channels with poor reliability, the same below. It shows that, for higher security, the key bits can be placed in less reliable frozen bit channels. But considering the maximum attack capability of Eve, it can be assumed that the key bits are placed in positions where the reliability of the frozen bit channels are "good".

For Bob, the positions of key bits in the frozen bits have no effect on BER, indicating that the positions of key bits in the frozen bits do not affect its decoding performance.

The Number of Key Bits in the Frozen Bits When the code length $N = 1024$ and the message length $K = 512$, the BER performance of Eve and Bob after inserting different numbers of key bits into the frozen bits is shown in Fig. 9.

For Eve, the BER increases with the number of key bits in the frozen bits. This suggests that for higher security, more key bits should be placed in the frozen bits.

For Bob, the number of key bits in the frozen bits has no effect on BER, indicating that the number of key bits in the frozen bits does not affect its decoding performance.

Key Reuse in the Frozen Bits Messages $\boldsymbol{m}$ and $\tilde{\boldsymbol{m}}$ are encoded to obtain codewords $\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G}_A + \boldsymbol{k}\boldsymbol{G}_{A^c}$ and $\tilde{\boldsymbol{c}} = \tilde{\boldsymbol{m}}\boldsymbol{G}_A + \tilde{\boldsymbol{k}}\boldsymbol{G}_{A^c}$, respectively. If $\boldsymbol{k} = \tilde{\boldsymbol{k}}$, Eve can get

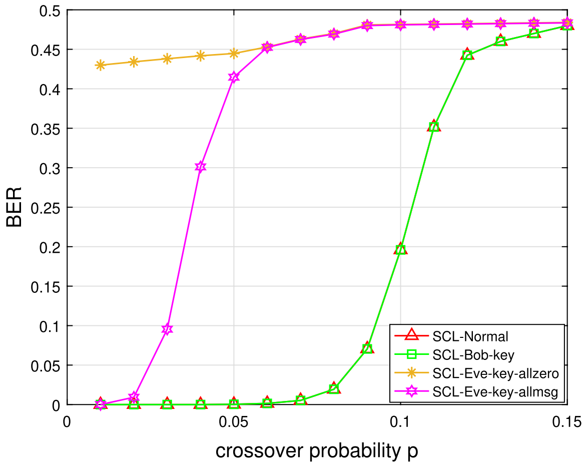

Fig. 7 The BER performance after Eve and Bob adopt different SCL decoding strategies with the key in the frozen bits

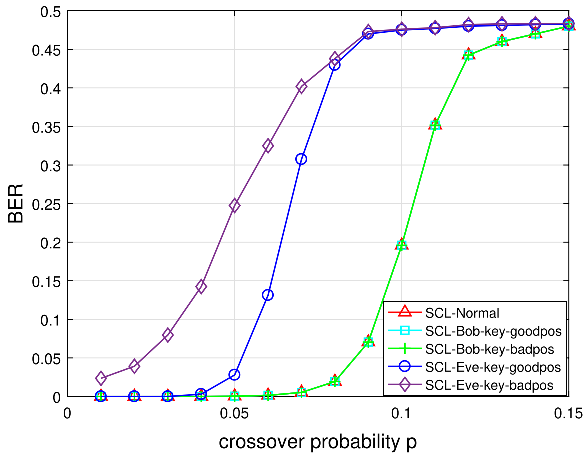

Fig. 8 The BER performance of Eve and Bob after inserting key bits into different positions of frozen bits

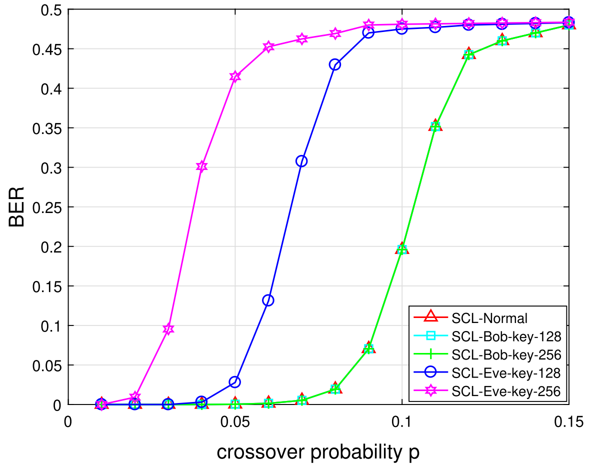

Fig. 9 The BER performance of Eve and Bob after inserting different numbers of key bits into the frozen bits

some information about the message through simple algebraic operations. Note that for security, the key in the frozen bits cannot be reused and must be time-varying.

Combination of reciprocity, noise, and computational entropy

To ensure correct message recovery by Bob, the polar code must effectively correct errors in Bob's channel. As shown in Fig. 9, when the crossover probability of Bob's channel is set to $p_B = 0.05$, a polar code with parameters $N = 1024$ and $K = 512$ demonstrates perfect error correction capabilities.

Assuming that Eve's channel noise is the same as Bob's, that is, $p_{E_0} = p_B = 0.05$. At this time, the noise entropy falls within the error correction capability of the polar code, so it is necessary to insert a key into the frozen bits to increase Eve's BER for security. As shown in Fig. 9, when a 128-bit key is inserted, Eve's BER is about 0.03, corresponding to the computational average min-entropy $\widetilde{H}^c_\infty = 22$ bits; when a 256-bit key is inserted, Eve's BER increases to about 0.41, corresponding to $\widetilde{H}^c_\infty = 389$ bits. This shows that under this parameter configuration, Eve, by expending a computational complexity of 2^{128} to perform an exhaustive search on 128-bit key, reduces the computational average min-entropy by 367 bits. In this scenario, the noise entropy is completely error-corrected by the polar code, while the reciprocity entropy and computational entropy contribute to determining the BER.

Assuming that Eve's channel noise is larger than Bob's, that is, $p_{E_0} > p_B = 0.05$. As shown in Fig. 9, when $p_{E_0} = 0.09$, Eve's BER is about 0.07. At this time, a portion of the noise entropy is error-corrected by the polar code, while another part contributes to Eve's BER. Moreover, the greater the value of p_{E_0}, the larger the BER contributed by the noise entropy. After inserting the key into the frozen bits, Eve's BER will be larger. When a 128-bit key is inserted, Eve's BER is about 0.47, corresponding to $\widetilde{H}^c_\infty = 468$ bits; when a 256-bit key is inserted, Eve's BER increases to about 0.48, corresponding to $\widetilde{H}^c_\infty = 483$ bits. This shows that under this parameter configuration, Eve, by expending a computational complexity of 2^{128} to perform an exhaustive search on 128-bit key, reduces the computational average min-entropy by 15 bits. In this scenario, a portion of the noise entropy is error-corrected by the polar code, while the remaining noise entropy, reciprocity entropy, and computational entropy collectively contribute to the BER.

Assuming that Eve's channel noise is smaller than Bob's, that is, $p_{E_0} < p_B = 0.05$. As shown in Fig. 9, when $p_{E_0} = 0.03$, the noise entropy is completely error-corrected by the polar code. Thus, a key needs to be inserted into the frozen bits to increase Eve's BER. A 128-bit key is insufficient to increase Eve's BER as it remains within the error correction capability of the polar code. Therefore, more key bits need to be inserted. When a 256-bit key is inserted, Eve's BER increases to about 0.1, corresponding to $\widetilde{H}^c_\infty = 77$ bits. This shows that under this parameter configuration, Eve, by expending a computational complexity of 2^{128} to perform an exhaustive search on 128-bit key, reduces the computational average min-entropy by 77 bits. In this scenario, all of the noise entropy and a portion of the reciprocity entropy are error-corrected by the polar code, while the remaining reciprocity entropy and computational entropy contribute to the BER.

Message rate of the scheme

In this subsection, we analyze the message rate of our scheme ($\mathcal{E}$), the joint scheme of key generation and one time pad (KG), the modular semantically-secure scheme based on the wire-tap channel model (WC), and the simple channel entropy combination scheme (SC), denoted as $Rate_{\mathcal{E}}$, $Rate_{KG}$, $Rate_{WC}$, and $Rate_{SC}$, respectively.

Set p_B to 0.05, p_{E_0} to 0.09. As shown in Fig. 9, a polar code with $N = 1024$ and $K = 512$ perfectly corrects errors in Bob's channel and reduces Eve's BER to $p_{E_{WC}} = 0.07$.

In our scheme, for simplicity, we insert a key of length $L_k = 512$ bits with an entropy of $H_0 = 208$ bits. Consequently, the length of the key's syndrome is calculated as $L_{syn} = \lceil L_k(1/C_{0.09} - 1)\rceil = 397$ bits, and the number of syndrome bits that need to be transmitted securely is $L_{syn_s} = H_0 - (L_k - L_{syn}) = 93$ bits. We set the security parameter α of our scheme to 2^{-80}, which means that the entropy of the key obtained by Eve in the exhaustive search attack is $H_{sk} = 80$ bits. Consequently, the remaining entropy of the key, the part that Eve doesn't know, is

$H_0 - H_{sk} = 208 - 80 = 128$ bits. In this scenario, simulation results indicate that the *PE* process can increase Eve's BER to $p_E = 0.47$. Subsequently, we can calculate that $L_{sec} = \lfloor -K\log(1-p_E)\rfloor - 2\log(1/\alpha) + 2 = 310$ bits. Therefore, we set the size of the message packet to $B = L_{sec} - L_{syn_s} = 217$ bits. Given these parameters, the message rate of our scheme is

$$Rate_{\mathcal{E}} = \frac{B}{N + (L_{syn} - L_{syn_s})} = 0.1634.$$

Next, we analyze the message rate of the joint scheme of key generation and one time pad when the message packet size is $B = 217$ bits. To ensure correct and secure transmission, Alice needs to send the key's syndrome used in the information reconciliation and the error-correcting codeword of the one-time pad result. We denote the number of bits quantized by Alice and Bob as $L_{k_{KG}}$. Therefore, the length of the key's syndrome that Alice needs to send during information reconciliation can be calculated as $L_{syn_{KG}} = \lceil L_{k_{KG}}(1/C_{0.09} - 1)\rceil$, which is also the amount of information leakage. In addition, to obtain the uniformly random key, $2\log(1/\alpha) - 2$ bits of key will be reduced in the process of privacy amplification. That is to say, $L_{k_{KG}} - L_{syn_{KG}} - 2\log(1/\alpha) + 2$ is no less than 217. From this, the minimum required value for $L_{k_{KG}}$ is 1664 bits, and further $L_{syn_{KG}} = 1289$ bits. We assume that the error-correcting code used to ensure the correct transmission of the message is optimal and its rate can reach the capacity $C_{0.05}$ of legitimate channel in our parameter setting. To sum up, the message rate of this scheme is

$$Rate_{KG} = \frac{B}{L_{syn_{KG}} + \lceil B/C_{0.05}\rceil} = 0.1361.$$

Then, we analyze the secure transmission scheme based on the wire-tap channel model. For the modular semantically-secure scheme, taking the RItE scheme as an example, to protect a message packet with a length of $B = 217$ bits, the average min-entropy of the secure codeword $\boldsymbol{x}$ from Eve's perspective needs to be at least $h = B + 2\log(1/\alpha) - 2 = 375$ bits. Correspondingly, the length of $\boldsymbol{x}$ is $L_{k_{WC}} = \lceil -h/(\log(1-p_{E_{WC}}))\rceil = 3582$ bits. From the above, the message rate of the RItE scheme is

$$Rate_{WC} = \frac{B}{L_{k_{WC}}/C_{0.05}} = 0.0432.$$

For the polar code-based secure transmission scheme, in contrast to the modular semantically-secure scheme, there is no entropy loss introduced by the extractor. If the (1024, 512)-polar code is perfectly polarized, the secrecy capacity $C_s = C_{0.05} - C_{0.09}$, allowing for the secure transmission of $\lfloor 1024 * C_s\rfloor = 153$ bits. However, experimental results indicate that polarization is not perfect, and the available entropy is only $\lfloor -K\log(1-p_{E_{WC}})\rfloor = 53$ bits, which is insufficient for transmitting $B = 217$ bits securely.

Finally, we analyze the simple channel entropy combination scheme. When transmitting a message of size $B = 217$ bits, a reciprocity entropy of $217 - 53 = 164$ bits is required. Similar to the key generation scheme, the necessary quantization bits amount to $L_{k_{SC}} = 1429$ bits, and the length of information reconciliation syndrome is $L_{syn_{SC}} = 1107$ bits. Additionally, to reduce network traffic, the syndrome required for the next message packet can be transmitted within this message packet, with a length of $L_{syn_{SCs}} = 512 - 217 = 295$ bits. Then, the message rate is

$$Rate_{SC} = \frac{B}{N + (L_{syn_{SC}} - L_{syn_{SCs}})} = 0.1182.$$

It can be observed that due to imperfect polarization, the message rate is lower than that of the joint scheme of key generation and one time pad.

In summary, when $p_B = 0.05$, $p_{E_0} = 0.09$, $N = 1024$, $K = 512$, $p_{E_{WC}} = 0.07$, $\alpha = 2^{-80}$, and $B = 217$, the efficiency comparison of the aforementioned four schemes are shown in Table 1.

The message rate of our scheme is approximately 1.2 times that of the joint scheme of key generation and one time pad, 3.8 times that of the RItE scheme based on the wire-tap channel model, and 1.4 times that of the simple channel entropy combination scheme. Under our parameter settings, the noise gap between the eavesdropping channel and the main channel is small. In the wire-tap channel model, security is achieved by accumulating the differences in uncertainty between Bob and Eve regarding the transmitted codeword bits in the channel. Therefore, when the noise gap is small, longer codewords are required to accumulate sufficient entropy to protect the message. In the key generation model, Alice and Bob have previously obtained keys with sufficient randomness and approximate consistency. The subsequent operations require the transmission of syndrome bits and privacy amplification to ensure both parties share a consistent and secure key. In scenarios with small noise gaps, the

Table 1 Efficiency comparison of the four schemes

Parameters **schemes**	L_k **(bits)**	L_{syn} **(bits)**	L_{syn_s} **(bits)**	**Rate**
$\mathcal{E}$	512	397	93	0.1634
KG	1664	1289	–	0.1361
WC	3582	–	–	0.0432
SC	1429	1107	295	0.1182

communication overhead of the key generation model is lower than that of the wire-tap channel model. Therefore, the RItE scheme based on the wire-tap channel mode (using only noise entropy) is less efficient compared to the the joint scheme of key generation and one time pad (using only reciprocity entropy) and the simple channel entropy combination scheme (utilizing both noise and reciprocity entropy). Moreover, it is also less efficient than our proposed scheme, which combines noise entropy, reciprocity entropy, and computational entropy.

Experimental testing

In this section, we implemented a simple system based on the universal software radio peripheral (USRP) in the laboratory environment to test our scheme.

Experimental environment

The experimental environment mainly includes software and hardware configurations, as well as the connection mode.

Software and hardware configurations

Software Configuration

- Ubuntu 18.04: software environment.
- UHD v3.15.0.0: USRP device drivers.
- Python 3.6.8: communication flow and algorithm implementation.
- Django 4.2.7: used for demonstration interface design.

Hardware Configuration

- USRP B210: three units.
- CDA-2990: one unit, for clock synchronization of the three USRP devices.
- SMA male-to-male coaxial cables: six pieces, used to connect the three USRP devices to the CDA-2990 clock source.
- ThinkPad T14 (CPU: AMD Ryzen 7 PRO 4750U with Radeon Graphics @ 1.70 GHz * 8; Memory: 16 GB; Storage: 512 GB): 4 units, used for processing USRP data.

Connection mode

The device connection mode is illustrated in Fig. 10. Three ThinkPad terminals are individually connected to three USRP B210 devices via USB3.0. Each USRP is then connected to the CDA-2990 clock source using two SMA male-to-male coaxial cables. The three ThinkPad terminals are linked to the demonstration terminal through a local area network (LAN). At the demonstration terminal, we control the behavior of the three ThinkPad terminals and showcase the results through a demonstration interface.

Experimental results

The secure transmission system consists of three stages. In the first stage, Alice and Bob extract nearly consistent keys from the channel. In the second stage, a consistent key is negotiated through channel coding. In the third stage, data transmission takes place. To ensure the reliability of data transmission, automatic retransmission and CRC error-checking mechanisms are introduced in the second and third stages. The actual communication effectiveness is demonstrated by transmitting images.

In the laboratory communication environment, the conditions assumed in Sect. 5.2 are all satisfied: the post-quantization key inconsistency rate is lower than the assumed 0.09; the noise in the legitimate channel is lower than the assumed 0.05; and the noise in the eavesdropping channel is higher than the assumed 0.09. Therefore, following our parameter configuration ($B = 217$, $L_{syn} = 397$, $L_{syn_s} = 93$, $K = 512$, $N = 1024$), Alice and Bob can correctly transmit the image, while Eve cannot obtain any information. This demonstrates that in the communication environment we established, the actual message rate of the scheme can achieve the theoretical results in Sect. 5.2. The demonstration of secure transmission is illustrated in Fig. 11.

Discussion

Our paper also has the following limitations that warrant further investigation:

(1) The reliability of the analysis of Eve's BER depends on the reliability of the assumption regarding Eve's best attack strategy. In this paper, we assume that Eve's best attack strategy is a hybrid approach com-

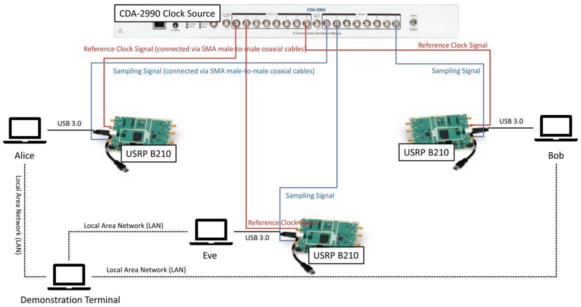

Fig. 10 The device connection mode

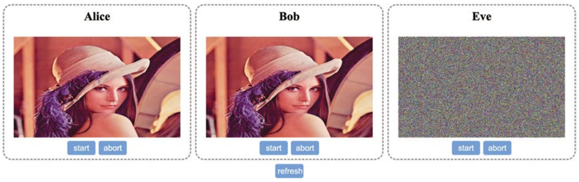

Fig. 11 Security transmission demonstration

bining exhaustive search with the state-of-the-art SCL decoding algorithm, which introduces computational entropy. Evaluating whether there is a more efficient attack algorithm for Eve and assessing the computational complexity are the issues we will consider next.

(2) Key generation relies on the reciprocity of the channel (ensuring key consistency), time variability (ensuring key randomness), and spatial decorrelation (ensuring uncertainty of the key to Eve). We have adopted the conclusions from Zhang et al. (2016), asserting that when Eve is located more than one half-wavelength away from either user, the eavesdropping channel is considered uncorrelated with the legitimate channel. However, we did not consider the scenario where Eve is within one half-wavelength. More accurately modeling Eve and analyzing Eve's attack capabilities is crucial.

(3) Currently, computationally secure coding is not yet practical Ishai et al. (2022, 2023). Therefore, our scheme utilizes information-theoretically secure coding to exploit entropy, which inherently incurs entropy loss. In future work, we plan to design practical computationally secure coding to enhance the utilization of entropy, thereby further improving the efficiency of the scheme.

Conclusion

In this paper, we propose a computational secure transmission framework based on polar codes by establishing a connection between BER and security parameters. The main idea is that polar codes not only realize the function of error correction coding, but also increase the eavesdropperś BER by combining channel entropy and computational entropy to meet the average min-entropy requirement of secure coding, thereby realizing the combination of secure coding and error correction coding. Furthermore, we introduce a BER-influence model after inserting the key into the frozen bits of polar codes. Through experimental simulations, we derive the BER-influence curve, where the number and positions of secret key bits serve as independent variables. Additionally, we propose a compact information reconciliation method that leverages secure transmission to minimize the information leakage. Compared with the joint scheme of physical layer key generation and one time pad, the modular semantically-secure scheme based on the wire-tap channel model, and the simple channel entropy combination scheme, the message rate of our scheme is about 1.2, 3.8 and 1.4 times better under concrete parameter settings. We validate the feasibility of our scheme through experimental testing.

Acknowledgments

We would like to thankthe anonymous reviewers and editors for detailedcomments and useful feedback.

Author contributions

AC and HMJ completed the main work of the paper and drafted the manuscript. LXH and BL participated in problem discussions and improvements of the manuscript. LWJ conducted experimental simulations. All authors read and approved the final manuscript.

Funding

This work is supported in part by the National Key R &D Program of China under Grant 2020YFB1807502 and in part by the CAS Project for Young Scientists in Basic Research under Grant YSBR-035.

Availability of data and materials

Not applicable.

Code availability

Not applicable.

Declarations

Ethical approval and Consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 6 November 2023 Accepted: 5 March 2024
Published online: 04 May 2024

References

An C, Liu Y, Lu X (2021) Evolution of the polar code-based encryption schemes. In: 2021 IEEE Globecom workshops (GC Wkshps), pp. 1–6. IEEE

Arikan E (2009) Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans Inf Theory 55(7):3051–3073

Balatsoukas-Stimming A, Parizi MB, Burg A (2015) LLR-based successive cancellation list decoding of polar codes. IEEE Trans Signal Process 63(19):5165–5179

Bellare M, Tessaro S (2012) Polynomial-time, semantically-secure encryption achieving the secrecy capacity. arXiv preprint arXiv:1201.3160

Bellare M, Tessaro S, Vardy A (2012) A cryptographic treatment of the wiretap channel. arXiv preprint arXiv:1201.2205

Bellare M, Tessaro S, Vardy A (2012) Semantic security for the wiretap channel. In: Annual cryptology conference, pp. 294–311. Springer

Bennett CH, Brassard G, Crépeau C, Maurer UM (1995) Generalized privacy amplification. IEEE Trans Inf Theory 41(6):1915–1923

Bloch M, Barros J, Rodrigues MR, McLaughlin SW (2008) Wireless information-theoretic security. IEEE Trans Inf Theory 54(6):2515–2534

Brassard G, Salvail L (1993) Secret-key reconciliation by public discussion. In: Workshop on the theory and application of of cryptographic techniques, pp. 410–423. Springer

Csiszár I, Korner J (1978) Broadcast channels with confidential messages. IEEE Trans Inf Theory 24(3):339–348

Guyue L, Aiqun H, Le S (2014) Secret key extraction in wireless channel. J Cryptol Res 1(3):211–224

Hershey JE, Hassan AA, Yarlagadda R (1995) Unconventional cryptographic keying variable management. IEEE Trans Commun 43(1):3–6

Hong Z (2020) A study on channel key generation based on physical layer. Master's thesis, Fujian Normal University

Hu A, Li G (2014) Physical layer security in wireless communication: survey. J Data Acquis Process 29(3):341–350

Ishai Y, Korb A, Lou P, Sahai A (2022) Beyond the csiszár-korner bound: best-possible wiretap coding via obfuscation. In: Annual international cryptology conference, pp. 573–602

Ishai Y, Jain A, Lou P, Sahai A, Zhandry M (2023) Computational wiretap coding from indistinguishability obfuscation. In: Annual international cryptology conference, pp. 263–293. Springer

Kim Y-S, Kim J-H, Kim S-H (2014) A secure information transmission scheme with a secret key based on polar coding. IEEE Commun Lett 18(6):937–940

Liu Y, Chen H-H, Wang L (2016) Physical layer security for next generation wireless networks: theories, technologies, and challenges. IEEE Commun Surv Tutor 19(1):347–376

Li G, Zhang Z, Zhang J, Hu A (2020) Encrypting wireless communications on the fly using one-time pad and key generation. IEEE Internet Things J 8(1):357–369

Lu X, Lei J, Li W, Lai K, Pan Z (2018) Physical layer encryption algorithm based on polar codes and chaotic sequences. IEEE Access 7:4380–4390

Lu X, Li W, Lei J, Shi Y (2019) A physical layer encryption algorithm based on partial frozen bits of polar codes and AES encrypter. In: 2019 9th international conference on information science and technology (ICIST), pp. 193–198. IEEE

Mathur S, Trappe W, Mandayam N, Ye C, Reznik A (2008) Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In: Proceedings of the 14th ACM international conference on mobile computing and networking, pp. 128–139

Maurer UM (1993) Secret key agreement by public discussion from common information. IEEE Trans Inf Theory 39(3):733–742

McEliece RJ (1978) A public-key cryptosystem based on algebraic. Coding Thv 4244:114–116

Nasrabadi NM, King RA (1988) Image coding using vector quantization: a review. IEEE Trans Commun 36(8):957–971

Sanenga A, Mapunda GA, Jacob TML, Marata L, Basutli B, Chuma JM (2020) An overview of key technologies in physical layer security. Entropy 22(11):1261

Sason I, Verdú S (2017) Arimoto-rényi conditional entropy and Bayesian hypothesis testing. In: 2017 ieee international symposium on information theory (ISIT), pp. 2965–2969. IEEE

Sharifian S, Lin F, Safavi-Naini R (2017) Hash-then-encode: a modular semantically secure wiretap code. In: International worskhop on communication security, pp. 49–63. Springer

Tal I, Vardy A (2015) List decoding of polar codes. IEEE Trans Inf Theory 61(5):2213–2226

Wu Y, Khisti A, Xiao C, Caire G, Wong K-K, Gao X (2018) A survey of physical layer security techniques for 5g wireless networks and challenges ahead. IEEE J Sel Areas Commun 36(4):679–695

Wyner AD (1975) The wire-tap channel. Bell Syst Tech J 54(8):1355–1387

Ye C, Mathur S, Reznik A, Shah Y, Trappe W, Mandayam NB (2010) Information-theoretically secret key generation for fading wireless channels. IEEE Trans Inf Forensics Secur 5(2):240–254

Zhang J, Duong TQ, Marshall A, Woods R (2016) Key generation from wireless channels: a review. IEEE Access 4:614–626

Zhang J, Woods R, Duong TQ, Marshall A, Ding Y (2016) Experimental study on channel reciprocity in wireless key generation. In: 2016 IEEE 17th international workshop on signal processing advances in wireless communications (SPAWC), pp. 1–5. IEEE

Zhihua Y (2016) The analysis and improvement of cascade. Master's thesis, Southwest University

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.