Secret Key Generation from Channel Noise with the Help of a Common Key

Tatsuya TOMARU†

SUMMARY Information-theoretically secure communications are possible when channel noise is usable and when the channel has an intrinsic characteristic that a legitimate receiver (Bob) can use the noise more advantageously than an eavesdropper (Eve). This report deals with the case in which the channel does not have such an intrinsic characteristic. Here, we use a pre-shared common key as a tool that extrinsically makes Bob more advantageous than Eve. This method uses error-correcting code in addition to the common key and noise, and manages the three components in random-number transmission. Secret keys are generated from noise, and messages are encrypted with the secret keys in a one-time pad manner. As a result, information leaks meaningful to Eve are restricted to the parity-check symbols for the random numbers. It is possible to derive the candidates of the common key from the parity check symbols, and the security of this method is quantified in terms of the amount of computations needed for an exhaustive search of the candidates, where we evaluate the security by assuming that all parity check symbols leak to Eve without bit errors. Noise contributes to not only generating secret keys but also enhancing the security because the candidates of the common key increase with it. 

key words: Channel, noise, error-correcting code, common key, secret key, bit error

1. Introduction

Highly confidential information, e.g., government and military secrets, must be communicated with maximum security between a limited number of parties. This kind of information might affect national fortunes 50 or even 100 years hence, and therefore, its security must be long term. This report proposes a method that meets this requirement. It discusses security under the following three conditions: (1) only technologies available at present can be used; (2) communication is world-wide; (3) only a limited number of parties communicate with each other.

The method of Wyner [1] using channel noise is one way of maximizing security, and it achieves information-theoretic security. When the bit-error rate (BER) of an eavesdropper (Eve) is higher than that of a legitimate receiver (Bob), this difference generates a secrecy capacity [1,2]. Wyner assumed that Bob receives signals at a lower bit-error rate than Eve, but this assumption was later loosened. That is, so long as some of Eve’s received errors are different from Bob’s, Bob’s receiving conditions don’t have to be better than Eve’s, and the amount of information corresponding to Eve’s unique errors can be transformed into secret keys through public discussions between the sender (Alice) and Bob1 [3,4]. The method using channel noise has since been widely studied, and the researches can classified into ones on channel-type models and ones on source-type models [4]. The former model is one in which Alice and Bob share randomness from the channel noise. It requires bit errors that only Eve suffers from in order to generate secret keys from the noise [5]. A broadcast channel is usually assumed to conform to this situation [2,4]. The latter model is one in which both Alice and Bob receive randomness from a source, and when the randomness is correlated between Alice and Bob, independently of Eve, secret keys are generated [6]. Approaches that take both models into consideration have also been studied [7–9].

The noisy channel method can achieve information-theoretic security. Another method that does so is quantum cryptography [10–12]. Quantum cryptography similarly possesses the concept of channel-type and source-type models. The BB84 protocol, wherein single photons are transmitted and received, corresponds to a channel-type model [10]. Quantum entanglement-related methods correspond to source-type models [13,14].

While methods having information-theoretic security are achievable in principle, they are difficult to apply to long-haul optical fiber transmissions. Optical fibers cause transmission losses of 0.2 dB/km, and as a result, quantum cryptography using single photons is limited to about 100 km. The noisy channel method faces another difficulty. Eavesdropping is easy in fiber communications if there is a preinstalled photo-coupler that divides the light. In particular, if the photo-coupler is near the transmitter, Eve can receive signals without being affected by channel noise. In this case, it cannot be

†The author is with Center for Exploratory Research, Research & Development Group, Hitachi, Ltd., Hatoyama, Saitama, 350-0395 Japan.

Email: tatsuya.tomaru.yq@hitachi.com

1 As a result of public discussions, Bob gains an advantage over Eve.
assumed that Eve will have unique errors, and thus, the noisy channel model cannot be applied.

Both the noisy channel method and quantum cryptography have an important mechanism wherein Bob can become more advantageous than Eve; the noisy channel method uses a broadcast channel or correlated randomness for this, and quantum cryptography uses a quantum-mechanics-based characteristic that measurement changes the quantum state. How these characteristics can be used has been the subject of extensive discussion. However, long-haul fiber transmissions do not possess such characteristics; another mechanism is needed in this case. We will turn our attention to the fact that the number of the communication parties is limited. For this specific case, we can assume a system wherein Alice and Bob share a common key consisting of random numbers with a uniform distribution (true random numbers) with a fixed length beforehand. The common key could be securely passed by hand, for example. Generally, the common key-sharing method is unsupervised as long as the security is higher than that discussed in this report. The common key gives Bob an advantage over Eve. Here, the common key is not used as a seed key, but is instead used as a tool for transforming the entropy of noise into that of secret keys. For this reason, the information of the common key is not reflected in the information on the transmission channel, and thus, the common key can be repeatedly used. The phase noise of a laser diode’s (LD) output, for example, can be used as a noise source (See section 7.4). Phase noise is always present in LD output, and it is sufficiently random [15].

There is a method called the \textit{aeq} (Y00) protocol that uses channel noise and a common key [16,17]. This method is basically a stream cipher with quantum fluctuations that uses a common key as a seed key, and it uses multiple bases. However, the method in this report is not a stream cipher, but secret-key generation. Messages are encrypted with the secret keys by one-time pad. A common key is used only inside the transmitter and receiver, and the information on the transmission channel does not reflect the common key-related information. The two methods hence belong to different concepts.

The method in this report is not resistant to brute force attacks because a common key is used. However, the secret keys are continuously generated from noise, and the messages are encrypted by using a one-time pad. Thus, information leaks are restricted in the random-number transmission stage for key agreement. To evaluate the security, Eve is assumed to exactly obtain parity check symbols of an algebraic error-correcting code that is used to transmit the random numbers. Even if the assumption is advantageous to Eve, she must decode the error-correcting code, which is a block code. To do this, she must list the candidates of information symbols by using parity check symbols and list the candidates of the common key. In other words, Eve has no other decrypting method that is more efficient than listing the candidates of information symbols. In addition, the number of candidates increases because of the existence of bit errors, and thus, security is strengthened even more. Computational security is generally achieved by relying on some sort of mathematical difficulty. For example, the security of Diffie-Hellman key agreement is founded on the existence of a difficult computation in number theory [18]. There is no assurance that the difficulty will never be overcome. An efficient algorithm for overcoming that difficulty might be found. However, our method does not assume any mathematical difficulties, and therefore, there is no threat that an efficient decrypting algorithm might be found. The method requires an exhaustive search for the candidates of the common key in decryption. The security of our method does not reach the level of information theoretic security, but it falls into some range of computational security. However, thanks to there being no salient threat, we do not need to be anxious about any unexpected decryptions. Our method will be useful for protecting highly confidential information like government and military secrets.

Cryptography generally has a trade-off between security and convenience. Methods with information-theoretic security have high security but their message transmission rate \( R_m \), defined by \( R_m = n_m/n_{all} \), where \( n_{all} \) is the total number of transmitted bits and \( n_m \) is the message part, is low (\( R_m << 1 \)), and long-haul transmissions using them are generally difficult. In contrast, methods with computational security achieve \( R_m \approx 1 \), but generally face the threat that an efficient decrypting method might be found. Supposing we interpret these two kinds of methods as being at opposite ends of a trade-off, our method is located in the middle, because it achieves computational security that removes the threat. However, in so doing, the message transmission rate is reduced to \( R_m \approx 1 \).

2. Framework

This report concerns key agreement consisting of random-number transmission and secret key generation using the transmitted random numbers. Messages are transmitted with a one-time-pad using the generated secret keys. This section describes the framework of the key agreement and defines the security of the method. The notation is such that when a character style like \( X, Y, Z \) designates sets, the corresponding random variables are described with capital letters, like \( X, Y, Z \), and corresponding elements are described with small letters, like \( x, y, z \). Bold letters like \( x, y, z \) designate row vectors of \( x, y, z \). Letters like \( X^n \) designate successive \( n \) letters.

Let us assume that there is noise in the transmission channel used for the key agreement.
Therefore, there are generally bit errors in the signals received by Bob and Eve. Let $s_{AB}$ be the secret keys to be shared between Alice and Bob. Generally in a key agreement protocol using a noisy channel, Alice encodes $s_{AB}$ such that $E_{g}: \{0, 1\}^{n} \rightarrow \{0, 1\}^{n}$, $s_{AB} \mapsto x$ and sends them to Bob; he receives and decodes them such that $D_{g}: \{0, 1\}^{n} \rightarrow \{0, 1\}^{n}$, $y \mapsto s_{AB}$. Because of bit errors, generally $x \neq y$. The mapping $E_{g}$ has two purposes: one is to make the information leaking to Eve meaningless; the other is to achieve accurate communications. The former purpose requires $n < n$, and thus $E_{g}$ is probabilistically performed. However, even if $E_{g}$ itself is simple, its inverse, i.e., $D_{g}$, is not easy. Therefore, the probabilistic encoding $E_{g}$ is not preferable for real systems. Deterministic encoding is better. For this reason, let us invert the process on Alice’s side such that $E_{g}^{-1}: \{0, 1\}^{n} \rightarrow \{0, 1\}^{n}$, $x \mapsto s_{AB}$ [19]. In this case, all processes of the method can be made deterministic and thereby practical. Now, we divide $E_{g}^{-1} (E_{g})$ into two stages, i.e., the encoding $E$ and the secret key generation $S$, as in Definition 1 below. Figure 1 shows the framework discussed in this report.

As mentioned above, noisy channel models generally have an intrinsic characteristic to make Eve disadvantageous. For example, the broadcast model assumes that Eve suffers from bit errors independent of Bob’s ones. However, we do not assume such an intrinsic characteristic. Instead, we assume a common key $k_{e}$ that Alice and Bob share beforehand to make Eve disadvantageous. The following Definition 1 defines the key agreement protocol discussed in this report.

**Definition 1 [Key agreement protocol]:** Let us assume that there is noise in the transmission channel. Alice and Bob share beforehand a common key $k_{e} \in \{0, 1\}^{n}$ consisting of a random $N_{k}$-bit string with uniform distribution over $\{0, 1\}^{n}$. Alice encodes a random $n$-bit string $x \in \{0, 1\}^{n}$ with a uniform distribution over $\{0, 1\}^{n}$ by using an $(n, k)$ block code of block length $n$ and information length $k$ ($n > k$), where $x_{i} \in \{0, 1\}^{k}$ are information symbols and $x_{2} \in \{0, 1\}^{n-k}$ is redundant information of $x_{1}$; and she sends $x$ to Bob. Bob receives $y$ and $y_{2}$ and obtains $y_{1}$ that are error-corrected using $k_{e}$. Alice and Bob respectively generate secret keys $s_{AB}$ and $s_{AB} \in \{0, 1\}^{n}$ from $x_{1}$ and $y_{1}$. The encoding $E: \{0, 1\}^{n} \times k_{e} \mapsto (\{0, 1\}^{k}, \{0, 1\}^{n-k})$, $x \times k_{e} \mapsto x_{1} \times x_{2}$, and the decoding $D: \{0, 1\}^{n} \times k_{e} \mapsto (\{0, 1\}^{k}, y_{1})$, $y \times y_{2} \times k_{e} \mapsto y_{1}$, and the secret key generation $S: \{0, 1\}^{n} \mapsto (\{0, 1\}^{k}, x_{1} \mapsto s_{AB}$ and $y_{1} \mapsto s_{AB}$.

Encoding $E$ uses the common key $k_{e}$. Decoding $D$ requires $k_{e}$ and decoding is difficult without $k_{e}$. A concrete coding method is described in section 5.2. As long as Bob does not fail in decoding, $x_{1} = y_{1}$ and $s_{AB}$ = $s_{AB}$. In evaluating security, we assume that authentication has been executed and that Eve does not tamper with the channel by inserting or modifying messages. Moreover, we assume that Eve is an outsider.

Let $z, z_{1}$, and $z_{2}$ be Eve’s information corresponding to $x, x_{1}$, and $x_{2}$ ($y, y_{1}$, and $y_{2}$) for Alice (Bob). Eve’s final aim is to eavesdrop on messages. Because the secret key $s_{AB}$ is used with a one-time pad, Eve needs to derive $s_{AB}$ from $z$ and $z_{2}$ to achieve her aim. For simplicity, we assume that all $n$, bits of $s_{AB}$ are used in message transmissions. Let $z^{*}$ and $z_{2}^{*}$ be another pair of $z$ and $z_{2}$, and let $s_{AB}^{*}$ be the secret key generated from $z^{*}$ and $z_{2}^{*}$. If Eve uses a chosen-plaintext attack against the message transmissions, she can obtain any number of $s_{AB}^{*}$s. Here, to simplify the description, all the secret keys that Eve obtains will be represented by $s_{AB}^{*}$. In accordance with these premises, we assume that Eve’s attack and aim are as follows.

**Eve’s attack:** (1) Eve can passively obtain all of the information. (2) Eve can obtain any number of $s_{AB}^{*}$. (3) Eve cannot control the equipment inside the transmitter and receiver or the environmental noise.

**Attack aim:** Eve’s aim is to guess at least one bit of the secret key $s_{AB}$.

Because Definition 1 uses a common key, the security of Definition 1 is computational. Thus, we give the following definition.

**Definition 2 [Computational security in key agreement]:** Let us suppose a game in which Eve runs a probabilistic polynomial algorithm to try to guess one bit at any position of $s_{AB}$ from $z$ ($z^{*}$), $z_{2}$ ($z_{2}^{*}$), and $s_{AB}^{*}$ in one arbitrary trial. Let $s_{AB}$ be the guessed one-bit secret key. Let the probability of $s_{AB} = s_{AB}^{*}$ be $p_{n}$, which is called the probability of successfully guessing secret key. A key agreement method is called computationally secure if there exists $k_{0} \in N$ such that $p_{n} < 1/2 + 1/P(N_{k})$ is satisfied at a common key length $N_{k}$ > $k_{0}$ for every polynomial equation $P(N_{k})$.

The security of the framework of Definition 1 is based on the fact that decoding $D$ is difficult without $k_{e}$. Therefore, let us define the difficulty of decoding along the lines of Definition 2.

**Definition 3 [Computational security in encoding]:** Let us define an encoding $E_{k}$ by $x_{1} = E_{k}(x, k_{e})$ in accordance
with Definition 1. Let us suppose a game in which Eve runs a probabilistic polynomial algorithm to try to guess $x_1(x_1^*)$ from $z(z^*)$, $z_2(z_2^*)$, and $s_{AB}$ in one arbitrary trial. Let $z_1(z_1^*)$ be the guess of $x_1(x_1^*)$. Let the probability of $x_1 = z_1(x_1^* = z_1^*)$ be $p_d$, which is called the probability of successfully guessing information symbols. $E_a$ is called computationally secure if there exists $k_0 \in \mathbb{N}$ such that $p_d < 1/2^k + 1/P(N_k)$ is satisfied at a common key length $N_k \geq k_0$ for every polynomial equation $P(N_k)$.

The security of an encryption using a common key is computational; it has no resistance against a brute-force attack on the common key. However, if there is no efficient decrypting method except for a brute-force attack, sufficient security is still obtainable by choosing a sufficiently long key. The issue in computational security is not the lack of resistance against a brute-force attack but the threat that an efficient decrypting method might be found. Therefore, if it is assumed that there is no efficient decryption method in an encryption system, the system is sufficiently secure. Definition 2 corresponds to that assurance, but what is assumed in order to achieve the security of Definition 2 is important. If Definition 2 is achieved without assuming any mathematical difficulties such as that in the Diffie-Hellman key agreement, there is no threat that an efficient decryption algorithm might be found. The protocol of Definition 1 ensures the security of Definition 2 by using noise in addition to the common key $k$. It does not assume any mathematical difficulties. If the amount of information of secret keys generated in the protocol of Definition 1 is limited to the entropy of noise, the generated secret keys are fresh. Computational security in the sense of Definition 2 can be achieved using this freshness, as will be shown in Theorem 1.

3. Concrete method based on Definition 1

The code length $n$ and information symbol length $k$ are generally assumed to be sufficiently long in any key-agreement protocol using noise. However, a practical system might limit the range of $n$ and $k$. Hence, we introduce a parameter $u$ and use $uk$ symbols as a unit of the key generation to overcome the limit. The following Method 1 makes the framework of Definition 1 specific from the viewpoint of an actual system, including the introduction of the parameter $u$. (See Fig. 2).

**Method 1:** The noise-assisted key-agreement protocol based on Definition 1 consists of algorithms $(R_x, E_a, E_b, S)$ in the transmitter and algorithms $(F_x, D_a, D_b, S)$ in the receiver.

**Transmitter:**

(1) $x \xleftarrow{} R_x$

(2a) $E_a: \{0, 1\}^u \times \{k\} \rightarrow \{0, 1\}^k$, $x \times k_e \mapsto x_1$

(2b) $E_b: \{0, 1\}^k \rightarrow \{0, 1\}^{n-k}$, $x_1 \mapsto x_2$

(3) $S: \{0, 1\}^{n-k} \rightarrow \{0, 1\}^n$, $x_1 \mapsto s_{AB}$

**Receiver:**

(1') $y \leftarrow F_x(x)$

(2a') $D_a: \{0, 1\}^u \times \{k\} \rightarrow \{0, 1\}^k$, $y \times k_e \mapsto y_1$

(2b') $D_b: \{0, 1\}^k \times \{0, 1\}^{n-k} \rightarrow \{0, 1\}^n$, $y_1 \times y_2 \mapsto y_1$

(3') $S: \{0, 1\}^{n-k} \rightarrow \{0, 1\}^n$, $y_1 \mapsto s_{AB}$

Here, $x_1$ and $y_1$ are respectively $x_1$ and $y_1$ of $u$ blocks. $x_1$ is generated from $x$ with permutations in $E_a$, and $y_1$ is chosen to satisfy $n_{a}/m \leq C_{0u}$, where $C_{0u}$ is introduced in the next section. This choice makes $E_a$ computationally secure in the sense of Definition 3, as shown in Lemma 11, and makes Method 1 computationally secure in the sense of Definition 2, as shown in Theorem 1. In process (1), $x$ is output from a random-number generator $R_x$ in the transmitter and is transmitted to a receiver. In process (1'), $y$ is received by a receiver, where $y$ includes transmission-carrier noise and environmental noise in $E_x$ and $F_x$, respectively. Here, the symbol $\leftarrow$ is used to show that $x$ is a probabilistic output and $y$ contains a probabilistic bit error. We assume a memory-less binary symmetric channel (BSC) with a BER of $p_e$ as a model of the noise source $F_x$ in the transmitter. An example of system conforming to this model is optical-fiber communications. Light already has fluctuations (noise) that cause bit errors at the moment it is emitted from its source. Another example is noise added on purpose. Processes (2a) and (2b) are the encoding $E$ in Definition 1 and processes (2a') and (2b') are decoding $D$ in Definition 1. $x_1$ is generated from $x$ by using the common key $k_e$ in $E_a$, and $y_1$ is similarly generated from $y$ by using $k_e$ in $D_a$. A concrete example is described in section 5.2. $E_b$ and $D_b$ are respectively encoding and decoding to achieve errorless communications between Alice and Bob. Thus, $x_1 = y_1$ as long as Bob does not fail in decoding. Redundant information $x_2$ is transmitted through an errorless public channel, and thus, $x_2 = y_2$. Summarizing what has been covered so far, one sees that $x$ and $x_2$ are transmitted from the transmitter and $y$ and $y_2$ are received at the receiver. $x_1$ and $y_1$ are only used inside the transmitter and receiver and they are not transmitted. Processes (3) and (3') describe the secret key generation $S$ in Definition 1, which is achieved through privacy amplification [19,20] using universal hashing. $S$ is performed in units of $u$ blocks. If $x_1^u = y_1^u$, then $s_{AB} = s_{BA}$. Thus, $s_{AB}$ can be shared by Alice and Bob and used in encrypted communications of messages. Figure 2 summarizes the algorithms in the form of a block diagram.
Eve is assumed to be able to receive signals in the best condition; i.e., she receives $z \leftarrow F_3(x)$ for $x$ without environmental noise. Let $p_E (p_E \leq 1/2)$ be Eve’s bit-error rate, and let $p_B (p_B \leq 1/2)$ be Bob’s bit-error rate. Because Bob’s signals are affected by environmental noise $F_1$, generally $p_B \geq p_E$, where equality corresponds to the case of a noiseless channel.

Redundant information $x_2$ is openly transmitted through an errorless public channel in Method 1. This is because this setup makes the security analysis easy. When $x_2$ is transmitted through a channel with errors, the setup makes Eve disadvantageous. Thus, even if $x_2$ is transmitted through the same channel as $x$ is, the security assured for Method 1 is kept (see section 6.2).

![Fig. 2 Block diagram of Method 1.](image)

### 4. Conditional Secrecy Capacity

The framework of Definition 1 (Method 1) aims to achieve the security of Definition 2 by limiting the entropy of the generated secret keys to that of noise. The idea is the same as that of secrecy capacity [1-4]. However, the secrecy capacity is defined for cases in which secret keys are generated from noise without a common key. It is not defined for the case of using a common key. For this reason, we define the conditional secrecy capacity $C_s$ as a similar quantity. This quantity is defined under the assumption that the common key $k_e$ is secret, and the quantity expresses how many secret keys are generated from noise under the assumption, where “secret” means that Eve has no information about the common key.

The number of bit errors originating from noise statistically fluctuates. $n$ and $k$ should be sufficiently large to reduce the statistical fluctuations in each block. However, their range might be limited in actual systems. Therefore, we define $C_s$ that is applicable even to $n$ and $k$ of limited size by introducing the parameter $u$. The following definition 4 assumes that Alice and Bob share a common key $k_e$ and it is kept secret from Eve. The prerequisite is expressed as “$|k_e \leftarrow K_e$” in the following.

**Definition 4 (Conditional secrecy rate and capacity):** In Method 1, the common key $k_e$ is assumed to be kept secret from Eve, and the encoding $E$ is assumed to be computationally secure in the sense of Definition 3. Under the assumptions, if the following four conditions are satisfied for a given $\gamma > 0$, $R_s$ is called the conditional secrecy rate for a given $\gamma$. The maximum of $R_s$ is $C_s$ and is called the conditional secrecy capacity for the given $\gamma$.

1. $\Pr\{S_{kA}^n \neq S_{kA}^n [k_e \leftarrow K_e] < \gamma$
2. $\Pr\{R(S_{kA}^n [k_e \leftarrow K_e]; S_{kE}^n)/n_i < \gamma \} > (1 - \gamma)\{1 - 1/2^k - 1/P(N_k)\}$
3. $\log_2(S_{kA}^n)/n_i < H(S_{kA}^n [k_e \leftarrow K_e])/n_i + \gamma$
4. $\Pr\{H(S_{kA}^n [k_e \leftarrow K_e])/n_i > R_s - \gamma \} > 1 - \gamma$

Here, $S_{kA}^n$ are secret keys of $n$ bits generated from $Z^n$ and $Z^{*}_{kA}$ through Eve’s arbitrary guess.

Item (1) assures that Alice and Bob can communicate with each other with a sufficiently small error probability. Item (2) assures that the leaks of secret keys to Eve are sufficiently small. The factor $(1 - \gamma)$ in $\Pr\{\ldots\} > (1 - \gamma)\{1 - 1/2^k - 1/P(N_k)\}$ considers the rare case where the number of bit errors in every symbol is extremely small, owing to statistical fluctuations. The factor $(1 - 1/2^k - 1/P(N_k))$ reflects the assumption that the encoding $E_2$ is computationally secure in the sense of Definition 3. Item (3) assures the uniformity of $S_{kA}^n$. Item (4) indicates the condition that $R_s$ should satisfy in accordance with items (1)-(3). The description $\Pr\{\ldots\} > 1 - \gamma$ considers the rare case where the number of bit errors is extremely small, similar to Item (2).

Reference [19] describes the theory of privacy amplification as methods of generating the secret key $S$. Let Alice’s and Bob’s information be a random $n_A$-bit string with a uniform distribution over $\{0, 1\}^{n_A}$ and let Eve’s corresponding information be $n_E$-bits. Let any $n_i$ of $0 < n_i < n_A - n_e$ be a safety parameter, and let $n_i = n_A - n_e = n_e$. Theorem 3 and corollary 5 of Ref. [19] respectively give $H(S_{kA}^n) = n - 2^{n_i - H(x_{1A})}/\ln2$ and $I(S_{kA}^n; S_{kE}^n) \leq 2^{-n_i}/\ln2$ when Alice and Bob generate an $n_i$-bit string from an $n_A$-bit string by universal hashing [19]. Here, $R(X_{1A}^{n_A})$ is the Rényi entropy for the collisions in two independent trials, and it is given by $R(X_{1A}^{n_A}) = -\log_2 P(X_{1A}^{n_A})$ and $P(X_{1A}^{n_A}) = \sum_{x_{1A} \in \{0, 1\}^{n_A}} P(x_{1A})^2$ by letting $P(x_{1A})$ be the occurrence probability of $x_{1A} \in \{0, 1\}^{n_A}$; $S_{kE}^n$ is the result of Eve’s arbitrary guess. Eve’s information is $n_E$ bits, but it is not restricted to $n_E$-bit strings.

The claims of Theorem 3 and Corollary 5 in ref. [19] are applicable to Method 1 under the condition that the common key $k_e$ is secret. The parameters $n$ and $k$ in Method 1 are determined such that signals with a bit-error rate of $p_B$ are error-correctable. Let $t_e$ be the number of bit errors definitely correctable per block, which is the
lower limit of the maximum number of errors that can be corrected. Let \( t_m \) be the upper limit of the maximum number of errors that can be corrected per block, in which bit errors have the possibility of being corrected but the possibility is indefinite. The numbers \( t_e \) and \( t_m \) are characteristic parameters of the used code.

Definition 4 considers the statistical fluctuations of the bit errors. Now, let us define some quantities as preparation. Let the number of Eve’s bit errors per \( u \) blocks of information symbols be \( n_{ae} \), and let its average be \( \bar{n}_{ae} = ukp_e \) and standard deviation be \( \sigma_{n}\). Let \( R_{e} > 0 = \{ r \in R \mid r > 0 \} \).

**Lemma 1:** In Method 1, the common key \( k \) is assumed to be kept secret from Eve. Let the transmission channel be a memory-less binary symmetric channel (BSC). Let \( p_{ae} = p_e(\bar{n}_{ae} - r\sigma_{n})/\bar{n}_{ae} \) by using an \( r \in R_{e} > 0 \) that satisfies \( \Pr\{n_{ae} < \bar{n}_{ae} - r\sigma_{n}\} < \gamma \) for a small given \( \gamma > 0 \). Let \( n_s \) be the safe parameter in the secret key generation \( S \). If encoding \( E_s \) is computationally secure in the sense of Definition 3, the four conditions in Definition 4 can be satisfied by appropriately selecting the parameters \( n, k, u \), and \( n_s \) for the small given \( \gamma > 0 \). The conditional secrecy capacity for the given \( \gamma \) is \( C_s = (k - t_m)H(p_{ae}) - n_s/\ln u \) when using the binary entropy function \( H(p) = -\log_2 p - (1 - p)\log_2(1 - p) \) (See Fig. 3). The whole secret key \( S_{x^n} \) is assumed to be used for message transmissions.

**Proof:** (1) Let \( n_{ab} \) be the number of Bob’s bit errors per block of code. Let \( e > 0 \) be a parameter that satisfies \( 1 - (1 - e)\gamma \leq \gamma \). The parameters \( n \) and \( k \) are determined such that \( \Pr\{n_{ab} > t_e\} < e \) for the small given \( e > 0 \). Bob can generate \( Y_1^k \) from \( Y^n \) by using the common key \( k \), and he can correct all the errors except for a small probability \( \Pr\{n_{ab} > t_e\} < e \). In this case, \( \Pr\{S_{x^n} \neq S_{x^n}^k | k_e \leftarrow K_e \} = 1 - [1 - \Pr\{n_{ab} > t_e\}]^u < 1 - (1 - e)\gamma \leq \gamma \). Thus, Definition 4(1) is satisfied.

(2) According to the assumption, the probability with which Eve successfully generates \( X^k \) without \( k_e \) in one arbitrary trial is bounded by \( 1/2^k + 1/P(N_k) \). Let us suppose that Eve does not succeed in generating \( X^k \). Even in the case, she obtains \( Z^n \) and \( Z_{e}^{a,k} \). Because all of \( S_{x^n} \) is used in the message transmissions in accordance with the assumption, the \( X^k \)-related information obtainable by Eve is restricted to \( Z^n \) and \( Z_{e}^{a,k} \). First, let us consider the information that Eve obtains from only \( Z^n \). Because Eve does not have \( k_e \), \( H(X|Z^n) \leq H((X|k_e \leftarrow K_e)|Z^n) \) is satisfied. Here, let “\( X_1^k | k_e \leftarrow K_e \)” denote “\( X_1^k \)” for simplicity. Then, \( H(X|Z^n) \leq H(X_1^k|Z^n) \). If Eve’s information is only \( Z^n \), even though \( Z_{e}^{a,k} \) is generated from \( Z^n \), the amount of information she gets is unchanged, i.e., \( H(X|Z^n) = H(X_1^k|Z^n) \). \( H(X_1^k|Z^n) \leq H(X_1^k|Z^n) \) is generally satisfied. Thus, \( H(X|Z^n) \leq H(X_1^k|Z^n) \). Because \( X \) is a binary random number with a uniform distribution, \( X_1 \) generated from \( X^n \) with permutations also has such a property, i.e., \( H(X) = H(X_1) = 1 \). Thus, \( H(X|Z^n) = H(X) - H(X_1) = H(X_1) - H(X_1|Z^n) = I(X_1;Z^n) \). Next, let us consider the information that Eve obtains from \( Z_{e}^{a,k} \) as well as \( Z^n \). When \( Z_{e}^{a,k} \), a permutation of \( Z_{e}^{a,k} \), and the \( Z_{e}^{a,k} \) function as a code, \( Z_{e}^{a,k} \) is error-corrected and Eve obtains \( X_1^k \). This case is included in the case in which Eve succeeds in generating \( X_1^k \). Because we are discussing the case in which Eve does not succeed in generating \( X_1^k \), \( Z_{e}^{a,k} \) and \( Z_{e}^{a,k} \) do not function as a code. In this case, Eve cannot correct errors, but \( Z_{e}^{a,k} \) involves redundant information for correcting \( t_m \) bits of the errors of \( Z_{e}^k \) at maximum, where \( t_m \) is a characteristic parameter of the used code. If the function of \( Z_{e}^{a,k} \) is evaluated most advantageously from the Eve’s standpoint, the effect of \( Z_{e}^{a,k} \) is to repair \( H(X_1^k) - H(X_1^k|Z^n) \) back to \( H(X_1^k) \) for \( t_m \) symbols of \( Z_{e}^k \) at maximum (see Fig. 3(c)). For the remaining \( K - t_m \) symbols, the mutual information \( H(X_1^k) - H(X_1^k|Z^n) \) is unchanged because of the correction limit of the used code. Hence, when Eve does not succeed in generating \( X^k \), the amount of information per \( u \) blocks is

\[
ne \leq ut_mH(X_1^k) + u(k - t_m)[H(X_1^k|Z^n)] \leq ut_mH(X_1^k) + u(k - t_m)[H(X_1^k) - H(X_1^k|Z^n)].
\]

Because \( H(X) = H(X_1^k) = 1 \), \( ne \leq ut_m + u(k - t_m)[1 - H(X)(Z^n)]. \)

\( H(X(Z^n)) = H(p_{ae}) \) using the average bit-error rate. However, if the actual number of bit errors in one block is less than the average number of bit errors determined by \( p_{ae} \), Eve actually obtains more information than the average amount of information. Therefore, we must take Eve’s situation into account by considering the statistical fluctuations of the bit errors. In particular, we will consider the statistical fluctuations for \( u \) blocks in the bit sequence because the unit of the secret key generation is \( u \) blocks. Because \( H(X(Z^n)) \) is described using a bit-error rate, we describe the statistical fluctuations by using those of the bit-error rate that is evaluated for every \( u \) blocks of the bit sequence. Because \( p_{ae} = p_e(\bar{n}_{ae} - r\sigma_{n})/\bar{n}_{ae} \) is defined using \( r \) that satisfies \( \Pr\{n_{ae} < \bar{n}_{ae} - r\sigma_{n}\} < \gamma \), \( H(X(Z^n)) \) in each sequence of \( u \) blocks satisfies \( H(X(Z^n)) \geq H(p_{ae}) \) except for a small probability \( \Pr\{n_{ae} < \bar{n}_{ae} - r\sigma_{n}\} < \gamma \). In this case, \( ne \leq ut_m + u(k - t_m)[1 - H(p_{ae})] \).

Because Alice’s information per \( u \) blocks is \( n_A = uk \), we have \( n_A - n_e \geq u(k - t_m)H(p_{ae}) \).

Let \( n_A = n_h - n_e \) for any positive safe parameter \( n_h < n_e \). According to Corollary 5 in ref. [19], \( I(S_x^n;K_e;S_x^n) \leq 2^{-n_e}/\ln 2 \) can be achieved by universal hashing. Because of \( n_A - n_e = O(u) \), \( n_h \) and \( n_e \) can be also chosen to be \( O(u) \). Thus, \( I(S_x^n;K_e;S_x^n) \leq 2^{-n_e}/n_e \ln 2 < \gamma \) can be satisfied.

---

2 In this report, notations \( O(u) \) and \( O(1/u) \) are used for \( u \to \infty \).
for the given $\gamma$ by appropriately choosing $u$. This relation is satisfied except for the small probability $Pr\{n_{ue} < \bar{n}_{ue} - r\sigma_{u2}\} < \gamma$ and for the case that Eve does not succeed in generating $X_i^{(t)}$. $p_3$ in Definition 3 is the probability of successfully guessing information symbols for one block. Let $p_{2u}$ be this probability for $u$ blocks. Generally, $1 - p_{2u} \geq 1 - p_4$, and from the assumption, $1 - p_4 > 1 - 1/\ln 2 - 1/P(N_k)$. Thus, $1 - p_{2u} > 1 - 1/\ln 2 - 1/P(N_k)$. According to the above-mentioned two conditions, $Pr\{S_{A,\gamma} \in [k \leftarrow K_c; (S_{A,\gamma}^{(n)})_{u} < \gamma > (1 - \gamma)\} = (1 - p_{2u})$. Hence, $Pr\{S_{A,\gamma} \in (k \leftarrow K_c; (S_{A,\gamma}^{(n)})_{u} < \gamma > (1 - \gamma)\}$ is $1 - 1/\ln 2 - 1/P(N_k)$, and Definition 4(2) is satisfied.

(3) According to Theorem 3 in Ref. [19], $H(S_{A,\gamma} | k \leftarrow K_c) \geq n_t - 2^{-r(u)} / \ln 2$. Because $n_t$, $n_R = n_t - n_e - n_0$, and $n_t$ and $n_R$ are chosen to satisfy $2^{-r(u)} / \ln 2 < \gamma$, the relation $H(S_{A,\gamma} | k \leftarrow K_c) / n_t \geq 1 - 2^{-r(u)} / n_t > 1 - \gamma > 1 - \gamma$ is obtained. Because of $|S_{A,\gamma}^{(n)}| = n_t$, $\log_2 |S_{A,\gamma}^{(n)}| / n_t = 1$ is obtained. Thus, $H(S_{A,\gamma} | k \leftarrow K_c) / n_t > \log_2 |S_{A,\gamma}^{(n)}| / n_t - \gamma$ is satisfied. Hence, Definition 4(3) is satisfied.

(4) Using $H(S_{A,\gamma} | k \leftarrow K_c) \geq n_t - 2^{-r(u)} / \ln 2$, $n_t = n_R = n_t - n_e - n_0$, and $n_t = n_t - u > 0(k - t_0)h(p_{e})$, which is satisfied except for the small probability $Pr\{n_{ue} < \bar{n}_{ue} - r\sigma_{u2}\} < \gamma$, we obtain

\[
H(S_{A,\gamma} | k \leftarrow K_c) / u + \gamma \\
\geq n_t / u - 2^{-r(u)} / u \ln 2 + \gamma \\
\geq (k - t_0)h(p_{e}) - n_0 / u + (\gamma - 2^{-r(u)} / u \ln 2).
\]

Using $\gamma > 2^{-r(u)} / u \ln 2 > 2^{-r(u)} / u \ln 2 > 2^{-r(u)} / u \ln 2$, we obtain $\gamma > 2^{-r(u)} / u \ln 2 > 0$. Therefore, $H(S_{A,\gamma} | k \leftarrow K_c) / u + \gamma > (k - t_0)h(p_{e}) - n_0 / u$. Definition 4(4) requires $H(S_{A,\gamma} | k \leftarrow K_c) / u + \gamma > R_{e}$ for the small probability $Pr\{n_{ue} < \bar{n}_{ue} - r\sigma_{u2}\} < \gamma$. If $R_{e} = (k - t_0)h(p_{e}) - n_0 / u$ is selected, it satisfies Definition 4(4). As long as $R_{e}$ is less than that, Definition 4(4) is satisfied. Therefore, the selected value is the lower bound of $C_{d}$, where $C_{d}$ is the maximum of $R_{e}$. Thus, if $n_c$ is appropriately selected in accordance with the above discussion, the conditional secrecy capacity for the given $\gamma$ is $C_{d} \geq C_{d} (k - t_0)h(p_{e}) - n_0 / u$.\hfill\Box

Let $C_{d0}$ be the lower bound of $C_{d}$ in Lemma 1, i.e., $C_{d0} = (k - t_0)h(p_{e}) - n_0 / u$. The conditional secrecy capacity originates from the entropy $h(p_{e})$ of bit errors, as shown in Fig. 3. The common key is used only for transforming the entropy $h(p_{e})$ of bit errors into that of secrecy keys. Therefore, the conditional secrecy capacity maintains $C_{d} > 0$ for repeated use of $k_c$. Method 1 restricts the secret key-generation rate to $C_{d0}$ in order to repeatedly use the common key $k_c$. Lemma 1 assumes that Method 1 is computationally secure in the sense of Definition 3, and it is in Lemma 11 that security is proved. The reason why Lemma 1 is shown here prior to Lemma 11 is to determine the amount of $S_{A,\gamma}^{(n)}$, i.e., $n_t \leq unC_{d0}$.

For simplicity, Lemma 1 assumes that all of $S_{A,\gamma}^{(n)}$ is used in message transmissions. On the other hand, the case in which only part of $S_{A,\gamma}^{(n)}$ is used in message transmissions is as follows. For example, if $n_e$ bits are used in message transmissions and $(n_t - n_e)$ bits are leaked to Eve, $C_{d0}$ is transformed into $C_{d0}' = (unC_{d0} - n_t + n_e) / un$. Although the conditional secrecy capacity varies depending on the amount of leaked information, the fact that $unC_{d0}'$ indicates the capacity actually needed in message transmissions does not change. For this reason, Lemma 1 assumed that all of $S_{A,\gamma}^{(n)}$ is used in message transmissions.
5.1 Maximum Number of Errors Correctable by an Algebraic Code

The upper limit of the maximum number of code word errors \( t_{ma} \) that an algebraic code can correct is determined by the Hamming distance between code word vectors. Here, the algebraic code is not limited to a binary code, and \( t_{ma} \) is defined for a general code. When the coding is binary, \( t_{ma} = t_{mc} \) and when it is over \( GF(2^n) \), \( t_{ma} = mt_{mc} \).

There is a theorem called the Singleton bound for an \((n, k)\) linear code, i.e., \( d \leq n - k + 1 \), where \( n \) is the code word length, \( k \) is information symbol length, and \( d \) is minimum distance \([22,23]\). When equality is satisfied in this theorem, the corresponding code is called a maximum distance separable (MDS) code. Reed-Solomon is the most practical such code. When the Hamming weight \( w(e) \) of an error \( e \) is \( w(e) \leq (d-1)/2 \), the error can be exactly corrected. This is a classical bound in error correction. When \( (d-1)/2 < w(e) \leq d-1 \), the candidates for the code word vectors can be listed, and the error has the possibility of being corrected (list decoding). However, when \( w(e) \) is beyond \( d-1 \), the code word vector with the error usually enters the region of another code word vector and the error is not correctly detected. MDS codes have this characteristic for almost all errors, and the upper limit of the maximum number of errors that the codes can correct is given by the distance \( t_{mc} = d-1 \). This distance is equal to \( n-k \) in MDS codes, i.e., \( t_{mc} = n-k \), and this is intuitively understandable because \( n-k \) is the number of redundant code words. The estimate of \( t_{mc} = n-k \) for the upper limit of the maximum number of errors that the codes can correct has a sufficient margin, because although recent studies have shown the possibility of list decoding \([24,25]\), correctability is restricted to the relatively nearby region of \( (d-1)/2 \) for practical choices of \( n \) and \( k \).

The above paragraph describes the case of hard-decision decoding. There is also soft-decision decoding. However, soft-decision decoding extends the classical bound only by one or a few code words depending on the code employed \([26,27]\). This quantity is sufficiently small compared with the classical bound described above for list decoding for sufficiently large \( d \).

As described above, MDS codes are excellent from the viewpoint of clarifying the upper limit of the maximum number of errors \( t_{ma} \) that the codes can correct. For that reason, any practical system would use MDS codes. The example shown in sections 6.1.2 and 7.2 is a case of using MDS codes.

5.2 Concrete coding method

This section describes a concrete example of the encoding \( E_a \) and \( E_b \). Encoding \( E_a \) divides \( x^{(N_a)} \in \{0, 1\}^{N_a} \) into \( b^{(N_1)} \in \{0, 1\}^{N_1} \) and \( b^{(N_2)} \in \{0, 1\}^{N_2} \) by using \( k_c \in \{0, 1\}^{N_c} \). \( E_a \) encodes \( b_l \) and \( b_u \) independently by using an \((n, k)\) linear code over \( GF(2^n) \). Here, \( N_1 + N_2 = N_a \). The following is a concrete example of \( E_a \) and \( E_b \).

**Coding 1 [with common key]:**

\[ E_a : \{0, 1\}^{N_a} \times \{k_c\} \rightarrow \{0, 1\}^{N_1} \times \{0, 1\}^{N_2}, \]

\[ x^{(N_a)} \times k_c \mapsto b_l^{(N_1)} \times b_u^{(N_2)}, \]

where \( x \rightarrow b_l \) for \( k_c = 1 \) and \( x \rightarrow b_u \) for \( k_c = 0 \).

\[ E_b \text{ [Systematic (}n, k\text{) coding over } GF(2^n)\text{]} : \]

\[ \{0, 1\}^m \rightarrow \{0, 1\}^{m(k-\delta)}, b_l \mapsto e_l \text{ and } b_u \mapsto e_u. \]

Here, \( e_l \) and \( e_u \) are respectively parity check symbol vectors of \( b_l \) and \( b_u \). \( N_1 \) and \( k \) satisfy \( r_1 \delta_1 = N_1 - N_0 \leq N_0 \).

The above restrictions on \( N_1 \) and \( k \) are to prevent Eve from deriving the common key \( k_c \) part-by-part, as will be described in section 5.3. \( N_1 \) and \( N_2 \) respectively denote the numbers of “1”s and “0”s in \( k_c \in \{0, 1\}^{N_c} \). Figure 4 schematically shows Coding 1. The random number sequence \( x \in \{0, 1\}^m \) is divided into two groups depending on “0” and “1” in \( k_c \). The first bit of \( k_c \) is “1” in Fig. 4, and the first bit of \( x \) is allocated to group I. The second bit of \( k_c \) is “0,” and the second bit of \( x \) is allocated to group II. Subsequent bits are similarly allocated. Random numbers in each group are error-correcting coded independently group-by-group. Because the coding is group-by-group, if the grouping is not correctly done in the receiver, parity check symbols cannot be used. Because Eve does not know the common key, she cannot divide the random number sequence into groups or correct the bit errors. This impossibility makes secret communications possible.

The common key needs to be extended to handle a long random number sequence \( x \). However, we will simply use \( k_c \) repeatedly to evaluate the basic performance of this method. Of course, there would be an extension that used \( k_c \) as a seed key of pseudo-random numbers, its cryptographic power would be computationally strengthened.

3 The reason why the notations \( x^{(N_a)}, b_l^{(N_1)}, b_u^{(N_2)} \) are introduced is to differentiate them from \( x \in \{0, 1\}^m \) and \( b_l \) \& \( b_u \in \{0, 1\}^m \) respectively.
5.3 Common Key Length and Code Length

Although Definition 1 (Method 1) uses a common key $k_e$, the random number sequence $x$ itself consists of true random numbers, and it never reflects $k_e$. However, because Eve can get redundant information $z_0 = x_0$ (the parity check symbols $c_1$ and $c_0$ in Coding 1), she can possibly derive $k_e$ from it. The restrictions on $N_1$ and $k$ imposed in Coding 1 are to minimize this possibility. In this section, we discuss these restrictions.

**Lemma 2:** In Coding 1 using a common key $k_e$, if max($N_1, N_2) \leq mk$, then all information of the common key $k_e$ is needed even when coding one block.

**Proof:** When each symbol of $x$ of $N_k$ bits is allocated to $b_1$ or $b_0$, using $k_e$ in accordance with Coding 1, if max($N_1, N_2) \leq mk$, the numbers of $b_1$ and $b_0$ are less than or equal to $mk$. Therefore, all information of the common key is needed even when coding one block. □

When a block code is used, the $k_e$-deriving process using parity check symbols must be performed in units of one block. If max($N_1, N_2) > mk$, there is part of $k_e$ that is not used for forming one block, and deriving part of $k_e$ becomes possible. Therefore, $N_k$ and $k$ should be determined under the condition of max($N_1, N_2) \leq mk$.

Let us describe $N_1$ and $N_2$ as functions of $k_e$, i.e., $N_1(k_e)$ and $N_2(k_e)$. If we wholly consider $\{0, 1\}^{N_k}$ as $k_e$, max($N_1(k_e), N_2(k_e)) = N_k$. However, many cases satisfy $N_1(k_e) \sim N_2(k_e) \sim N_k/2$; therefore, we will restrict the set of common keys to the case satisfying $N_1(k_e) \sim N_2(k_e) \sim N_k/2$ as follows.

\[
\text{[Set of common keys $k_e$]: The set $\mathcal{K}_e$ of common keys $k_e$ of length $N_k$ is restricted to}
\]

\[
\mathcal{K}_e = \{k_e \in \{0, 1\}^{N_k} | -r_1\sigma_1 \leq N_1(k_e) - N_0 \leq r_1\sigma_1\}. 
\]

Here, $r_1$ is a design parameter that is typically chosen to be 3.

Because $N_1(k_e) + N_2(k_e) = N_k \in \mathbb{N}$, if $-r_1\sigma_1 \leq N_1(k_e) - N_0 \leq r_1\sigma_1$, then automatically $-r_1\sigma_1 \leq N_2(k_e) - N_0 \leq r_1\sigma_1$.

Therefore, if $k_e \in \mathcal{K}_e$, then max($N_1(k_e), N_2(k_e)) = N_0 + \lfloor r_1\sigma_1 \rfloor$, where $\lfloor x \rfloor$ denotes the maximum integer $\leq x$.

Thus, if $k$ is determined according to $N_0 + \lfloor r_1\sigma_1 \rfloor \leq mk$, all of $k_e$ is used to form one block in accordance with Lemma 2. The restrictions imposed on $N_1$ and $k$ in Coding 1 are for the above reasons.

Pr($k_e \in \mathcal{K}_e$) for $k_e \in \{0, 1\}^{N_k}$ is estimated as follows. The probability that each bit of a randomly chosen $k_e$ is 0 or 1 is $p = 1/2$. Thus, $N(k_e^\prime) (j = 1$ and 2) obeys a binomial distribution $P[N(k_e^\prime)]=\left(\begin{array}{c}N_k \\ N_j(k_e^\prime)\end{array}\right)p^{N_j(k_e^\prime)}(1-p)^{N_k-N_j(k_e^\prime)}$. The average is $N_0 = Nk/2$, and the variance is $\sigma_1^2 = N_k p(1-p)$. Thus,

Pr($k_e \in \mathcal{K}_e$) = \[ \frac{\sum_{N_j(k_e^\prime) \leq \lfloor r_1\sigma_1 \rfloor} P[N_j(k_e^\prime)]}{\sum_{N_j(k_e^\prime) \leq \lfloor r_1\sigma_1 \rfloor} P[N_j(k_e^\prime)]}, \]

where $\lfloor x \rfloor$ denotes the minimum integer $\geq x$. Let $\delta = Pr(k_e \in \{0, 1\}^{N_k}\setminus \mathcal{K}_e$. $\delta$ is given by $\delta = 1 - Pr(k_e \in \mathcal{K}_e)$. For example, when $r_1 = 3$ and the binomial distribution is approximated with a normal distribution, Pr($k_e \in \mathcal{K}_e$) = 0.9973, and $\delta = 0.0027$.

6. Deriving the common key

The process by which Eve tries to derive the common key $k_e$ is equivalent to her trying to derive $x_1 (x_1^*)$ of one block as described in this section. To derive secret key $s_{IA}$ in Method 1, $u$ blocks of $x_1$ are needed. Therefore, the computational complexity of deriving $s_{IA}$ is at least that of deriving $k_e$ as shown in Lemma 10. Thus, we first evaluate the computational complexity of deriving $k_e$.

The information obtainable by Eve is $z$ ($z^*$), $c_1$ ($c_1^*$), $c_0$ ($c_0^*$), and $s_{IA}^*$ that is not used in message transmissions, where $c_1^*$ and $c_0^*$ are the parity check symbol vectors corresponding to $z^*$. First, we will consider that only $z$ ($z^*$), $c_1$ ($c_1^*$), and $c_0$ ($c_0^*$) are leaked and estimate the computational complexity of deriving $k_e$. Section 6.1.1 considers the case without bit errors, and section 6.1.2 considers the case with bit errors. Next, section 6.2 takes $s_{IA}^*$ into consideration, and it is shown that the computational complexity of deriving $k_e$ does not decrease even if $s_{IA}^*$ is taken into consideration (Lemma 8). Using these results, the computational complexity of deriving $k_e$ is quantified (Lemma 9), and Method 1 using Coding 1 is proved to be computationally secure in the sense of Definition 2 (Theorem 1).
6.1 Deriving the common key using parity check symbols

6.1.1 Case without bit errors
Because Eve can obtain the parity check symbols \( c_1 \) and \( c_i \) in Method 1, if \( x \) (\( y \)) is transmitted without bit errors \((z = x)\), she can derive \( k_c \). Let us estimate the computational complexity of deriving \( k_c \). The routine of Coding 1 is 

\[
(x^{(N)}; k_c) \mapsto b_i^{(N)} \mapsto b_{II}^{(N)} \mapsto c_1 \cdots c_{II}.
\]

The information that Eve can obtain is \( z \) and \( c_1 \cdots c_{II} \). Because \( z \) itself has no information, the derivation of \( k_c \) is based on \( c_1 \cdots c_{II} \). Here, \( b_i \) and \( b_{II} \) are derived from \( c_1 \cdots c_{II} \), and \( k_c \) is derived by comparing \( b_i \) and \( b_{II} \) and \( z \).

Figure 5 shows the relation between the random number sequence and the first block of group I. Let \( G \) be the set of all elements over \( GF(2^m) \). Let \( b_0 \in G^k \) be an information symbol vector in the first block of group I that is obtained from a random number sequence \( x \) using \( k_c \). We will describe \( x \mapsto b_0 \) as \( b_0 = f(x; k_c) \), where \( b_0 \) is a row vector with \( k \) components over \( GF(2^m) \). Let \( c^{(p0)} \in G^m \) be the parity check symbol vector corresponding to \( b_0 \). \( c^{(p0)} \) is given by \( c^{(p0)} = b_0 G_0 \), where \( G_0 \) is the parity check symbol vector generating part of the generator matrix \( G \). The following lemma states a quantitative property about \( c^{(p0)} \).

**Lemma 3:** When only a parity check symbol vector is given in an \((n, k)\) linear code over \( GF(2^m) \), \( 2^{mk}/2^{m(n-k)} \) kinds of information symbol vectors exist for each parity check symbol vector.

**Proof:** An \((n, k)\) linear code over \( GF(2^m) \) consists of \( mk \) bits of information symbols and \( m(n-k) \) bits of parity check symbols. When the information symbols are derived from only parity check symbols, \( mk - m(n-k) \) bits cannot be determined. Therefore, \( 2^{mk}/2^{m(n-k)} \) kinds of information symbols exist for each parity check symbol vector. □

Let us define the set \( \mathcal{B}_0 \) for the \( 2^{mk}/2^{m(n-k)} \) kinds of information symbol vectors that are associated with \( c^{(p0)} \):

\[
\mathcal{B}_0 = \{ b_0 \in \mathcal{B} \mid c^{(p0)} = b_0 G_0 \}.
\]

Of course, \( b_0 \in \mathcal{B}_0 \).

Next, we define the set \( \mathcal{K}_0 \) by using \( \mathcal{B}_0 \) and \( \mathcal{K}_c \):

\[
\mathcal{K}_0 = \{ k_c \in \mathcal{K}_c \mid b_0 = f(x; k_c) \in \mathcal{B}_0 \}.
\]

The elements of \( \mathcal{K}_0 \) are the candidates of the common key. The number of candidates can be determined as follows:

**Lemma 4:** Suppose a common key \( k_c \in \mathcal{K}_c \) is used according to Coding 1. A random number sequence and a parity check symbol vector for the first block of group I or II are exactly given, and one of the positions of the random number sequence corresponding to the first bit of the common key is given to form the first block. The number of the candidates for the common key in this case is \( N_{\text{cand}} = 2^{mk-m(n-k)}(1-\delta) \) on average. Here, \( (1-\delta) \) is a factor due to \( k_c \notin \{0, 1\}^k \mathcal{K}_c \).

**Proof:** Let the random number sequence be \( x \), and let the parity check symbol vector be \( c^{(p0)} \), where the parity check symbol vector is represented by that of the first block of group I. The candidates of the common key are obtained by listing the elements of \( \mathcal{B}_0 \), comparing the elements with \( x \), and listing the elements of \( \mathcal{K}_0 \). The parity check symbol vectors are of \( 2^{mk} \) kinds, and the number of elements of \( \mathcal{K}_c \) is \( 2^{mk}(1-\delta) \). In this case, when a parity check symbol vector \( c^{(p0)} \) is given, the number of candidates of the common key is \( N_{\text{cand}} = 2^{mk-m(n-k)}(1-\delta) \) on average. □

The information obtainable by Eve about the first block of group I is \( z=x \) and \( c^{(p0)} \). According to Lemma 4, Eve can narrow down the candidates of \( k_c \) to \( N_{\text{cand}} \) on average. This number can be made tremendously large if we appropriately choose \( N_k \), \( m \), \( n \), \( k \), and \( \delta \). However, a listing is possible in principle even though no memory with a high enough capacity exists. Eve can check each of the listed elements by decoding the blocks of group II and other blocks of group I, and she can continue this process until the candidates of \( k_c \) have been narrowed down to one.

**Corollary 1:** Let us assume that only a random number sequence and parity check symbols are given in Method 1 using Coding 1. It is impossible to derive only part of the common key.

**Proof:** This claim is apparently true from the fact that deriving the common key is processed in units of one block and that one block is constructed using all the information about the common key, due to the condition \( N_0 + \lfloor r \sigma \rfloor \leq mk \). □

**Corollary 2:** Let us assume that only a random number sequence and parity check symbols are given in Method 1 using Coding 1. The computational complexity of deriving at least one bit of the common key is \( O(N_{\text{cand}}) \) under the condition that no bit errors exist. In other words, an exhaustive search of \( N_{\text{cand}} \) is needed.

**Proof:** Because deriving only part of the common key is impossible according to Corollary 1, the whole common key needs to be derived even for only one bit. In this case, the computational complexity is \( O(N_{\text{cand}}) \) because the process in narrowing down the candidates of the common key based on Lemma 4 involves the complexity of \( O(N_{\text{cand}}) \). □
6.1.2 Case with bit errors

When bit errors exist, Eve needs to consider all possible error patterns to derive the common key. She modifies information symbols according to each bit-error pattern and uses the strategy described in the above subsection.

The number of errors obeys a binomial distribution, and the occurrence probability is highest for the average number of errors. Therefore, let us first concretely estimate the number of error patterns for the average number of errors.

There is no bit error in parity check symbols in Method 1. The number of information symbols in an \((n, k)\) code over \(\text{GF}(2^m)\) is \(mk\) bits, and the average number of bit errors is \(\bar{n}_e = \rho n/mk\) in one block. The number of bit-error patterns is \(\binom{mk}{\bar{n}_e}\), and it can be approximated using Stirling’s formula, \(n! \approx \sqrt{2\pi n} n^ne^{-n}\), as

\[
\binom{mk}{\bar{n}_e} \approx \sqrt{2\pi (mk - \bar{n}_e)} \left(\frac{mk}{mk - \bar{n}_e}\right)^{\bar{n}_e} \left(\frac{mk}{\bar{n}_e}\right)^{\bar{n}_e}.
\]

The parameters \(n\) and \(k\) are determined so as to correct errors with a sufficient margin; e.g., the \(3\sigma_2\) region is included inside the error-correctable region with a sufficient margin, where \(\sigma_2\) is the standard deviation of the bit-error distribution. An MDS code is error-correctable for code word errors satisfying \(w_d(e) \leq (d - 1)/2\), as described in section 5.1; therefore, the bit errors up to \((n-k)/2\) are exactly correctable, and \(\bar{n}_e + 3\sigma_2 < (n-k)/2\) is the condition for determining \(k\). For example, Let us consider the case of \(m = 8, n = 255, k = 167\), and \(p_{\text{eff}} = 0.1\), where \(p_{\text{eff}}\) is the code-error rate given by \(p_{\text{eff}} = 1 - (1 - \rho e)^n\). In this case, \(\bar{n}_e \approx 17.5\), \(\sigma_2 \approx 4.15\), \(\bar{n}_e + 3\sigma_2 = 29.95 < (n-k)/2 = 44\), and \(\binom{mk}{\bar{n}_e} \approx 2.8 \times 10^{29}\) (1.0 \(\times 2^{131}\)).

**Lemma 5:** Let us assume that only a random number sequence and parity check symbols are given in Method 1 using Coding 1. Let \(N_{\text{ep}}\) be the number of error patterns in one block. When bit errors exist, the computational complexity of deriving at least one bit of the common key is \(O(N_{\text{ep}} \cdot N_{\text{can}})\).

**Proof:** Let us choose the first block of group I as a representative (see Fig. 5). Let \(e_i\) be an error vector in the first block of group 1, where the Hamming weights satisfy \(w_0(e_i) \leq (d - 1)/2, i = 1, 2, 3, \ldots\). We define \(b_i' = b_i + e_i\) for \(b_i \in B_i\). Given \(b_i = b_0 + e_i\) and \(c_i(p_0) = b_0 G_p\), then \(b_i' G_p = (b_0 + e_i) G_p = (b_0 + b_i - b_0) G_p = b_i G_p = e_i(p_0)\), and \(B_i \cap B_j = \phi\) for \(B_i = \{b_i' | c_i(p_0) = b_i G_p\}\). Let \(x_e\) be a random number sequence with bit errors. Moreover, if we define \(\mathcal{K}_d = \{k_{e0} \in \mathcal{K}_c | b_i' = f(x_i | k_{e0}) \in B_i\}\), the elements of \(\mathcal{K}_d\) are the candidates of the common key in error pattern \(e_i\). Similarly, if we define \(b_j = b_0 + e_i, e_i(p_0) = b_j G_p, B_j = \{b_j' | e_i(p_0) = b_i G_p\}\), and \(\mathcal{K}_d = \{k_{e0} \in \mathcal{K}_c | b_j' = f(x_i | k_{e0})\} \in B_j\) for another error vector \(e_j\), then \(B_i \cap B_j = \phi\) and \(B_i \cap B_j = \phi\). Because \(b_i' = f(x_i | k_{e0})\) is a single-value function, \(\mathcal{K}_d \cap \mathcal{K}_d = \phi\). Thus, there is no overlap between the candidates of the common key for different error vectors. Because the occurrence of each error pattern is probabilistic, Eve must consider all such patterns when deriving the common key and there are candidates of the common key described in Lemma 4 for each pattern. Thus, the computational complexity of deriving at least one bit of the common key is \(O(N_{\text{ep}} \cdot N_{\text{can}})\) according to Corollary 2. \(\Box\)

For the parameters described in this section, when only the error patterns for the average number \(\bar{n}_e \approx 17.5\) of errors are considered, the computational complexity of deriving the common key is \(O(2^{131} \cdot N_{\text{can}})\) according to Lemma 5.

This estimation considers only the error patterns for the average number of errors, while the actual number of bit errors is distributed with a standard deviation of \(\sigma_2\) around \(\bar{n}_e\). We need to consider all possible error patterns, and their probabilities of occurring as well. The number of patterns when each probability is different can be estimated using the Shannon entropy. For example, an entropy of 131 bits effectively corresponds to \(2^{131}\) error patterns.

The parameters used in coding are controlled such thatBob can correct any errors. The number of bit errors is in the error-correctable region, i.e., \(\Pr\{n_e > (d - \ldots\}
1)2} << 1. In this case, Lemma 6 is satisfied.

**Lemma 6:** Let us assume that only a random number sequence and parity check symbols are given in Method 1 using Coding 1. Let \( n_e \) be the number of bit errors in one block. If \( \Pr \{n_e > (d - 1)/2 \} << 1 \) is satisfied, the computational complexity of deriving at least one bit of the common key is \( O(2^H N_{\text{cand}}) \), where \( H_p \approx mk \cdot h(p_{E}) \).

**Proof:** The number of error patterns in one block is \( \binom{mk}{n_e} \) for \( n_e \) bit errors, and the occurrence probability of each error pattern is \( p_n = p_{E_n}^{n_n} (1 - p_{E_n})^{mk-n_n} \). The Shannon entropy of the error-correctable region, i.e., \( 0 \leq n_e \leq (d - 1)/2 \), is \( H_p = - \sum_{n_e} \binom{mk}{n_e} p_n \log_2 p_n \). Therefore, we obtain \( H_p \approx mk \cdot h(p_{E}) \) by extending the region of the sum to \( mk \). This quantity considers all error patterns for an \( mk \) bit sequence. In this case, it is equal to the equivocation for \( mk \) bits, and \( H_p \approx mk \cdot h(p_{E}) \) at least one bit of the common key is \( O(2^H N_{\text{cand}}) \), and \( H_p \approx mk \cdot h(p_{E}) \) \( \Box \).

This computational complexity can be checked by making the following rough estimate. Suppose \( m = 8 \), \( k = 167 \), and \( p_{\text{eff}} = 0.1 \) (\( p_{E} \approx 0.0131 \)); then \( H_p \approx 134 \). From \( \sigma_z \approx 4.15 \) and \( \binom{mk}{n_e} \approx 1.0 \times 2^{131} \), we find that \( 2\sigma_z \binom{mk}{n_e} \approx 2^{H_p} \), and \( 2^{H_p} \) is surely the effective number of error patterns.

The truth or falseness of each candidate can be judged by decoding a sufficient number of blocks with the candidate common key as follows: When a candidate is true, the number of bit errors is distributed around \( n_e \) in all blocks, and parity check symbols are never an error. In contrast, parity check symbols can be an error when a candidate is false. Moreover, in this case, because the information symbols become a haphazard sequence, the code word vector for it is probabilistically uniformly spread out over the code word vector space, and the number of bit errors is uniformly distributed throughout the correctable error numbers. Thus, each candidate can be judged as being true or false from the distribution of errors if a sufficient number of blocks are checked.

**Lemma 7:** Let us assume that only random number sequence and parity check symbols are given in Method 1 using Coding 1. The computational complexity of deriving the information symbols of one block, i.e., \( x_i \left( x_i \right) \), is equal to that of deriving at least one bit of the common key.

**Proof:** As shown in the proof of Lemma 4 and the following paragraph, the process of deriving the common key \( k_s \) consists of listing the candidates of information symbols and \( k_s \), checking each candidate \( k_s \) using other blocks, and obtaining the final solution. The process of deriving the information symbols of a target block also consists of listing the candidates of information symbols and checking them. To check them, the candidates of \( k_s \) are listed and each candidate \( k_s \) is checked using other blocks. It is when the final solution of \( k_s \) is confirmed that the candidate of the information symbols is confirmed. According to Corollary 1, it is impossible to derive only part of the common key. Thus, Lemma 7 is satisfied. \( \Box \)

6.2 Deriving the common key by using \( s_{eA}^* \)

According to the assumption, Eve obtains not only \( z (z^*) \), \( c_1 (c_1^*) \), and \( c_2 (c_2^*) \) but also \( s_{eA}^* \) but is not used message transmissions. Can \( s_{eA}^* \) case deriving the common key \( k_s \)? The following Lemma 8 sweeps away this concern.

The conditional secrecy capacity is \( C_s \geq C_s = (k - t_m)/n \cdot h(p_{e}) - n_1/u \) for binary coding from Lemma 1. When the coding is over GF(2^n), \( n \) and \( k \) are translated into \( mn \) and \( mk \), and \( t_m = t_m \). Therefore, \( C_s \geq C_s = (k - t_m)/n \cdot h(p_{e}) - n_1/u \) when the coding is over GF(2^n). The condition for \( n_1 \) in Method 1 is translated into \( n_1/u \) \( \leq C_s \). Thus, \( n_1/u \leq C_s = m(k - t_m)/h(p_{e}) - n_1/u \). Let \( H'_n = n_1/u \) and \( H_0 = mnC_0; \) then, \( H'_n \leq H_0 \). \( H'_n \) is the number of secret keys generated per block. In Method 1, \( u \geq 1 \).

Secret keys are generated from noise. There is a rare case where the number of bit errors is extremely small owing to the statistical fluctuations of noise. We assume that the rare case is bounded with a small quantity \( \gamma \), i.e., \( \Pr \{n_{ue} < n_1/e - n_1/2u \} < \gamma \). In addition, we assume that Eve’s residual information after the secret key generation is also bounded by the small quantity \( \gamma \), i.e., \( I(S_{eA}^* | k_s \leftarrow K_s; S_{kA}^* | n_e) < \gamma \).

**Lemma 8:** In Method 1 using Coding 1, the computational complexity of Eve’s deriving at least one bit of the common key is equal to that of deriving it only from a random number sequence and parity check symbols if the effects of \( \Pr \{n_{ue} < n_1/e - n_1/2u \} < \gamma \) and \( I(S_{eA}^* | k_s \leftarrow K_s; S_{kA}^* | n_e) < \gamma \) are negligible, where \( \gamma \) is a
small quantity.

**Proof:** According to the assumption, Eve can obtain \( s_{k_A^*} \) that is not used for message transmissions. Let us assume that \( s_{k_A^*} \) is generated from the block Eve wants to analyze. If the inverse operation of universal hashing used in generating secret keys were easy for Eve, the information symbols \( x_1^* \) in that block could be derived, and the number \( N_n \) of candidates would satisfy \( N_n \geq 2^{mk\cdot n} \), where equality corresponds to the case of \( u = 1 \). When the information symbols are derived by using one block of parity check symbols, the number of candidates is \( N_p = 2^{mk(u-\alpha_k)} \) according to Lemma 3. When the error-correcting code works correctly, the amount of redundant information \( m(n - k) \) exceeds the entropy of the bit errors \( mk\cdot h(p_k) \), i.e., \( m(n - k) \geq mk\cdot h(p_k) \). Because of \( H_s = m(k - t_{se})\cdot h(p_d) - n/u \cdot mk\cdot h(p_k) > H_s \) is satisfied. Because \( H_s \geq H_s' \), \( m(n - k) > H_s' \). Thus, \( N_p < N_n \). Next, let us assume that Eve tries to correlate the information in the random-number transmission stage with \( s_{k_A^*} \). However, because \( s_{k_A^*} \) is generated in the capacity of \( n/u \leq nmc_C^* \), as long as Eve fails to derive \( x_1^* \), the number of candidates in the random-number transmission stage is uncorrelated with \( s_{k_A^*} \) if the effects of \( \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} < \gamma \) and \( I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e < \gamma \) are negligible. (See Fig. 3 and proof (2) of Lemma 1.) Without any correlation, it is advantageous for Eve to use the information in the random-number transmission stage when trying to derive the common key, but not to use \( s_{k_A^*} \) because of \( N_p < N_n \). Therefore, Eve will use the information in the random-number transmission stage until she succeeds in deriving \( x_1^* \). Thus, Lemma 8 is satisfied. □

\[
\Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} \text{ can be made exponentially small, as follows. The number of bit errors obeys a binomial distribution. When it is approximated with a normal distribution, } \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} = \left(\frac{1}{\sqrt{2\pi}}\right)\int_{-\infty}^{(n_{ue} - \bar{p}_{ue})/r} e^{-t^2/2} dt = O(e^{-r^2/2})/r, \text{ where } t = (n_{ue} - \bar{p}_{ue})/\sigma_{ue}. \text{ It is exponentially small if } r \text{ is appropriately chosen. } I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e \text{ can be also exponentially small. As described in the proof of Lemma 1, } I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e \leq 2^{-n/n} \ln 2. \text{ Because } n_e \text{ and } n_e \text{ can be chosen to be } O(u/n), \text{ when } u/n \text{ is sufficiently large, } 2^{-n/n} \ln 2 \text{ is exponentially small. Thanks to these characteristics, we can choose a sufficiently small } \gamma. \]

**Corollary 3:** In Method 1 using Coding 1, the computational complexity of Eve’s deriving \( x_1^* \) is equal to that of deriving it only from a random number sequence and parity check symbols if the effects of \( \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} < \gamma \) and \( I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e < \gamma \) are negligible, where \( \gamma \) is a small quantity.

**Proof:** The corollary is apparent from the proof of Lemma 8. □

Lemma 9 follows from Lemmas 6 and 8.

**Lemma 9:** In Method 1 using Coding 1, the computational complexity of Eve’s deriving at least one bit of the common key \( k_e \) is \( O(2^{H_e} N_{\text{cand}}) \) if the effects of \( \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} < \gamma \) and \( I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e < \gamma \) are negligible, where \( \gamma \) is a small quantity.

**Corollary 4:** The effective key length in Method 1 using Coding 1 is \( N_k - m(n - k) + mk\cdot h(p_k) + \log_2(1 - \delta). \)

**Proof:** The corollary is apparent from \( \log_2\left(2^{H_e} N_{\text{cand}}\right) = N_k - m(n - k) + mk\cdot h(p_k) + \log_2(1 - \delta). \) □

**Lemma 10:** In Method 1 using Coding 1, the computational complexity of Eve’s deriving at least one bit of the secret key \( s_{k_A} \) is at least \( O(2^{H_e} N_{\text{cand}}) \) if the effects of \( \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} < \gamma \) and \( I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e < \gamma \) are negligible, where \( \gamma \) is a small quantity.

**Proof:** Because secret keys are generated from \( x_1 \) in units of \( u \) blocks, when Eve derives at least one bit of the secret key \( s_{k_A} \), she needs \( x_1 \) for \( u \) blocks; moreover, she needs to perform algorithm \( S \) for generating the secret keys. From Corollary 5, the computational complexity of only deriving one block of \( x_1 \) is \( O(2^{H_e} N_{\text{cand}}) \) if the effects of \( \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} < \gamma \) and \( I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e < \gamma \) are negligible. To derive at least one bit of \( s_{k_A} \), algorithm \( S \) must be analyzed moreover. Thus, Lemma 10 is satisfied. □

**Lemma 11:** In Method 1 using Coding 1, encoding \( E_3 \) is computationally secure in the sense of Definition 3.

**Proof:** Let \( \eta = 1/\gamma \). From Corollary 5, the computational complexity of Eve’s deriving \( x_1 (x_1^*) \) is \( \{O(2^{H_e} N_{\text{cand}})\}1 - O(1/\eta) + O(1/\eta) \} \) by taking into account \( \Pr\{n_{ue} < \bar{p}_{ue} - r\sigma_{ue}\} < \gamma \) and \( I(S_{k_A^*}^{ue}|k_e-K_e; S_{k_e}^{ue})/n_e < \gamma \), where \( \gamma \) is a
small quantity; the term $O(1/\eta)$ comes from these rare cases, and the term $O(2^{h_2} N_{\text{cand}}[1 - O(1/\eta)])$ comes from the other cases. Thus, the probability of successfully guessing information symbols, $p_0$ in Definition 3, is

\[ p_0 \leq 1/2^k + 1/\left( O(2^{h_2} N_{\text{cand}}[1 - O(1/\eta)]) + O(1/\eta) \right). \]

The parameter $mk$ is determined such that it satisfies $N_0 + [r \sigma_1] \leq mk$, and thus, $mk = O(N_k)$. Moreover, $mn = O(mk)$. Thus, $N_k - m(n-k) = O(N_k)$. As is apparent from $N_{\text{cand}} = 2^{N_k-m(n-k)(1-\delta)}$, the parameters $N_k$, $m$, $n$, and $k$ are chosen such that $N_k - m(n-k) > 0$. In summary, $N_k - m(n-k) = O(N_k) > 0$. Hence, $N_{\text{cand}} = 2^{N_k-m(n-k)(1-\delta)} > P(N_k)$ is satisfied at $N_k \to \infty$ for every polynomial equation $P(N_k)$. In addition, $\gamma \to 0$ can be chosen for $N_k \to \infty$. Therefore, when $k_0$ is chosen sufficiently large, $p_0 < 1/2^k + 1/P(N_k)$ is satisfied for $N_k \geq k_0$. Thus, encoding $E_0$ in Method 1 using Coding 1 is computationally secure in the sense of Definition 3. □

The following theorem is obtained from Lemma 10.

**Theorem 1:** Method 1 using Coding 1 is computationally secure in the sense of Definition 2.

**Proof:** Let $\eta = 1/\gamma$. From Lemma 10, the computational complexity of Eve’s deriving any one bit of the secret key $s_A$ is at least $\{O(2^{h_2} N_{\text{cand}}[1 - O(1/\eta)]) + O(1/\eta)\}$ by taking into account $Pr\{n_{\text{sec}} < n_{\text{sec}} - r \sigma_2\} < \gamma$ and $I(s_A^n | K, K_2; S_{\text{sec}}^n)/n_c < \gamma$, where $\gamma$ is a small quantity. Therefore, the probability of successfully guessing the secret key, $p_0$ in Definition 2, is

\[ p_0 \leq 1/2^k + 1/\left( O(2^{h_2} N_{\text{cand}}[1 - O(1/\eta)]) + O(1/\eta) \right). \]

The parameter $mk$ is determined such that it satisfies $N_0 + [r \sigma_1] \leq mk$, and thus, $mk = O(N_k)$. Moreover, $mn = O(mk)$. Thus, $N_k - m(n-k) = O(N_k)$.

In this case, $N_0 + [r \sigma_1] = 1335 < mk$ is satisfied for $r_1 = 3.5$. In other words, we can choose $N_k = 2496$ in this code. The computational complexity of deriving the common key is proportional to $2^{h_2} N_{\text{cand}} \approx 2^{mk \cdot [r_1]} = 2^{1926(1-\delta)} \approx 2^{1926(1-\delta)}$ for $p_\text{e} \approx 0.0131$ due to Lemma 6. This value seems to be sufficiently large; if a larger value is required, we can enlarge the block size.

So far, we have used the common key repeatedly. This is because we wanted to evaluate the basic performance of the proposed method. However, if we use the common key more cryptographically, i.e., as a seed key of pseudo-random numbers, the security will increase computationally. Moreover, bit errors will still enhance security. In the analysis described in section 6.1.2, the security was determined by only the
computational complexity of the analysis for one block. This is because
the common key is repeatedly used. However, if pseudo-
random numbers are used instead, the number of blocks
needed to derive the common key increases. Let the needed
number of blocks be \( N_T \). In this case, the entropy of
noise that affects the analysis is \( N_T\cdot mk\cdot h(p_E) \), and the
effective key length increases. This effect is powerful
because it is information theoretic. For the above example,
where \( m(n-k)=704 \) and \( H_p \approx 134 \),
\[ 2^{N_H} \cdot N_{\text{cond}} > 2^{N_k}(1-\delta) \]
is satisfied for \( N_T \geq 6 \). In this case, an
exhaustive search of \( k_e \) is needed for deriving the
common key. Moreover, the following observations can
be made. We assumed that parity check symbols exactly
leak to Eve in Method 1. However, when we use Method
2, it adds bit errors to the parity check symbols, and
consequently, its security increases. Moreover, \( \chi \) and \( \chi_e \) are transmitted without encryption in Methods 1 and 2. If
\( \chi \) and \( \chi_e \) are encrypted with pseudo-random numbers,
security increases computationally, although another key
is needed.

| Table 1 Summary of Method 1 using Coding 1 | Example |
|-------------------------------------------|---------|
| (0) Share common key \( k_e \ni \{0, 1\}^n \cdot r \ni \{0, 1\}^n \cdot n \ni N_k \cdot N_e \ni N \approx 1248 \) | \( N_k \approx 2496 \) |
| (1) Generate random number sequence \( x \ni \{0, 1\}^n \) | \( r \approx 3.5 \) |
| (2a) Divide random number sequence \( x^{n/N_k} \ni b_1^{(n/N_k)} \ni b_0^{(n/N_n)} \ni e \) | \( \sigma_1 \approx 24.98 \) |
| (2b) Perform \( (n, k) \) block coding: \( b_1 \ni e_1 \) and \( b_0 \ni e_0 \) | Fig. 1 |
| \( N_e \ni [r, s] \ni m \leq m \) | \( n = 255 \) |
| (3) Generate secret key (Privacy amplification) Table II |

7.2 Conditional secrecy capacity

Let us estimate an example of the conditional secrecy
capacity when using Method 1, a \((255, 167)\) linear MDS
code over \( GF(2^5) \), and \( P_{\text{eff}} = 0.1 \) \((p_E \approx 0.0131)\).

As described in section 6.2, when the coding
is over \( GF(2^m) \), \( C_s \ni (k - t_{\text{sec}})/n \ni h(p_{\text{sec}}) - n/umn \). If \( u = 1 \)
and \( r = 3 \), then \( n_{se} \ni ukp_{\text{sec}} = 17.5 \), \( s_{\text{sec}} \ni 0.00376 \), and \( h(p_{\text{sec}}) \approx 0.0357 \). Thus, \( C_s \ni (k - t_{\text{sec}})/n \ni h(p_{\text{sec}}) - n/umn \approx 0.0111 - n/umn \). If we choose \( n_{se} = 10 \), then \( \gamma \ni 0.00615 \) and \( \gamma \ni 12.5 \). Here, \( \gamma \) \ni is used
to indicate a lower bound.

Let us determine \( \gamma \) by referring to the above
values, although this process is the inverse of that from
the viewpoint of the meaning that \( \gamma \) should be given first.
The condition in Definition 4 (1) is \( \Pr\{S_{k_e}^{(n)} \neq S_{k_e}^{(n)} | k_e \ni K_e \} < \gamma \). Let \( n_{kb} \) be the number of Bob’s code errors
in one block. When \( u = 1 \), then \( \Pr\{S_{k_e}^{(n)} \neq S_{k_e}^{(n)} | k_e \ni K_e \} = \Pr\{n_{kb} > (n - k)/2 \} \). The method in this report works efficiently when \( p_B - p_E \ll p_E \) (see section 7.5).

Therefore, let us assume \( p_B = p_E \) as an example. In this case,
\( \Pr\{n_{kb} > (n - k)/2 \} < 4.70 \times 10^{-10} \), where the third
decimal place is rounded up. Definition 4 (2) requires
\( I(S_{k_e}^{(n)}|k_e \ni K_e; S_{k_e}^{(n)}) | n < \gamma \), except for the rare case of
\( n_{se}/umn < p_{\text{sec}} \). The probability of the rare case is
\( \Pr\{n_{se}/umn < p_{\text{sec}} \} < 4.48 \times 10^{-4} \) for \( p_{\text{sec}} = 0.1 \), and
\( I(S_{k_e}^{(n)}|k_e \ni K_e; S_{k_e}^{(n)}) | n < 2^{n/2} pn \ln 2 < 1.13 \times 10^{-4} \). From
the above two kinds of small values, \( \gamma \leq \max(4.70 \times 10^{-10},
4.48 \times 10^{-4}, 1.13 \times 10^{-4}) \). Table 1 summarizes
these values.

\( C_s \) increases as \( u \) increases, and \( \gamma \) can be decreased as \( r \) and \( n \) are increased. For \( u = 10 \) and \( r = 5 \), then \( n_{se} \ni ukp_{\text{sec}} = 17.5 \), \( s_{\text{sec}} \ni 0.00376 \), and \( h(p_{\text{sec}}) \approx 0.0357 \). Thus, \( C_s \ni 0.0212 - n/umn \). Here, if \( n_{se} = 16 \), then \( C_s \ni 0.0204, n_{se} \ni 416 \). Table 1 lists \( \gamma \)-related values.

It also shows the case of \( u = 10 \), \( r = 3 \), and \( n_{se} = 10 \). When
\( u \ni x_e \), then \( h(p_E) \approx 0.101 \) and \( C_s \ni 0.0312 \).

| Table 2 Lower bound of conditional secrecy capacity and related quantities at \( (n, k) = (255, 167) \), \( m = 8 \), and \( P_{\text{eff}} = 0.1 \) \((p_E \approx 0.0131)\). (1) is related to Definition 4 (1); (2.1) is related to Definition 4 (2) and (4); (2.2) is related to Definition 4 (2). The third decimal place is rounded up in those rows. \( n_{kb} \) denotes the number of Bob’s code errors in one block. \( n_{se} \) denotes the number of Eve’s bit errors in \( u \) blocks. |
|---|---|---|---|---|
| \( u \) | \( 1 \) | \( 10 \) | \( 10 \) | \( 10 \) |
| \( r \) | \( 1 \) | \( 3 \) | \( 5 \) | \( 10 \) |
| \( n_{se} \ni 10 \) | \( 16 \) | \( 10 \) | \( 10 \) |
| \( (1) \Pr\{n_{kb} > (n - k)/2 \} < 4.70 \times 10^{-10} \) | \( 4.70 \times 10^{-10} \) | \( 4.70 \times 10^{-10} \) | \( 4.70 \times 10^{-10} \) |
| \( (2.1) \Pr\{n_{se}/umn < p_{\text{sec}} \} < 4.48 \times 10^{-4} \) | \( 9.63 \times 10^{-4} \) | \( 5.07 \times 10^{-4} \) | \( 5.07 \times 10^{-4} \) |
| \( (2.2) \gamma \ni 4.48 \times 10^{-4} \) | \( 9.63 \times 10^{-4} \) | \( 5.07 \times 10^{-4} \) | \( 5.07 \times 10^{-4} \) |
| \( \gamma \ni 12.5 \) | \( 50.6 \) | \( 41.6 \) |

7.3 Multiple codes

Two kinds of error-correcting codes are often combined
to make the error correction perfect, e.g., product codes
and concatenated codes [23]. The method in this report
can be modified to suit double coding using two kinds of
common keys. A concrete example is as follows.

The random numbers, amounting to \( N_k \) blocks of groups I and II, of a sequence coded using \( k_e \) are
shuffled, and the shuffled sequence is then coded using
another common key \( k_{ed} \). The parameters for the two
codes do not need to be the same. Decoding is possible
from either the \( k_{ed} \) or \( k_{ed} \)-related code, and this double
coding is resistant to burst errors. For example, let us
decode the \( k_{ed} \)-related code first and assume there are
residual errors. Because the random number sequence is
shuffled, the residual errors are distributed over multiple
blocks in the \( k_{ed} \)-related code. They can be corrected
through \( k_{ed} \)-related error correction. Here, although
the shuffling process becomes computationally expensive,
\( N_k \) should be as large as possible. The value of \( N_k \) should
be determined on the basis of the processing performance
of the transmitter and receiver.
The double coding is for complete error correction, but there is a possibility that all errors will be corrected in one decoding. The security of this method, therefore, is quantified by the complexity of the decryption process of one of the two codes. An important thing in double coding is to prevent the parity check symbols in one of the two codes from affecting the complexity of the decryption process in the other code.

As mentioned in section 6.1.1, the process of deriving the common key includes listing the candidates of the information symbols for one block as a basic component. Because this listing is a closed process for one block, the parity check symbols in the \( k_d \) \( \cd \)-related code do not contribute to the process of listing the candidates of the information symbols in the \( k_d \) \( \cd \)-related code. Therefore, the security of this method is determined by the complexity of deriving only one of the common keys. However, the conditional secrecy capacity changes. Here, let the block size be the same for both codings. Because redundant information on the \( k_d \) \( \cd \)-related code can correct \( t_m \) \( \cd \) bits at maximum, the conditional secrecy capacity is \( C_s \geq (k - t_m - t_{\cd})/n \cdot h(p_{\cd}) - n_s/\ln n \). Here, \( t_m \) and \( t_{\cd} \) can be set less than those of single coding thanks to double coding.

7.4 Noise source

The output of an LD used in optical communications includes noise; the phase of the output light is especially noisy and is sufficiently random [15]. Coding methods like Phase-Shift Keying (PSK) or Differential Phase-Shift Keying (DPSK) use the phase of light. Thus, the method in this report can use phase noise-related bit errors as a resource that is always available in optical communications.

7.5 BER in random-number transmission channel

As mentioned in the preceding subsection, the phase noise of an LD output is directly usable in optical communications. However, when the environmental noise \( F_l \) in a transmission channel is large, the condition \( p_E \approx p_B \) \( p_B \approx p_E \) is not satisfied, where much redundant information is required and the conditional secrecy capacity decreases. One solution in this case is to code the transmitter output \( F_c(X) \) with an error-correcting code and to build a pseudo-errorless channel \( p_E \approx p_B \). Because the purpose of this coding is to transmit a random number sequence with errors correctly, the decoded sequence has errors, and Eve does not obtain any new information.

8. Summary

Secure communications using noise generally need a mechanism to make Eve less advantageous than Bob. However, such a mechanism does not always exist intrinsically. This report described an extrinsic method that makes Eve disadvantageous by using a common key. The common key, error-correcting code, and noise are managed in a cooperative manner, and the secret keys are generated from noise. Messages are encrypted with the secret keys by using a one-time pad. As a result, information leaks that are meaningful to Eve are restricted to the parity-check symbols for the random numbers. It is possible to derive the candidates of the common key from the parity check symbols, and the security of this method can be quantified in terms of the computations needed for an exhaustive search of the candidates. We calculated the number of the candidates of the common key by assuming all parity check symbols were leaked to Eve without bit errors. The number is \( 2^{h_p} N_{\cd} \), and it determines the security of this method. Its logarithm \( N_K = m(n - k) + mk \cdot h(p_{\cd}) + \log_2 (1 - \delta) \) corresponds to the effective key length. Methods with computational security generally face the threat that an efficient decryption method might be found. However, this method does not rely on any mathematical difficulties, and therefore, there is no threat that a more efficient decryption method than an exhaustive search might be found. The method requires listing the information symbols from the parity check symbols followed by listing the candidates of the common key in decryption. This threat-less form of security can be used to protect highly confidential information like government and military secrets, although its security level is computational. However, it requires privacy amplification to assure high security, and this reduces the message transmission rate to \( R_m \ll 1 \).

Acknowledgments

The author thanks Hisayoshi Sato, Keisuke Hakuta, Tomohiko Uyematsu, and Masashi Ban for their insightful comments.

References

[1] A. D. Wyner, “The wire-tap channel,” Bell Syst. Tech. J., vol. 54, no. 8, pp. 1355–1387, October 1975.
[2] I. Csiszár and J. Körner, “Broadcast channels with confidential messages,” IEEE Trans. Inf. Theory, vol.IT-24, no. 3, pp. 339–348, May 1978.
[3] U. M. Maurer, “Secret key agreement by public discussion from common information,” IEEE Trans. Inf. Theory, vol. 39, no. 3, 733–742, May 1993.
[4] R. Ahlswede and I. Csiszár, “Common randomness in information theory and cryptography – part I: secret sharing,” IEEE Trans. Inf. Theory, vol. 39, no. 4, 1121–1132, July 1993.
[5] S. Venkatesan and V. Anantharam, “The common randomness capacity of a pair of independent discrete memoryless channels,” IEEE Trans. Inf. Theory, vol. 44, no. 1, pp. 215–224, January 1998.
[6] I. Csiszár and P. Narayan, “Common randomness and secret key generation with helper,” IEEE Trans. Inf. Theory, vol. 46, no. 2, pp. 344–366, March 2000.
[7] A. Khisti, S. N. Diggavi, and G. Wornell, “Secret-key generation with correlated sources and noisy channels,” IEEE International Symposium Information Theory, Toronto, Canada, pp. 1005–1009, July 2008.

[8] V. M. Prabhakaran, K. Eswaran, and K. Ramchandran, “Secrecy via sources and channels – A secret key – Secret message rate tradeoff region,” IEEE International Symposium Information Theory, Toronto, Canada, pp. 1010–1014, July 2008.

[9] H. Ahmadi and R. S.-Naini, “Secret keys from channel noise,” in Advances in Cryptology –EUROCRYPT 2011, ed. K. G. Paterson, Springer, Berlin Heidelberg, vol. 6632, pp. 266–283, May 2011.

[10] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” IEEE International conference on computers, systems and signal processing, Bangalore, India, pp. 175–179, December 1984.

[11] B. Kraus, N. Gisin, and R. Renner, “Lower and upper bounds on the secret-key rate for quantum key distribution protocols using one-way classical communication,” Phys Rev. Lett., vol. 95, no. 8, 080501, August 2005.

[12] R. Renner, N. Gisin, and B. Kraus, “Information-theoretic security proof for quantum-key-distribution protocols,” Phys. Rev. A, vo. 72, no. 1, 012332, July 2005.

[13] A. Ekert, “Quantum cryptography based on Bell’s theorem,” Phys. Rev. Lett., vol. 67, no. 6, pp. 661–663, August 1991.

[14] C. H. Bennett, G. Brassard, and N. D. Mermin, “Quantum cryptography without Bell’s theorem,” Phys. Rev. Lett., vol. 68, no. 5, pp. 557–559, February 1992.

[15] B. Qi, Y.-M. Chai, H.-K. Lo, and L. Qian, “High-speed quantum random number generation by measuring phase noise of a single-mode laser” Opt. Lett., vol. 35, no. 3, pp. 312–314 (2010).

[16] H. P. Yuen, “KCQ: A new approach to quantum cryptography I. General principles and key generation,” http://arxiv.org/abs/quant-ph/0311061v2, accessed Jun. 3.2015.

[17] G. A. Barbosa, E. Corndorf, P. Kumar, and H. P. Yuen, “Secure communication using mesoscopic coherent states,” Phys. Rev. Lett., vol. 90, no. 22, 227901, June 2003.

[18] W. Diffie and M. E. Hellman, “New directions in cryptography,” IEEE Trans. Inf. Theory vol. 22, no. 6, pp. 644 – 654, November 1976.

[19] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, “Generalized privacy amplification,” IEEE Trans. Inf. Theory, vol. 41, no. 6, pp. 1915–1923, November 1995.

[20] J. L. Carter and M. N. Wegman, “Universal class of hash functions,” J. Comput. Syst. Sci. vol. 18, no. 2, pp. 143–154, April 1979.

[21] C. H. Bennett, G. Brassard, and J. M. Robert, “Privacy amplification by public discussion,” SIAM J. Cmput., vol. 17, no. 2, pp. 210–229, April 1988.

[22] For example, W. C. Huffman and V. Pless, Fundamentals of Error-Correcting Codes, Cambridge, New York, 2003.

[23] For example, J. Justesen and T. Hoholdt, A Course in Error-Correcting Codes, European Mathematical Society, Zürich, 2004.

[24] V. Guruswami and M. Sudan, “Improved decoding of Reed–Solomon and algebraic-geometry codes,” IEEE Trans. Inf. Theory, vol. 45, no. 6, pp. 1757–1767, September 1999.

[25] Y. Wu, “New list decoding algorithms for Reed–Solomon and BCH codes,” IEEE Trans. Inf. Theory, vol. 54, no. 8, pp. 3611–3630, August 2008.

[26] E. Berlekamp, “Bounded distance +1 soft-decision Reed–Solomon decoding,” IEEE Trans. Inf. Theory vol. 42, no. 3, pp. 704–720, May 1996.

[27] N. Kamiya, “On algebraic soft-decision decoding algorithms for BCH Codes,” IEEE Trans. Inf. Theory vol. 47, no. 1, pp. 45–58, January 2001.