# On the Semantic Security in the General Bounded Storage Model: A New Proof

Mohammad Moltafet, Hamid R. Sadjadpour, and Zouheir Rezki

#### Abstract

In the bounded storage model introduced by Maurer, the adversary is computationally unbounded and has a bounded storage capacity. In this model, *information-theoretic* secrecy is guaranteed by using a publicly available random string whose length is larger than the adversary storage capacity. The protocol proposed by Maurer is simple, from the perspective of implementation, and efficient, from the perspective of the initial secret key size and random string length. However, he provided the proof of the security for the case where the adversary can access a constant fraction of the random string and store only *original bits* of the random string. In this paper, we provide a new proof of the security of the protocol proposed by Maurer for the *general* bounded storage model, i.e., the adversary can access all bits of the random string, and store the output of any Boolean function on the string. We reaffirm that the protocol is *absolutely semantically secure* in the general bounded storage model.

Index Terms- Perfect security, information-theoretic secrecy, bounded storage model.

## I. INTRODUCTION

The most effective approach of secrecy coding that provides *perfect secrecy* is the one-time pad (or Vernam cipher). Vernam introduced his cipher in 1926 [1] and Shannon in 1949 [2] proved that the one-time pad scheme provides perfect secrecy. According to the one-time pad scheme, to securely exchange a message, an independent and uniformly distributed one-time pad whose

The authors are with the Department of Electrical and Computer Engineering, University of California Santa Cruz (UCSC), Santa Cruz, CA 95064, USA (e-mail: mmoltafe@ucsc.edu, hamid@soe.ucsc.edu, zrezki@ucsc.edu).

size equals the message length needs to be established between the transmitter and receiver. More specifically, in [2], Shannon proved that perfect secrecy against an all-powerful adversary with unbounded storage capacity and unbounded computational power who has complete access to the communication line is only achievable if the uncertainty of the secret key is at least as great as that of the plaintext, this is also known as *Shannon impossibility result* [3].

In 1992, Maurer introduced the bounded storage model and proposed the first protocol in the bounded storage model that provides *information-theoretic* secrecy in the seminal work [4]. In the bounded storage model, the adversary is computationally unbounded and has a bounded storage capacity, and information-theoretic secrecy is guaranteed by using a publicly available random string whose length is larger than the adversary storage capacity and message size. In this model, the adversary performs an attack in two phases. In phase one, first, the transmitter and receiver establish a secret key, and then, the random string is broadcast. Using the shared secret key, an encryption protocol, and the random string, the transmitter and receiver compute a final key to encrypt and decrypt the message. In this phase, the adversary can compute a function on the random string and store the result. In the second phase, the random string is not available and the adversary is provided with the ciphertext, *the secret key, unbounded storage capacity*, and unbounded computational power. He tries to get information about the encrypted message using the provided information.

Security in the bounded storage model is directly related to the size of the public random string; the larger the random string, the more secure the system. Let k denote the security parameter which determines the length of the random string, m denote the message length, and n denote a large positive integer. In [4], the size of the secret key is  $k \log_2 n$ , the length of the random string is kn, and the author provided the proof of the security for the case where the adversary can access a constant fraction of the random string and store only *original bits* of the random string. Until 1997, it was an open problem to achieve information-theoretic secrecy in a *general* bounded storage model, where the adversary can access all bits of the random string. The authors of [5] provided a

protocol and proved that it provides information-theoretic secrecy in the general bounded storage model. However, the protocol is much more complicated than the protocol provided in [4] and the security results are not as efficient as desired. It requires that the transmitter and receiver transmit and store a considerable number of bits. In addition, the protocol is implemented by using multiplication in the field  $F_{2^l}$ , which is costly for the required amount of transmission. More specifically, to make sure that the probability of revealing information about the plaintext to the adversary is smaller than au, the protocol requires Alice and Bob to transmit  $3/ au^2$  bits and store  $3/\tau^2 \log |\alpha|$  bits, where  $|\alpha|$  is the length of the random string. For example, if we consider  $|\alpha| = 2^{37}$ , and we require  $\tau = 10^{-6}$ , Alice and Bob have to transmit  $3 \times 10^{12}$  bits, store  $1.11 \times 10^{14}$  bits, and the protocol requires multiplications of element in the field  $F_{2^{3 \times 10^{12}}}$ . The authors of [6] provided a simple protocol (distinct from the one introduced in [4]) that is provably secure in the general bounded storage model. The protocol exploits a secret key of length  $k \log_2 n$  and requires a random string with length mkn bits which is a considerable number of random bits compared to the protocol introduced in [4]. In [7], the authors extended the work in [6] and provided a new provably secure protocol in which the size of the random string is n which is shorter than the previous one. However, the main problem with this protocol is that the size of the secret key is  $mk \log_2 n$ , which is much longer than the message length. In [8], the authors proved that by using the protocol provided in [7], the shared secret key can be used to securely transmit an exponential number of messages against an adaptive attacker, i.e., the attacker can adaptively learn the final keys. The work in [9] is the first work that provided the proof of the security for the protocol provided in [4] in the general bounded storage model with a secret key of size  $k \log_2 n$  and a random string of size k(n + m - 1). They proved that the statistical distance between the final key and the uniform distribution is very small.

In this paper, we provide a new proof for the security of the provided protocol by Maurer in [4] in the *general* bounded storage model with a secret key of size  $k \log_2 n$  and a random string of size kn (and thus, smaller than that required in [9]). We reaffirm that the protocol is *absolutely semantically secure* in the general bounded storage model. The proof is different from (and simpler than) the approach used in [9]. The main idea behind the proof is as follows. First, we demonstrate that if the adversary is provided with all but one bit of the plaintext, the probability that he can compute the missing bit is exponentially small in the security parameter k, which is called *bit security*. Next, we establish the relationship between bit security and semantic security. Specifically, we illustrate that if the adversary can compromise the semantic security of the protocol, he can compute the missing bit, thus contradicting the bit security. In the proof of bit security, our main approach is to demonstrate that the number of strings for which an arbitrary decoding function of the adversary in Phase II can compute the missing bit is very small compared to the number of random strings resulting in the same output as in the first phase of the attack.

# A. Organization

The paper is organized as follows. The encryption and decryption protocol and main results of the paper are presented in Section II. In Section III, the proof of security in the general bounded storage model is presented. Finally, concluding remarks are made in Section IV.

# B. Notation

A random vector is denoted by a bold capital letter, whereas the corresponding underlined capital letter denotes a realization of the random vector; a random variable is denoted by a capital letter, whereas the corresponding small letter denotes a realization of the random variable; and a set is denoted by a calligraphy letter. Let  $\mathcal{G}$  be a finite set, then,  $G \stackrel{R}{\leftarrow} \mathcal{G}$  denotes choosing G uniformly at random from  $\mathcal{G}$ . All the logarithm functions in this paper have base 2.

# II. THE PROTOCOL AND MAIN RESULTS

Let  $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(k)})$ , where  $\boldsymbol{\alpha}^{(j)} \in \{0, 1\}^n$  for all  $1 \leq j \leq k$ , denote a random string with length kn and  $\boldsymbol{\alpha}^{(j)}[i]$  denote the (i + 1)th element of string  $\boldsymbol{\alpha}^{(j)}$ , i.e., the bits in string  $\boldsymbol{\alpha}^{(j)}$  are indexed from 0 to n - 1. The bits of the random string  $\boldsymbol{\alpha}$  are uniformly distributed and statistically independent. Let  $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$  denote a group with addition modulo n as the group operation which is shown by +. Next, we present the encryption protocol.

#### A. The Protocol

Suppose that Alice wants to send message  $\mathbf{M} = (M_1, \ldots, M_m) \in \{0, 1\}^m$  to Bob in the presence of an adversary whose storage capacity is bounded by  $\beta = \gamma kn$ , with  $\gamma < 1$ . Note that the message size is smaller than n. The main goal is to provide the final key  $\mathbf{X} = (X_1, \ldots, X_m) \in \{0, 1\}^m$  for encryption and decryption of the message  $\mathbf{M}$ . In this regard, first, they establish a secret key  $\mathbf{Z} = (Z_1, \ldots, Z_k)$  such that  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , and thus of size  $|\mathbf{Z}| = k \log n$  bits. Then, using the shared key  $\mathbf{Z}$ , they compute the tuple of sub-keys S containing m sub-keys, denoted as  $S = (\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m)})$ , where  $\mathbf{S}^{(i)}$  is computed as follows:

$$\mathbf{S}^{(i)} = \mathbf{Z} + (i-1)\mathbf{1}$$
  
=  $(Z_1 + (i-1), \dots, Z_k + (i-1)),$  (1)

where  $\mathbf{1} = (1, ..., 1) \in \mathbb{Z}_n^k$ . The random string  $\boldsymbol{\alpha}$  is publicly available and Alice and Bob observe it on the fly and by use of the set of sub-keys compute the  $i^{th}$  bit of the final key X, i.e.,  $X_i$  for all  $1 \le i \le m$ , as follows:

$$X_i = \bigoplus_{j=1}^k \boldsymbol{\alpha}^{(j)} [S_j^{(i)}], \qquad (2)$$

where  $S_j^{(i)} = Z_j + (i - 1)$ , i.e., the *j*th element of the sub-key  $S^{(i)}$ . Now, Alice computes  $C = M \oplus X$ , where  $\oplus$  denotes bit-wise XOR, and sends C to Bob as the ciphertext, and Bob decrypts the message by computing  $M = C \oplus X$ . The steps of the encryption and decryption procedure are summarized in Algorithm 1.

**Remark 1.** To implement the encryption and decryption protocol presented in Algorithm 1, Alice and Bob use only mk bits of the random string  $\alpha$  whose places are determined according to the set of sub-keys S, and the protocol is implemented by exploiting the simple XOR operation. 1 Message:  $\mathbf{M} = (M_1, \dots, M_m) \in \{0, 1\}^m$ , 2 Secret key:  $\mathbf{Z} = (Z_1, \dots, Z_k)$  such that  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , 3 Random string:  $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(k)}), \ \boldsymbol{\alpha}^{(j)} \in \{0, 1\}^n, \ \forall 1 \leq j \leq k$ , 4 for i = 1 to m do 5 Compute sub-key  $\mathbf{S}^{(i)}$ : 6  $\mathbf{S}^{(i)} = (Z_1 + (i - 1), \dots, Z_k + (i - 1)),$ 7 Compute the *i*th bit of the final key  $\mathbf{X}, X_i$ : 8  $X_i = \bigoplus_{j=1}^k \boldsymbol{\alpha}^{(j)} [S_j^{(i)}],$ 9 end 10 Alice and Bob set the final key  $\mathbf{X} = (X_1, \dots, X_m),$ 11 Alice encrypts  $\mathbf{C} = \mathbf{M} \oplus \mathbf{X},$  and sends  $\mathbf{C}$  to Bob, 12 Bob decrypts the message  $\mathbf{M} = \mathbf{C} \oplus \mathbf{X}.$ 

The random string  $\alpha$  is publicly available and the adversary chooses any recording function  $A_1(\alpha) : \{0,1\}^{kn} \to \{0,1\}^{\beta}$ , computes  $\zeta = A_1(\alpha)$ , and stores  $\zeta$ . We show that even if the adversary later gets the secret key Z and his computational power is unbounded, the encryption is secure. It is worth noting that the restriction on the adversary's storage is applied during the transmission of  $\alpha$ , after that there is no restriction of the adversary's storage. Without loss of generality, in this paper, results are derived for  $\gamma = 0.45$ .<sup>1</sup> Next, we present the attack model.

1) Attack Model: In the bounded storage model, formally, the adversary performs an attack in two phases as follows:

- Phase I: The random string α <sup>R</sup> {0,1}<sup>kn</sup> is broadcast. The adversary performs an arbitrary recording function A<sub>1</sub>(α) : {0,1}<sup>kn</sup> → {0,1}<sup>β</sup> on the random string α and computes ζ = A<sub>1</sub>(α). At the end of this phase, the adversary stores ζ.
- Phase II: The adversary is provided with the ciphertext C, the output of Phase I,  $\zeta$ , the secret key Z, and infinite computing power and infinite storage space. Using the provided information, the adversary tries to gain information on the message M by applying any decoding function  $A_2(\zeta, \mathbf{Z}, \mathbf{C})$ .

<sup>&</sup>lt;sup>1</sup>Similar results can be derived for any  $\gamma < 1$ .

We prove that the security of the protocol follows the absolute version of the notion of semantic security [10], i.e., the protocol is semantically secure in a system allowing a computationally unbounded adversary, and thus, the protocol is absolutely semantically secure. In other words, we prove that for any recording function  $A_1(\alpha)$  and any decoding algorithm  $A_2(\zeta, \mathbf{Z}, \mathbf{C})$ , the probability that the adversary with unbounded computational power gains even one bit of information on the message  $\mathbf{M}$  is exponentially small in the security parameter k. More specifically, consider two equiprobable messages  $\mathbf{M}^0$  and  $\mathbf{M}^1$ . One of the two messages is chosen uniformly at random, encrypted using the provided protocol, and transmitted. The adversary wishes to know which one of the messages is transmitted. We show that using any recording function  $A_1(\alpha)$  and any decoding algorithm  $A_2(\zeta, \mathbf{Z}, \mathbf{C})$  with unbounded computational power, the adversary cannot distinguish between  $\mathbf{M}^0$  and  $\mathbf{M}^1$  from the ciphertext  $\mathbf{C}$ , except with an exponentially small probability in the security parameter k. The following theorem presents the security of the protocol.

**Theorem 1.** For any two equiprobable messages  $\mathbf{M}^0$  and  $\mathbf{M}^1$  of size m, for any recording function  $A_1(\boldsymbol{\alpha}) : \{0,1\}^{kn} \to \{0,1\}^{\beta}$ , for any decoding algorithm  $A_2$ , for  $\boldsymbol{\alpha} \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , and  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , the advantage of the adversary in distinguishing between the encryption of the two messages is upper-bounded as

$$\left| \Pr\left( A_2(\boldsymbol{\zeta}, \mathbf{Z}, \mathbf{M}^1 \oplus \mathbf{X}) = 1 \right) - \Pr\left( A_2(\boldsymbol{\zeta}, \mathbf{Z}, \mathbf{M}^0 \oplus \mathbf{X}) = 1 \right) \right|$$
$$< m(2^{-k/6+1} + 2^{-0.002kn+2}). \tag{3}$$

Proof: See Section III.

**Remark 2.** As long as the shared secret key Z is not revealed to the adversary, it can be reused to transmit new messages by using new random strings. In other words, the secret key is established between Alice and Bob once and for all messages transmitted.

**Remark 3.** In practice,  $2^{-0.002kn+2}$  is negligibly small compared with  $2^{-k/6+1}$ , thus, during the

informal discussions on the security of the protocol in the bounded storage model, we drop the negligible term  $2^{-0.002kn+2}$ . For example, for the parameters  $n = 2^{45}$ ,  $m = 2^{25}$ , and k = 300 the probability of distinguishing between the encryption of the two messages is upper-bounded by  $2^{-23}$ .

## III. PROOF OF SECURITY

To prove Theorem 1, first, we prove bit security (see Proposition 1 for details) demonstrating that if the adversary is given all but the *i*th bit of the message M, the probability of correctly computing the *i*th bit of the final key X is exponentially small in the security parameter k. Subsequently, we establish the connection between bit security and semantic security. More specifically, we show that if the adversary can distinguish between messages  $M^0$  and  $M^1$ , then he can compute the *i*th missing bit of the final key, contradicting the bit security (see Section III-B for details).

Next, we provide the formal attack model for the bit security.

- Phase I: The random string α 
   <sup>R</sup> {0,1}<sup>kn</sup> is broadcast. The adversary performs an arbitrary recording function B<sub>1</sub>(α) : {0,1}<sup>kn</sup> → {0,1}<sup>β</sup> on the random string α, computes η = B<sub>1</sub>(α), and stores η.
- Phase II: The adversary is provided with i) all but the *i*th bits of the final key X, denoted as X<sup>-i</sup>, ii) the output of Phase I, η, iii) the secret key Z, and iv) infinite computing power and infinite storage space. Using the provided information, the adversary tries to compute the *i*th missing bit of the final key X, using any decoding algorithm B<sub>2</sub>(η, Z, X<sup>-i</sup>).

Let us show the *i*th bit of the final key X as  $Z(i, \alpha)$ , i.e.,  $Z(i, \alpha) \triangleq X_i$ . This notation shows that the *i*th bit of the final key X is calculated by using the random string  $\alpha$  and the secret key Z. Then, the bit security of the protocol is presented in the following proposition.

**Proposition 1** (Bit security). For any recording function  $B_1(\alpha) : \{0,1\}^{kn} \to \{0,1\}^{\beta}$ , for any

decoding algorithm  $B_2$ , for  $\boldsymbol{\alpha} \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , and  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , we have

$$\left| \Pr\left( B_2(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \right) - \frac{1}{2} \right| < 2^{-k/6} + 2^{-0.002kn+1}.$$
(4)

## A. Proof of Proposition 1

The main idea behind the proof of bit security is to show that the number of random strings  $\underline{\alpha} \in \{0,1\}^{kn}$  for which any decoding algorithm  $B_2$  can provide a desirable result is negligible compared with the number of all random strings  $\underline{\alpha}'$  for which we have  $B_1(\underline{\alpha}') = \eta$ .

**Remark 4.** It suffices to prove the theorem for the case where the recording function  $B_1$  and the decoding algorithm  $B_2$  are deterministic [7], [9]. This is because a randomized recording function is an algorithm that uses a random help string to compute its output. A randomized algorithm with a fixed help string gives rise to a deterministic algorithm [7, Remark 3, Page 5]. The same argument applies to the decoding algorithm.

Next, we present some necessary definitions and preliminary results.

**Definition 1.** Let  $K = n^k$ ,  $N = 2^{nk}$ ,  $(\underline{Z}_1, \ldots, \underline{Z}_K)$  denote an enumeration of all possible secret keys, and  $(\underline{\alpha}_1, \ldots, \underline{\alpha}_N)$  denote an enumeration of all possible random strings of length nk.

**Definition 2.** For a bit  $w \in \{0, 1\}$ , we define  $\overline{w} = (-1)^w$ , and for a vector  $\underline{W} = (w_1, \dots, w_t) \in \{0, 1\}^t$ , we define  $\overline{\underline{W}} = (\overline{w}_1, \dots, \overline{w}_t)$ .

**Definition 3.** For a string  $\underline{\alpha} \in \{0,1\}^{nk}$ , we define  $\underline{\nu}(i,\underline{\alpha}) = (\overline{Z_1}(i,\underline{\alpha}), \dots, \overline{Z_K}(i,\underline{\alpha}))$ . We use the discrepancy function  $d(\underline{\nu}(i,\underline{\alpha}))$  to measure the excess of ones over zeros, or vice versa, in the vector  $\underline{\nu}(i,\underline{\alpha})$ , which is defined as

$$d(\underline{\nu}(i,\underline{\alpha})) = \bigg| \sum_{j=1}^{K} \overline{\underline{Z}_{j}(i,\underline{\alpha})} \bigg|.$$
(5)

**Lemma 1.** For any  $\underline{\alpha} \in \{0,1\}^{nk}$ , such that neither the fraction of ones nor that of zeros in  $\underline{\alpha}$ 

March 29, 2024

is no less than 1/8,<sup>2</sup> we have

$$d(\underline{\nu}(i,\underline{\alpha})) \le K2^{-k/3}.$$
(6)

Proof: See Appendix V-A.

In the next lemma, which is derived from Lemma 1, we show that for almost all  $\underline{\alpha} \in \{0, 1\}^{nk}$ , we have  $d(\underline{\nu}(i, \underline{\alpha})) \leq K2^{-k/3}$ .

**Lemma 2.** Let  $\mathcal{D}$  denote the set of strings  $\underline{\alpha} \in \{0,1\}^{nk}$  for which  $d(\underline{\nu}(i,\underline{\alpha})) > K2^{-k/3}$ , i.e.,  $\mathcal{D} = \{\underline{\alpha} \in \{0,1\}^{nk} : d(\underline{\nu}(i,\underline{\alpha})) > K2^{-k/3}\}$ . Then, an upper-bound on the cardinality of  $\mathcal{D}$  is given as  $|\mathcal{D}| < 2^{0.544kn}$ .

Proof: See Appendix V-B.

Next, by using Lemma 2, we present a lemma that is useful in the proof of bit security.

**Lemma 3.** Let  $\mathbf{V}$  be the  $K \times N$  matrix whose *j*th column is  $\underline{\nu}(i, \underline{\alpha}_j)^T$ , where T denotes the transpose operation. Let  $\Delta = \mathbf{V}^T \mathbf{V}$ , and  $\delta_{j,j'}$  denote the *j*'th element of the *j*th row of matrix  $\Delta$ . Then, for each fixed *j*, the number of elements  $\delta_{j,j'}$  in the *j*th row such that  $|\delta_{j,j'}| > K2^{-k/3}$  is at most  $2^{0.544kn}$ .

*Proof:* See Appendix V-C.

In Lemma 4, we show that knowing a portion of the final key X does not provide any information about the missing part.

**Lemma 4.** For any  $\underline{Z} \in \mathbb{Z}_n^k$ , and for any  $\alpha \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , components of  $\mathbf{X} \in \{0,1\}^m$ , i.e.,  $X_1, \ldots, X_m$ , are statistically independent.

Proof: See Appendix V-D.

<sup>&</sup>lt;sup>2</sup>It is worth noting that 1/8 is an appropriate number for  $\gamma = 0.45$ , for other values of  $\gamma$  it needs to be changed. However, 1/8 is not the only suitable number for  $\gamma = 0.45$ ; one can use another appropriate value for which the only difference would be the coefficient of the security parameter k on the right-hand side of the inequality in (3).

Let  $\mathbf{X}^{-i} = (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_m)$  denote all the bits of the final key  $\mathbf{X}$  except the *i*th bit. The following lemma shows that knowing  $\mathbf{X}^{-i}$  does not provide any information on the secret key  $\mathbf{Z}$ . More specifically, the mutual information between  $\mathbf{X}^{-i}$  and  $\mathbf{Z}$  is zero, i.e.,  $I(\mathbf{Z}; \mathbf{X}^{-i}) = 0.$ 

**Lemma 5.** For any  $\alpha \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , for any  $i \in \{1, \dots, m\}$ , and for any  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , the secret key  $\mathbf{Z}$  is statistically independent of  $\mathbf{X}^{-i}$ . More specifically,  $\Pr(\mathbf{X}^{-i}|\mathbf{Z}) = \Pr(\mathbf{X}^{-i})$ .

Proof: See Appendix V-E.

Now, we define a condition under which we say the decoding algorithm  $B_2$  using the output of Phase I is good [7, Definition 7]. Then, we show that the number of random strings for which  $B_2$  is good is very small compared to the number of strings  $\underline{\alpha} \in \{0, 1\}^{nk}$  for which  $B_1(\underline{\alpha}) = \underline{\eta}$ .

**Definition 4** (Goodness of the decoding algorithm  $B_2$ ). For a string  $\underline{\alpha} \in \{0, 1\}^{nk}$ , and a fixed  $\underline{\eta} \in \{0, 1\}^{\beta}$ , derived from Phase I, we say that the decoding algorithm  $B_2$  using  $\underline{\eta}$  is good for  $\underline{\alpha}$  if for  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , we have

$$\left| \Pr\left( B_2(\underline{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \underline{\alpha}) \right) - \frac{1}{2} \right| \ge 2^{-k/6}.$$
(7)

It will turn out that an equivalent inequality as (7), which is presented in Lemma 6, is useful in the proof of bit security. Before presenting Lemma 6, we define the enumeration of all outputs of the decoding algorithm  $B_2$  for all possible secret keys in the following.

**Definition 5.** For a fixed  $\underline{\eta} \in \{0, 1\}^{\beta}$ , we define  $\underline{H}(i, \underline{\eta}) = (B_2(\underline{\eta}, \underline{Z}_1, \underline{X}^{-i}), \dots, B_2(\underline{\eta}, \underline{Z}_K, \underline{X}^{-i}))$ , *i.e., the enumeration of all outputs of the decoding algorithm*  $B_2$  *for all possible secret keys.* 

In the following lemma, by using  $\underline{H}(i, \eta)$ , we provide another form of goodness definition.

**Lemma 6.** For a string  $\underline{\alpha} \in \{0,1\}^{nk}$ , and a fixed  $\eta \in \{0,1\}^{\beta}$ , the decoding algorithm  $B_2$  using

 $\eta$  is good for  $\underline{\alpha}$  if

$$\left|\overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha})\right| \ge \frac{2K}{2^{k/6}}.$$
(8)

Proof: See Appendix V-F.

Next, we derive an upper-bound on the number of random strings  $\underline{\alpha} \in \{0, 1\}^{nk}$  for which the decoding algorithm  $B_2$  using  $\underline{\eta}$  is good. Note that here we consider all possible random strings  $\underline{\alpha} \in \{0, 1\}^{nk}$ , regardless of if  $B_1(\underline{\alpha}) = \eta$  or not.

**Lemma 7.** Let  $L_{\underline{H}(i,\eta)}$  denote the set of all random strings for which  $B_2$  using  $\eta$  is good, i.e.,

$$L_{\underline{H}(i,\underline{\eta})} = \left\{ \underline{\alpha} \in \{0,1\}^{nk} : \left| \overline{\underline{H}(i,\underline{\eta})} \underline{\nu}(i,\underline{\alpha}) \right| \ge \frac{2K}{2^{k/6}} \right\}.$$
(9)

Then, an upper-bound on the cardinality of  $L_{\underline{H}(i,\underline{\eta})}$  is given as  $|L_{\underline{H}(i,\underline{\eta})}| < 2^{0.544nk+k/3}$ .

Proof: See Appendix V-G.

The main step of the proof is to show that the number of strings for which  $B_2$  using  $\underline{\eta}$  is good, i.e.,  $|L_{\underline{H}(i,\underline{\eta})}|$ , is very small compared to the number of strings  $\underline{\alpha} \in \{0,1\}^{nk}$  for which  $B_1(\underline{\alpha}) = \underline{\eta}$ , i.e., the pre-image of  $\underline{\eta}$  under the recording function  $B_1$ . The pre-image of  $\underline{\eta}$  is given as

$$B_1^{-1}(B_1(\underline{\alpha})) = \left\{ \underline{\theta} \in \{0, 1\}^{nk} : B_1(\underline{\theta}) = B_1(\underline{\alpha}) \right\}.$$
 (10)

In the following lemma, we show that for all but a tiny fraction of strings  $\underline{\alpha} \in \{0,1\}^{nk}$ , the pre-image of  $\underline{\eta}$  under  $B_1$ , i.e.,  $B_1^{-1}(B_1(\underline{\alpha}))$ , contains at least  $2^{0.548kn}$  strings in  $\{0,1\}^{nk}$ .

**Lemma 8.** For any recording function  $B_1(\alpha) : \{0,1\}^{kn} \to \{0,1\}^{\beta}$ , for any  $\alpha \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , we have

$$\Pr\left(\left|B_1^{-1}(B_1(\boldsymbol{\alpha}))\right| < 2^{0.548kn}\right) \le 2^{-0.002kn}.$$
(11)

Proof: See Appendix V-H.

Next, by using Lemmas 7 and 8, we show that the probability that the decoding algorithm

 $B_2$  using  $B_1(\alpha)$  is good for the random string  $\alpha$ , is exponentially small in kn.

**Lemma 9.** For any recording function  $B_1(\alpha) : \{0,1\}^{kn} \to \{0,1\}^{\beta}$ , any decoding algorithm  $B_2$ , any  $\alpha \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , we have

$$\Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \ge \frac{2K}{2^{k/6}}\right) \le 2^{-0.002kn+1}.$$
(12)

Proof: See Appendix V-I.

Now, we are ready to prove Proposition 1, i.e., bit security. The probability that the adversary can compute the missing bit  $X_i$  by using the output of Phase I, i.e.,  $\eta$ , the secret key Z, and all other bits of the final key  $X^{-i}$ , is given as

$$\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha})\right) =$$

$$\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \in L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) \Pr\left(\boldsymbol{\alpha} \in L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) +$$

$$\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) \Pr\left(\boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right).$$
(13)

Next, by using (13), we derive an upper-bound and a lower-bound on the probability of computing the missing bit  $X_i$ ,  $\Pr(B_2(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}))$ .

The upper-bound is given as

$$\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha})\right) \stackrel{(a)}{\leq} \\\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) + \Pr\left(\boldsymbol{\alpha} \in L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) \stackrel{(b)}{\leq} \\ \frac{1}{2} + 2^{-k/6} + 2^{-0.002kn+1}, \tag{14}$$

where (a) follows from ignoring the probabilities  $\Pr(B_2(\eta, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \alpha) | \alpha \in L_{\mathbf{H}(i,\eta)})$ and  $\Pr(\alpha \notin L_{\mathbf{H}(i,\eta)})$  from the right-hand side of (13), and (b) follows because i) according to Lemma 9, an upper-bound on  $\Pr(\alpha \in L_{\mathbf{H}(i,\eta)})$  is given as  $\Pr(\alpha \in L_{\mathbf{H}(i,\eta)}) \leq 2^{-0.002kn+1}$ and ii) from the definition of goodness of algorithm  $B_2$ , Def. 4, when  $\alpha \notin L_{\mathbf{H}(i,\eta)}$ , we have

13

 $\left| \Pr\left( B_2(\boldsymbol{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \right) - \frac{1}{2} \right| < 2^{-k/6}, \text{ which, in turn, implies that}$  $\Pr\left( B_2(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})} \right) < \frac{1}{2} + 2^{-k/6}.$ 

The lower-bound is given as

$$\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha})\right) \stackrel{(a)}{\geq} \\\Pr\left(B_{2}(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) \Pr\left(\boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) \stackrel{(b)}{\geq} \\ \frac{1}{2} - 2^{-k/6} - 2^{-0.002kn+1}, \tag{15}$$

where (a) follows from ignoring the probabilities  $\Pr\left(B_2(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \in L_{\mathbf{H}(i, \boldsymbol{\eta})}\right)$  and  $\Pr\left(\boldsymbol{\alpha} \in L_{\mathbf{H}(i, \boldsymbol{\eta})}\right)$  from the right-hand side of (13), and (b) follows because i) according to Lemma 9, a lower-bound on  $\Pr\left(\boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right)$  is given as  $\Pr\left(\boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) \geq 1 - 2^{-0.002kn+1}$  and ii) from the definition of goodness of algorithm  $B_2$ , when  $\boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}$ , we have  $\left|\Pr\left(B_2(\boldsymbol{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha})\right) - \frac{1}{2}\right| < 2^{-k/6}$ , which, in turn, implies that

$$\Pr\left(B_2(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \middle| \boldsymbol{\alpha} \notin L_{\mathbf{H}(i, \boldsymbol{\eta})}\right) > \frac{1}{2} - 2^{-k/6}.$$

Finally, using the bounds in (14) and (15), we have

$$\left| \Pr\left( B_2(\boldsymbol{\eta}, \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha}) \right) - \frac{1}{2} \right| < 2^{-k/6} + 2^{-0.002kn+1}.$$

which completes the proof of Proposition 1.

# B. Proof of Theorem 1

Having proved the bit security, we show the relationship between the bit security and semantic security in the next lemma. More specifically, we show that if the adversary can break the semantic security of the protocol, then he can compute the ith missing bit in Phase II, which contradicts the bit security.

**Lemma 10.** For any two equiprobable messages  $M^0$  and  $M^1$  of size m, any recording function

 $A_1(\boldsymbol{\alpha}): \{0,1\}^{kn} \to \{0,1\}^{\beta}$ , any decoding algorithm  $A_2$ , for  $\boldsymbol{\alpha} \leftarrow \{0,1\}^{kn}$ , and  $\mathbf{Z} \leftarrow \mathbb{Z}_n^k$ , if

$$\left| \Pr\left( A_2(\boldsymbol{\zeta}, \mathbf{Z}, \mathbf{M}^1 \oplus \mathbf{X}) = 1 \right) - \Pr\left( A_2(\boldsymbol{\zeta}, \mathbf{Z}, \mathbf{M}^0 \oplus \mathbf{X}) = 1 \right) \right| = \epsilon,$$
(16)

then, there is an *i*, a recording function  $B_1$ , and a decoding algorithm  $B_2$  such that

$$\left|\Pr\left(B_2(B_1(\boldsymbol{\alpha}), \mathbf{Z}, \mathbf{X}^{-i}) = \mathbf{Z}(i, \boldsymbol{\alpha})\right) - \frac{1}{2}\right| \ge \frac{\epsilon}{2m}.$$
(17)

*Proof:* The proof follows similar steps as for the proof of Lemma 23 in [7].

Finally, combining (4), (16), and (17) we have

$$\left| \Pr\left( A_2(\boldsymbol{\zeta}, \mathbf{Z}, \mathbf{M}^1 \oplus \mathbf{X}) = 1 \right) - \Pr\left( A_2(\boldsymbol{\zeta}, \mathbf{Z}, \mathbf{M}^0 \oplus \mathbf{X}) = 1 \right) \right|$$
$$< m(2^{-k/6+1} + 2^{-0.002kn+2}), \tag{18}$$

which completes the proof of Theorem 1.

## **IV.** CONCLUSIONS

In this paper, we considered a general bounded storage model where the adversary can access all bits of the random string, and store the output of any Boolean function on the string. We reaffirm that the protocol provided by Maurer in [4] is absolutely semantically secure in the general bounded storage model with a secret key and a random string of efficient sizes.

The interesting future work would include leveraging the concept of the bounded storage model to develop absolutely semantically secure protocols for emerging applications in communication systems. Examples include but are not limited to cloud storage, cloud computing, and secure multi-party communication in the presence of an adversary with unbounded storage and unbounded computational power.

15

#### V. APPENDIX

# A. Proof of Lemma 1

Let p denote the fraction of ones, and p' = 1 - p denote the fraction of zeros in  $\underline{\alpha}$ , then we have

$$d(\underline{\nu}(i,\underline{\alpha})) \stackrel{(a)}{=} K \left| \Pr(\mathbf{Z}(i,\underline{\alpha}) = 0) - \Pr(\mathbf{Z}(i,\underline{\alpha}) = 1) \right|$$

$$\stackrel{(b)}{=} K |p' - p|^{k}$$

$$\stackrel{(c)}{\leq} K 2^{-k/3}, \tag{19}$$

where equality (a) follows from the fact that i)  $\Pr(\mathbf{Z}(i,\underline{\alpha}) = 0)$  ( $\Pr(\mathbf{Z}(i,\underline{\alpha}) = 1)$ ) is calculated as the ratio of the number of zeros (ones) in  $\underline{\nu}(i,\underline{\alpha})$  over the number all elements in  $\underline{\nu}(i,\underline{\alpha})$ which is K, and ii)  $d(\underline{\nu}(i,\underline{\alpha}))$  measures the excess of ones over zeros, or vice versa, in the string  $\underline{\nu}(i,\underline{\alpha})$ , and equality (b) follows because for  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ , we have

$$\Pr(\mathbf{Z}(i,\underline{\alpha}) = 1) = \sum_{j \text{ odd}} \binom{k}{j} p^j p'^{k-j},$$
  
$$\Pr(\mathbf{Z}(i,\underline{\alpha}) = 0) = \sum_{j \text{ even}} \binom{k}{j} p^j p'^{k-j},$$
(20)

thus, using the Binomial theorem [11, Page 162], we have

$$|\Pr(\mathbf{Z}(i,\underline{\alpha}) = 0) - \Pr(\mathbf{Z}(i,\underline{\alpha}) = 1)|$$

$$= \left| \sum_{j \text{ even}} \binom{k}{j} p^{j} p'^{k-j} - \sum_{j \text{ odd}} \binom{k}{j} p^{j} p'^{k-j} \right|$$

$$= |p' - p|^{k}.$$
(21)

Finally, inequality (c) follows because the fractions of ones and zeros in  $\underline{\alpha}$  are both no less than 1/8, thus,

$$|p - p'|^{k} \le \left(\frac{7}{8} - \frac{1}{8}\right)^{k}$$
  
< 2<sup>-k/3</sup>. (22)

## B. Proof of Lemma 2

Let  $o(\underline{\alpha})$  denote the number of ones in the string  $\underline{\alpha}$ . Then, according to Lemma 1, the string  $\underline{\alpha}$  belongs to  $\mathcal{D}$ , i.e.,  $\underline{\alpha} \in \mathcal{D}$ , implies that  $o(\underline{\alpha}) < \frac{nk}{8}$  or  $o(\underline{\alpha}) > \frac{7nk}{8}$ . Thus, for a random string  $\alpha$ , the probability of the event  $\alpha \in \mathcal{D}$  is upper-bounded as

$$\Pr\left(\boldsymbol{\alpha} \in \mathcal{D}\right) \le \Pr\left(o(\boldsymbol{\alpha}) < \frac{nk}{8} \text{ or } o(\boldsymbol{\alpha}) > \frac{7nk}{8}\right).$$
(23)

Using Stirling's approximation [11, Page 598], for  $\boldsymbol{\alpha} \stackrel{R}{\leftarrow} \{0,1\}^{kn}$ , we have

$$\Pr\left(o(\boldsymbol{\alpha}) < \frac{nk}{8} \text{ or } o(\boldsymbol{\alpha}) > \frac{7nk}{8}\right) \le 2^{-nk(1-h(1/8))+1}$$
$$= 2^{-0.4563kn+1}$$
$$= 2^{-0.4560kn-0.0003kn+1}$$
$$\stackrel{(a)}{\le} 2^{-0.456kn} \tag{24}$$

where h(.) is the binary entropy, i.e.,  $h(1/8) = 1/8 \log(8) + 7/8 \log(8/7)$ , and (a) follows because in practice, n is very large<sup>3</sup> such that -0.0003kn + 1 < 0. Finally, using (24), we have

$$|\mathcal{D}| = 2^{kn} \Pr\left(\boldsymbol{\alpha} \in \mathcal{D}\right)$$
$$< 2^{0.544kn}.$$

# C. Proof of Lemma 3

From the definition of  $\Delta$ , the value of  $|\delta_{j,j'}|$  is given as

$$\begin{aligned} |\delta_{j,j'}| &= |\underline{\nu}(i,\underline{\alpha}_j).\underline{\nu}(i,\underline{\alpha}_{j'})| = \\ \left| (\overline{\underline{Z}_1(i,\underline{\alpha}_j)}, \dots, \overline{\underline{Z}_K(i,\underline{\alpha}_j)}).(\overline{\underline{Z}_1(i,\underline{\alpha}_{j'})}, \dots, \overline{\underline{Z}_K(i,\underline{\alpha}_{j'})}) \right| &= \\ \left| \overline{\underline{Z}_1(i,\underline{\alpha}_j)} \ \overline{\underline{Z}_1(i,\underline{\alpha}_{j'})} + \dots + \overline{\underline{Z}_K(i,\underline{\alpha}_j)} \ \overline{\underline{Z}_K(i,\underline{\alpha}_{j'})} \right| \stackrel{(a)}{=} \\ d(\underline{\nu}(\underline{\alpha}_j \oplus \underline{\alpha}_{j'})), \end{aligned}$$
(25)

<sup>3</sup>In practice, n is greater than  $2^{45}$ .

March 29, 2024

where . represents the inner product of two vectors and (a) follows from the fact that  $\overline{\underline{Z}_{j''}(i, \underline{\alpha}_j)} \overline{\underline{Z}_{j''}(i, \underline{\alpha}_{j'})}$  can be written as

$$\overline{\underline{Z}_{j''}(i,\underline{\alpha}_{j})} \ \overline{\underline{Z}_{j''}(i,\underline{\alpha}_{j'})} = (-1)^{\underline{Z}_{j''}(i,\underline{\alpha}_{j})} (-1)^{\underline{Z}_{j''}(i,\underline{\alpha}_{j'})}$$

$$= (-1)^{\underline{Z}_{j''}(i,\underline{\alpha}_{j})\oplus\underline{Z}_{j''}(i,\underline{\alpha}_{j'})}$$

$$\stackrel{(a)}{=} (-1)^{\underline{Z}_{j''}(i,\underline{\alpha}_{j}\oplus\alpha_{j'})}$$

$$= \overline{\underline{Z}_{j''}(i,\underline{\alpha}_{j}\oplus\alpha_{j'})},$$
(26)

where (a) follows from the definition of  $\underline{Z}_{j''}(i, \underline{\alpha}_j)$  in (2). Thus, the *j*th row of matrix  $\Delta$  is

$$\left(d(\underline{\nu}(\underline{\alpha}_j \oplus \underline{\alpha}_1), d(\underline{\nu}(\underline{\alpha}_j \oplus \underline{\alpha}_2), \dots, d(\underline{\nu}(\underline{\alpha}_j \oplus \underline{\alpha}_N))\right)$$

Since the sequence of strings  $\underline{\alpha}_{j} \oplus \underline{\alpha}_{1}, \underline{\alpha}_{j} \oplus \underline{\alpha}_{2}, \dots, \underline{\alpha}_{j} \oplus \underline{\alpha}_{N}$  enumerates all possible binary strings with length kn, it follows directly from Lemma 2 that the number of elements  $\delta_{j,j'}$  in the *j*th row of  $\Delta$  such that  $|\delta_{j,j'}| > K2^{-k/3}$  is at most  $2^{0.544kn}$ .

## D. Proof of Lemma 4

Recall that  $X_i = \bigoplus_{j=1}^k \alpha^{(j)} [Z_j + i - 1]$ . We show that for any two distinct *i* and *i'*, the sets of bits of random string  $\alpha$  that are used to compute  $X_i$  and  $X_{i'}$  have no bit in common, thus, the components of **X** are computed by using different bits of the random string  $\alpha$  and consequently, they are statistically independent. By looking at the procedure of computing  $X_i$  and  $X_{i'}$ , we can see that the sets of bits of random string  $\alpha$  that are used to compute  $X_i$  and  $X_{i'}$  have no bit in common if the components of the *k*-tuple  $\mathbf{D}_{ii'} = \mathbf{S}^{(i)} - \mathbf{S}^{(i')}$  are all non-zero. Using (1), we have  $\mathbf{D}_{ii'} = (i - i')\mathbf{1}$ , which is a *k*-tuple with non-zero components for any two distinct *i* and *i'*.

## E. Proof of Lemma 5

Let  $(z_1, \ldots, z_k)$  denote an arbitrary realization of the secret key Z,  $p_1$  denote the probability that any bit in the random string  $\alpha$  is one, and  $p'_1 = 1 - p_1$  denote the probability that any bit in the random string  $\alpha$  is zero. Then, the probability  $\Pr(\mathbf{X}^{-i}|\mathbf{Z})$  is calculated as

$$\Pr(\mathbf{X}^{-i}|\mathbf{Z}) = \Pr\left(\mathbf{X}^{-i} = (x_{1}, \dots, x_{i-1}, x_{i+1}, \dots, x_{m}) | \mathbf{Z} = (z_{1}, \dots, z_{k})\right)$$

$$= \Pr\left(\bigoplus_{j=1}^{k} \alpha^{(j)} [S_{j}^{(1)}] = x_{1}, \dots, \bigoplus_{j=1}^{k} \alpha^{(j)} [S_{j}^{(m)}] = x_{m}) | \mathbf{Z} = (z_{1}, \dots, z_{k})\right)$$

$$\stackrel{(a)}{=} \Pr\left(\bigoplus_{j=1}^{k} \alpha^{(j)} [z_{j}] = x_{1}, \dots, \bigoplus_{j=1}^{k} \alpha^{(j)} [z_{j} + m - 1] = x_{m}\right)$$

$$\stackrel{(b)}{=} \prod_{l \in \{1, \dots, m\} \setminus \{i\}} \Pr\left(\bigoplus_{j=1}^{k} \alpha^{(j)} [z_{j} + l - 1] = x_{l}\right)$$

$$= \prod_{l \in \{1, \dots, m\} \setminus \{i\}} \left(\sum_{t=0}^{\lfloor k/2 \rfloor} \left(\frac{k}{2t + x_{l}}\right) p_{1}^{2t + x_{l}} p_{1}^{(k-(2t+x_{l}))}\right)\right) \frac{1}{n^{k}}$$

$$\stackrel{(c)}{=} \sum_{\underline{Z} \in \{0, \dots, n-1\}^{k}} \left(\prod_{l \in \{1, \dots, m\} \setminus \{i\}} \left(\sum_{t=0}^{\lfloor k/2 \rfloor} \left(\frac{k}{2t + x_{l}}\right) p_{1}^{2t + x_{l}} p_{1}^{(k-(2t+x_{l}))}\right)\right) \Pr(\mathbf{Z} = \underline{Z})$$

$$= \sum_{\underline{Z} \in \{0, \dots, n-1\}^{k}} \left(\Pr(\mathbf{X}^{-i}|\mathbf{Z})\right) \Pr(\mathbf{Z} = \underline{Z})$$

$$= \Pr(\mathbf{X}^{-i}), \qquad (27)$$

where  $\lfloor . \rfloor$  represents the floor function, (a) follows from (1), (b) follows from Lemma (4), i.e., for  $l \neq l'$  the set of bits of  $\alpha$  that are used to compute  $x_l$  has no common bit with the set of bits of  $\alpha$  that are used to compute  $x_{l'}$ , and (c) follows because  $\mathbf{Z}$  is chosen uniformly at random from  $\mathbb{Z}_n^k$ , i.e.,  $\mathbf{Z} \stackrel{R}{\leftarrow} \mathbb{Z}_n^k$ .

# F. Proof of Lemma 6

First, let us suppose that  $\overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha}) \geq \frac{2K}{2^{k/6}}$ . Then, we have

$$\Pr\left(B_2(\underline{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \underline{\alpha})\right)$$

$$\stackrel{(a)}{=} \frac{\overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha}) + \frac{K - \overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha})}{2}}{K} = \frac{1}{2} + \frac{\overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha})}{2K},$$
(28)

where (a) follows because  $\underline{H}(i,\underline{\eta}).\underline{\nu}(i,\underline{\alpha})$  is equivalent to the difference between the number of zero and non-zero elements in the tuple  $\underline{H}(i,\underline{\eta}) - \underline{\nu}(i,\underline{\alpha})$ . By using  $\underline{H}(i,\underline{\eta}).\underline{\nu}(i,\underline{\alpha}) \geq \frac{2K}{2^{k/6}}$ , we have

$$\Pr\left(B_2(\underline{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \underline{\alpha})\right) - \frac{1}{2} \ge \frac{1}{2^{k/6}}.$$
(29)

Now, suppose that  $\overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha}) \leq -\frac{2K}{2^{k/6}}$ . Then, we have

$$\Pr\left(B_{2}(\underline{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \underline{\alpha})\right)$$

$$= \frac{\underline{K + \underline{H}(i, \underline{\eta})} \cdot \underline{\nu}(i, \underline{\alpha})}{\underline{2}}$$

$$= \frac{1}{2} + \frac{\underline{H}(i, \underline{\eta}) \cdot \underline{\nu}(i, \underline{\alpha})}{\underline{2K}}.$$
(30)

By using  $\overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha}) \leq -\frac{2K}{2^{k/6}}$ , we have

$$\Pr\left(B_2(\underline{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \underline{\alpha})\right) - \frac{1}{2} \le -\frac{1}{2^{k/6}}.$$
(31)

Finally, combining (29) and (31), we have  $\left| \Pr\left( B_2(\underline{\eta}, \mathbf{Z}, \underline{X}^{-i}) = \mathbf{Z}(i, \underline{\alpha}) \right) - \frac{1}{2} \right| \geq \frac{1}{2^{k/6}}$ , which completes the proof.

## G. Proof of Lemma 7

Let

$$\begin{split} L^+_{\underline{H}(i,\underline{\eta})} &= \left\{ \underline{\alpha} \in \{0,1\}^{nk} : \overline{\underline{H}(i,\underline{\eta})}.\underline{\nu}(i,\underline{\alpha}) \geq \frac{2K}{2^{k/6}} \right\},\\ L^-_{\underline{H}(i,\underline{\eta})} &= L_{\underline{H}(i,\underline{\eta})} - L^+_{\underline{H}(i,\underline{\eta})}, \end{split}$$

and  $\underline{\xi}$  be a binary row vector of length  $N = 2^{nk}$ , where  $\underline{\xi}_j = 1$  indicates that  $\underline{\alpha}_j \in L^+_{\underline{H}(i,\underline{\eta})}$ ; otherwise,  $\underline{\xi}_j = 0$ . Then, we have

$$\frac{2K}{2^{k/6}} \left| L^{+}_{\underline{H}(i,\underline{\eta})} \right| \stackrel{(a)}{\leq} \overline{\underline{H}(i,\underline{\eta})} \cdot \mathbf{V} \cdot \underline{\xi} \\
\stackrel{(b)}{\leq} \| \overline{\underline{H}(i,\underline{\eta})} \| \| \mathbf{V} \cdot \underline{\xi} \| \\
\stackrel{(c)}{\leq} \sqrt{K} \| \mathbf{V} \cdot \underline{\xi} \|,$$
(32)

where  $\|.\|$  represents the Euclidean norm, (a) follows from the definition of  $L^+_{\underline{H}(i,\underline{\eta})}$ , (b) follows from the Cauchy-Schwartz inequality, and (c) comes from the fact that the Euclidean norm of a binary vector with length K is calculated as  $\sqrt{K}$ . The remaining task is to derive an upper-bound on  $\|\mathbf{V}.\underline{\xi}\|$ . To this end, first, we find an upper-bound on  $\|\mathbf{V}.\underline{\xi}\|^2$ , which is given as

$$\begin{split} \|\mathbf{V}.\underline{\xi}\|^{2} &= \underline{\xi}^{T}.\mathbf{V}^{T}\mathbf{V}.\underline{\xi} \\ &= \sum_{j=1}^{N}\sum_{j'=1}^{N}\delta_{jj'}\underline{\xi}_{j}\underline{\xi}_{j'} \\ \stackrel{(a)}{\leq} \sum_{j=1}^{N}\sum_{j'=1}^{N}|\delta_{jj'}|\underline{\xi}_{j}\underline{\xi}_{j'} \\ &= \sum_{j=1}^{N}\underline{\xi}_{j}\left(\sum_{j':|\delta_{jj'}|>K2^{-k/3}}|\delta_{jj'}|\underline{\xi}_{j'} + \sum_{j':|\delta_{jj'}|\leq K2^{-k/3}}|\delta_{jj'}|\underline{\xi}_{j'}\right) \\ \stackrel{(b)}{\leq} \sum_{j=1}^{N}\underline{\xi}_{j}\left(\sum_{j':|\delta_{jj'}|>K2^{-k/3}}|\delta_{jj'}| + \sum_{j':|\delta_{jj'}|\leq K2^{-k/3}}|\delta_{jj'}|\underline{\xi}_{j'}\right) \\ \stackrel{(c)}{\leq} \left|L_{\underline{H}(i,\underline{\eta})}^{+}\right|\left(K2^{0.544kn} + \sum_{j':|\delta_{jj'}|\leq K2^{-k/3}}|\delta_{jj'}|\underline{\xi}_{j'}\right) \\ \stackrel{(d)}{\leq} \left|L_{\underline{H}(i,\underline{\eta})}^{+}\right|\left(K2^{0.544kn} + \left|L_{\underline{H}(i,\underline{\eta})}^{+}\right|K2^{-k/3}\right), \end{split}$$
(33)

where (a) follows because  $|\delta_{jj'}| \ge \delta_{jj'}$ , (b) follows because for each j, we take the summation over all  $|\delta_{jj'}| > K2^{-k/3}$ , regardless of if  $\underline{\xi}_{j'} = 1$  or not, (c) follow from Lemma 3 and the fact that the maximum value of  $|\delta_{jj'}|$  is K (recall that  $|\delta_{jj'}| = |\underline{\nu}(i, \underline{\alpha}_j) \cdot \underline{\nu}(i, \underline{\alpha}_{j'})|$ , where  $\underline{\nu}(i, \underline{\alpha}_j)$  and  $\underline{\nu}(i,\underline{\alpha}_{j'}) \text{ are binary vectors of length } K), \text{ and } (d) \text{ follows because for } j': |\delta_{jj'}| \leq K2^{-k/3}, \text{ we have } \sum_{j':|\delta_{jj'}| \leq K2^{-k/3}} |\delta_{jj'}| \underline{\xi}_{j'} \leq \left| L^+_{\underline{H}(i,\underline{\eta})} \right| K2^{-k/3}.$ 

Now, combining (32) and the bound in (33), we have

$$\frac{2K}{2^{k/6}} \left| L_{\underline{H}(i,\underline{\eta})}^+ \right| \le \sqrt{K} \left| L_{\underline{H}(i,\underline{\eta})}^+ \right|^{1/2} \left( K 2^{0.544kn} + \left| L_{\underline{H}(i,\underline{\eta})}^+ \right| K 2^{-k/3} \right)^{1/2}$$

Some algebra shows that

$$\left|L_{\underline{H}(i,\underline{\eta})}^{+}\right| \le \frac{2^{0.544kn+k/3}}{3}.$$
 (34)

Following the same approach used to derive the upper-bound (34), an upper-bound on  $\left|L_{\underline{H}(i,\underline{\eta})}^{-}\right|$  is given as

$$\left|L_{\underline{H}(i,\underline{\eta})}^{-}\right| \le \frac{2^{0.544kn+k/3}}{3}.$$
 (35)

Finally, using (34) and (35), an upper-bound on  $\left|L_{\underline{H}(i,\underline{\eta})}\right|$  is given as

$$\left| L_{\underline{H}(i,\underline{\eta})} \right| = \left| L_{\underline{H}(i,\underline{\eta})}^{+} \right| + \left| L_{\underline{H}(i,\underline{\eta})}^{-} \right|$$

$$\leq \frac{2}{3} 2^{0.544kn + k/3}$$

$$< 2^{0.544kn + k/3}.$$
(36)

# H. Proof of Lemma 8

The recording algorithm  $B_1$  maps the set of all binary strings  $\{0,1\}^{kn}$  into  $2^{\beta}$  disjoint subsets  $\mathcal{F}_1, \ldots, \mathcal{F}_{2^{\beta}}$ , where  $\beta = 0.45kn$ . Thus, we have

$$\Pr\left(\left|B_{1}^{-1}(B_{1}(\boldsymbol{\alpha}))\right| < 2^{0.548kn}\right)$$

$$= \frac{\left\{\underline{\alpha} : \left|B_{1}^{-1}(B_{1}(\underline{\alpha}))\right| < 2^{0.548kn}\right\}}{2^{kn}}$$

$$= \frac{\sum_{j:|\mathcal{F}_{j}| < 2^{0.548kn}} |\mathcal{F}_{j}|}{2^{kn}}$$

$$\leq \frac{2^{0.548kn}2^{0.45kn}}{2^{kn}}$$

$$= 2^{-0.002kn}.$$
(37)

# I. Proof of Lemma 9

By using the law of total probability, we have

$$\Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \geq \frac{2K}{2^{k/6}}\right)$$

$$= \Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \geq \frac{2K}{2^{k/6}}\right| |B_1^{-1}(B_1(\boldsymbol{\alpha}))| \geq 2^{0.548kn}\right)$$

$$\Pr\left(\left|B_1^{-1}(B_1(\boldsymbol{\alpha}))\right| \geq 2^{0.548kn}\right)$$

$$+ \Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \geq \frac{2K}{2^{k/6}}\right| |B_1^{-1}(B_1(\boldsymbol{\alpha}))| < 2^{0.548kn}\right)$$

$$\Pr\left(\left|B_1^{-1}(B_1(\boldsymbol{\alpha}))\right| < 2^{0.548kn}\right)$$

$$\stackrel{(a)}{\leq} \Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \geq \frac{2K}{2^{k/6}}\right| |B_1^{-1}(B_1(\boldsymbol{\alpha}))| \geq 2^{0.548kn}\right)$$

$$+ \Pr\left(\left|B_1^{-1}(B_1(\boldsymbol{\alpha}))\right| \geq 2^{0.548kn}\right)$$

$$+ \Pr\left(\left|B_1^{-1}(B_1(\boldsymbol{\alpha}))\right| < 2^{0.548kn}\right)$$

$$= 2^{-0.002kn-0.002kn+k/3} + 2^{-0.002kn}$$

$$\stackrel{(e)}{\leq} 2^{-0.002kn} + 2^{-0.002kn}$$

$$= 2^{-0.002kn+1}, \qquad (38)$$

where (a) follows from ignoring the probabilities  $\Pr\left(\left|B_1^{-1}(B_1(\boldsymbol{\alpha}))\right| \geq 2^{0.548kn}\right)$  and

$$\Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \geq \frac{2K}{2^{k/6}} \left| \left| B_1^{-1}(B_1(\boldsymbol{\alpha})) \right| < 2^{0.548kn} \right),\right.$$

(b) follows from i) using the upper-bound derived on  $\Pr(|B_1^{-1}(B_1(\alpha))| < 2^{0.548kn})$  in Lemma 8 and ii) the fact that using Lemma 7, we have

$$\Pr\left(\left|\overline{\mathbf{H}(i,\boldsymbol{\eta})}.\boldsymbol{\nu}(i,\boldsymbol{\alpha})\right| \geq \frac{2K}{2^{k/6}} \left| \left| B_1^{-1}(B_1(\boldsymbol{\alpha})) \right| \geq 2^{0.548kn} \right| \leq \frac{2^{0.544nk+k/3}}{2^{0.548kn}},$$

and (c) follows because in practice, n is sufficiently large such that -0.002kn + k/3 < 0.

March 29, 2024

#### REFERENCES

- G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," J. Amer. Inst. Elec. Eng., vol. 55, no. 2, pp. 109–115, Feb. 1926.
- [2] C. E. Shannon, "Communication theory of secrecy systems," Bell Syst. Tech. J., vol. 28, no. 4, pp. 656-715, Oct. 1949.
- [3] Y. Dodis, "Shannon impossibility revisited," in *Proc. Int. Conf. on Inf. Theoretic Security*, Montreal, QC, Canada, Aug.15– 17, 2012, pp. 100–110.
- [4] U. M. Maurer, "Conditionally-perfect secrecy and a provably-secure randomized cipher," J. of Cryptol., vol. 5, no. 1, pp. 53–66, Jan. 1992.
- [5] C. Cachin and U. Maurer, "Unconditional security against memory-bounded adversaries," in *Proc. Int. Cryptology Conf.*, Santa Barbara, California, USA, Aug. 17–21, 1997, pp. 292–306.
- [6] A. Yonatan and M. O. Rabin, "Information theoretically secure communication in the limited storage space mode," in *Proc. Int. Cryptology Conf.*, Santa Barbara, California, USA, Aug. 15–19, 1999, pp. 65–79.
- [7] Y. Aumann, Y. Z. Ding, and M. O. Rabin, "Everlasting security in the bounded storage model," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1668–1680, Jun. 2002.
- [8] Y. Z. Ding and M. O. Rabin, "Hyper-encryption and everlasting security," in Proc. Symp. Theoretical Asp. of Computer Sci. Antibes, Juan les Pins, France, Mar. 14–16, 2002, pp. 1–26.
- [9] S. Dziembowski and U. Maurer, "Tight security proofs for the bounded-storage model," in *Proc. ACM Symp. on Theory* of Computing, Quebec, Canada, May 19–21, 2002, pp. 341–350.
- [10] S. Goldwasser and S. Micali, "Probabilistic encryption," J. Comput. Syst. Sci., vol. 28, no. 2, pp. 270-299, Apr. 1984.
- [11] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*. Addison-Wesley, Second Edition, 1994.