Perfectly Secure Encryption of Individual Sequences [*]

Neri Merhav

Department of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 32000, ISRAEL
E–mail: merhav@ee.technion.ac.il

Abstract

In analogy to the well–known notion of finite–state compressibility of individual sequences, due to Lempel and Ziv, we define a similar notion of "finite–state encryptability" of an individual plaintext sequence, as the minimum asymptotic key rate that must be consumed by finite–state encrypters so as to guarantee perfect secrecy in a well–defined sense. Our main basic result is that the finite–state encryptability is equal to the finite–state compressibility for every individual sequence. This is in parallelism to Shannon's classical probabilistic counterpart result, asserting that the minimum required key rate is equal to the entropy rate of the source. However, the redundancy, defined as the gap between the upper bound (direct part) and the lower bound (converse part) in the encryption problem, turns out to decay at a different rate (in fact, much slower) than the analogous redundancy associated with the compression problem. We also extend our main theorem in several directions, allowing: (i) availability of side information (SI) at the encrypter/decrypter/eavesdropper, (ii) lossy reconstruction at the decrypter, and (iii) the combination of both lossy reconstruction and SI, in the spirit of the Wyner–Ziv problem.

Index Terms: Information–theoretic security, Shannon's cipher system, secret key, perfect secrecy, individual sequences, finite–state machine, compressibility, incremental parsing, Lempel–Ziv algorithm, side information.

1 Introduction

The paradigm of individual sequences and finite–state machines (FSMs), as an alternative to the traditional probabilistic modeling of sources and channels, has been studied and explored quite extensively in several information–theoretic problem areas, including data compression [5], [13], [14], [18], [21], [24], [26], [27], [30], source/channel simulation [9], [15], classification [29], [31],

[*]This research was supported by ISF grant no. 208/08.

prediction [2], [3], [12] [20], [22], [32], denoising [19], and even channel coding [8], [17], just to name very few representative references out of many more. On the other hand, it is fairly safe to say that the entire literature on information—theoretic security, starting from Shannon's classical work [16] and ending with some of the most recent work in this problem area (see, e.g., [4], [6], [7], [10], [23] for surveys as well as references therein), is based exclusively on the probabilistic setting.

To the best of our knowledge, the only exception to this rule is an unpublished memorandum by Ziv [25]. In that work, the plaintext source to be encrypted, using a secret key, is an individual sequence, the encrypter is a general block encoder, and the eavesdropper employs an FSM as a message discriminator. Specifically, it is postulated in [25] that the eavesdropper may have some prior knowledge about the plaintext that can be expressed in terms of the existence of some set of "acceptable messages" that constitutes the a-priori level of uncertainty (or equivocation) that the eavesdropper has concerning the plaintext message: The larger the acceptance set, the larger is the uncertainty. Next, it is assumed that there exists an FSM that can test whether a given candidate plaintext message is acceptable or not: If and only if the FSM produces the all–zero sequence in response to that message, then this message is acceptable. Perfect security is then defined as a situation where the size of the acceptance set is not reduced (and hence neither is the uncertainty) in the presence of the cryptogram. The main result in [25] is that the asymptotic key rate needed for perfectly secure encryption in that sense, cannot be smaller (up to asymptotically vanishing terms) than the Lempel–Ziv (LZ) complexity of the plaintext source [30]. This lower bound is obviously asymptotically achieved by one–time pad encryption of the bit-stream obtained by LZ data compression of the plaintext source. This is in parallelism to Shannon's classical probabilistic counterpart result, asserting that the minimum required key rate is equal to the entropy rate of the source.

In this paper, we also consider encryption of individual sequences, but our modeling approach and the definition of perfect secrecy are substantially different. Rather than assuming that the encrypter and decrypter have unlimited resources, and that it is the eavesdropper which has limited resources, modeled in terms of FSMs, in our setting, the converse is true. We adopt a model of a finite–state encrypter, which receives as inputs the plaintext stream and the secret key bitstream, and it produces a ciphertext, while the internal state variable of the FSM, that designates limited memory of the past plaintext, is evolving in response to the plaintext input. Based on this model, we define a notion of *finite–state encryptability* (in analogy to the notions of finite–state compressibility [30] and the finite–state predictability [2]), as the minimum achievable rate at which keys bits must be consumed by any finite–state encrypter in order to guarantee perfect security against an unauthorized party, while keeping the cryptogram decipherable at the legitimate receiver, which has access to the key. Our main result is that the finite–state encryptability is equal to the finite–state compressibility, similarly as in [25].

More precisely, denoting by $c(x^n)$ the number of LZ phrases associated with the plaintext $x^n = (x_1, \ldots, x_n)$, we show that number of key bits required by any encrypter with s states, normalized by n (i.e., the key rate), cannot be smaller than $[c(x^n) \log c(x^n)]/n - \delta_s(n)$, where $\delta_s(n) = O(s \log(\log n)/\sqrt{\log n})$. On the other hand, this bound is obviously essentially achievable by applying the LZ '78 algorithm [30], followed by one–time pad encryption (i.e., bit–by–bit XORing between compressed bits and key bits), since the compression ratio of the LZ '78 algorithm is also $[c(x^n) \log c(x^n)]/n$, up to vanishingly small terms. It follows then that the finite–state encryptability of every (infinite) individual sequence is equal to its finite–state compressibility.

While the idea of LZ data compression, followed by one–time padding is rather straightforward, our main result, that no finite–state encrypter can do better than that for any given individual sequence, may not be quite obvious since the operations of compression and encryption are basically different – secret key encryption need not necessarily be based on compression followed by one–time padding, definitely not if both operations are formalized in the framework of finite–state machines.

For finite sequences of length n, the difference between the upper bound (of the direct part) and the lower bound (of the converse part), which can be thought of as some notion of redundancy, is again $O(s \log(\log n)/\sqrt{\log n})$, which decays much more slowly than the corresponding redundancy in data compression [30, Theorems 1, 2], which is roughly $O((\log s)/\log n)$.

Finally, we extend our main basic theorem in two directions, first, one at a time, and then simultaneously. The first extension is in allowing availability of side information (SI) at all three parties (encrypter, legitimate decrypter and eavesdropper) or at the decrypter and the eavesdropper only. We assume that the SI sequence is an individual sequence as well. We also assume that it is the same SI that is available to all three parties in the first case or to both the legitimate decrypter and the eavesdropper, in the second case. Extensions to situations of different versions of the SI at different users is deferred to the last step, which will possess the most general scenario we study in this work. Our main result is essentially unaltered, except that the LZ complexity, $\rho_{LZ}(x^n) \stackrel{\triangle}{=} [c(x^n)\log c(x^n)]/n$, is replaced by the conditional LZ complexity given the SI, to be defined later (see also [11], [28]). Our second extension is to the case where lossy reconstruction is allowed at the legitimate receiver (first, without SI). Here the LZ complexity is replaced by a notion of "LZ rate–distortion function," $r_{LZ}(D;x^n)$, which means the smallest of LZ complexity among all sequences that are within the allowed distortion relative to the input plaintext sequence. While our framework allows randomized reconstruction sequences (that may depend on the random key), we find that at least asymptotically, there is nothing to gain from this degree of freedom, as optimum performance can be achieved by a scheme that generates deterministic reproductions. Finally, we allow both SI and lossy reconstruction at the same time. Moreover, every party might have access to a different version of the SI. The SI available to the legitimate receiver is assumed to be generated by the plaintext source via a known memoryless channel. Here we no longer characterize the performance in terms LZ complexities of sequences, but in terms of the Wyner–Ziv rate–distortion function for individual sequences using finite–state encoders and decoders [13].

It should be pointed out that throughout the entire paper, most of our emphasis is on converse theorems (lower bounds). The compatible direct parts (upper bounds) will always be attainable by a straightforward application of the suitable data compression scheme, followed by one–time padding.

The outline of the remaining part of this paper is as follows. In Section 2, we establish some notation conventions and we formally define the model and the problem. In Section 3, we assert and prove the main result. Finally, in Section 4, we extend our results in the above–mentioned directions, and we point out how exactly the proof of the basic theorem should be modified in each case in order to support our assertions.

2 Notation Conventions and Problem Formulation

We begin by establishing some notation conventions. Throughout this paper, scalar random variables (RV's) will be denoted by capital letters, their sample values will be denoted by the respective lower case letters, and their alphabets will be denoted by the respective calligraphic letters. A similar convention will apply to random vectors and their sample values, which will be denoted with same symbols superscripted by the dimension. Thus, for example, A^m (m – positive integer) will denote a random m-vector $(A_1, ..., A_m)$, and $a^m = (a_1, ..., a_m)$ is a specific vector value in $\mathcal{A}^m$, the m–th Cartesian power of $\mathcal{A}$. The notations a_i^j and A_i^j, where i and j are integers and $i \leq j$, will designate segments $(a_i, \ldots, a_j)$ and $(A_i, \ldots, A_j)$, respectively, where for $i = 1$, the subscript will be omitted (as above). For $i > j$, a_i^j (or A_i^j) will be understood as the null string.

Sources and channels will be denoted generically by the letter P or Q, subscripted by the name of the RV and its conditioning, if applicable, exactly like in ordinary textbook notation standards, e.g., $P_{X^m}(x^m)$ is the probability function of X^m at the point $X^m = x^m$, $P_{W|S^m}(w|s^m)$ is the conditional probability of $W = w$ given $S^m = s^m$, and so on. Whenever clear from the context, these subscripts will be omitted. Information theoretic quantities, like entropies and mutual informations, will be denoted following the usual conventions of the information theory literature, e.g., $H(K^m)$, $I(W; X^m|S^m)$, and so on.

A finite–state encrypter is defined by a sixtuplet $E = (\mathcal{X}, \mathcal{Y}, \mathcal{Z}, f, g, \Delta)$, where $\mathcal{X}$ is a finite input alphabet of size $|\mathcal{X}| = \alpha$, $\mathcal{Y}$ is a finite set of binary words, $\mathcal{Z}$ is a finite set of states, $f : \mathcal{Z} \times \mathcal{X} \times \{0,1\}^* \to \mathcal{Y}$ is the output function, $g : \mathcal{Z} \times \mathcal{X} \to \mathcal{Z}$ is the next–state function, $\Delta : \mathcal{Z} \times \mathcal{X} \to \{0, 1, 2, \ldots\}$, and $\{0,1\}^*$ is the set of all binary strings of finite length. The set $\mathcal{Y}$ is allowed to contain binary strings of various lengths, including the null word λ (whose length is zero). When two infinite sequences, $\boldsymbol{x} = x_1, x_2, \ldots$, $x_i \in \mathcal{X}$, henceforth the *plaintext sequence* (or, the source sequence), and $\boldsymbol{u} = u_1, u_2, \ldots$, $u_i \in \{0,1\}$, $i = 1, 2, \ldots$, henceforth the *key sequence*, are fed into an encrypter E, it produces an infinite output sequence $\boldsymbol{y} = y_1, y_2, \ldots$, $y_i \in \mathcal{Y}$, henceforth the *cryptogram*, while passing through an infinite sequence of states $\boldsymbol{z} = z_1, z_2, \ldots$, $z_i \in \mathcal{Z}$, according to the following recursive equations, implemented for $i = 1, 2, \ldots$

$$t_i = t_{i-1} + \Delta(z_i, x_i), \qquad t_0 \stackrel{\Delta}{=} 0 \tag{1}$$

$$k_i = (u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots, u_{t_i}) \tag{2}$$

$$y_i = f(z_i, x_i, k_i) \tag{3}$$

$$z_{i+1} = g(z_i, x_i) \tag{4}$$

where it is understood that if $\Delta(z_i, x_i) = 0$, then $k_i = \lambda$, the null word of length zero,[1] namely, no

[1] Note that the evolution of the state z_i depends only on the source inputs $\{x_i\}$, not on the key bits. The rationale

key bits are used in the i–th step. By the same token, if $y_i = \lambda$, no output is produced at this step, i.e., the system is idling and only the state evolves in response to the input. An encrypter with s states, or an s–state encrypter, E, is one with $|\mathcal{Z}| = s$. It is assumed that the plaintext sequence $\boldsymbol{x}$ is deterministic (i.e., an individual sequence), whereas the key sequence $\boldsymbol{u}$ is purely random, i.e., for every positive integer n, $P_{U^n}(u^n) = 2^{-n}$,

A few additional notation conventions will be convenient: By $f(z_1, x^n, k^n)$, we refer to the vector y^n produced by E in response to the inputs x^n and k^n when the initial state is z_1. Similarly, the notation $g(z_1, x^n)$ will mean the state z_{n+1} and $\Delta(z_1, x^n)$ will designate $\sum_{i=1}^n \Delta(z_i, x_i)$ under the same circumstances. An encrypter E is said to be *perfectly secure* if for every two positive integers n, m $(m \geq n)$ and for every $\boldsymbol{x} \in \mathcal{X}^\infty$ and $y_n^m \in \mathcal{Y}^{m-n+1}$, the probability $\Pr\{Y_n^m = y_n^m | \boldsymbol{x}\}$ is independent of $\boldsymbol{x}$.

An encrypter is referred to as *information lossless* (IL) if for every $z_1 \in \mathcal{Z}$, every sufficiently large[2] n and all $x^n \in \mathcal{X}^n$ and $k^n \in \mathcal{K}^n$, the quadruple $(z_1, k^n, f(z_1, x^n, k^n), g(z_1, x^n))$ uniquely determines x^n. It will henceforth be assumed, without loss of generality, that z_1 is a certain fixed member of $\mathcal{Z}$. Given an encrypter E and an input string x^n, the encryption key rate of x^n w.r.t. E is defined as

$$\sigma_E(x^n) \stackrel{\Delta}{=} \frac{\ell(k^n)}{n} = \frac{1}{n}\sum_{i=1}^n \ell(k_i), \tag{5}$$

where $\ell(k_i) = \Delta(z_i, x_i)$ is the length of the binary string k_i and $\ell(k^n) = \sum_{i=1}^n \ell(k_i)$ is the total length of k^n.

The set of all perfectly secure, IL encrypters $\{E\}$ with no more than s states will be denoted by $\mathcal{E}(s)$. The minimum of $\sigma_E(x^n)$ over all encrypters in $\mathcal{E}(s)$ will be denoted by $\sigma_s(x^n)$, i.e.,

$$\sigma_s(x^n) = \min_{E \in \mathcal{E}(s)} \sigma_E(x^n). \tag{6}$$

is that the role of z_i is to store past memory of the information sequence x^n, in order to take advantage of empirical correlations and repetitive patterns in that sequence, whereas memory of past key bits, which are i.i.d., is irrelevant. Nonetheless, it is possible to extend the encrypter model to have two separate state variables, one evolving with dependence on $\{x_i\}$ only (as above) and one with dependence on both $\{x_i\}$ and $\{k_i\}$, where the former state variable plays a role in the update of t_i and the latter plays a role in the output function.

[2] It should be pointed out that this definition of information losslessness is more relaxed (and hence more general) than the definition in [30]. While in [30], the requirement is imposed for *every* positive integer n, here it is required only for all sufficiently large n. Note that lack of information losslessness in the more restrictive sense of [30] is not in contradiction with the ability to reconstruct the source at the legitimate decoder. All it means is that reconstruction of x^n may require more information than just (z_1, y^n, k^n, z_{n+1}), for example, some additional data from times later than $n + 1$ may be needed.

Finally, let

$$\sigma_s(\boldsymbol{x}) = \limsup_{n\to\infty} \sigma_s(x^n), \tag{7}$$

and define the *finite–state encryptability* of $\boldsymbol{x}$ as

$$\sigma(\boldsymbol{x}) = \lim_{s\to\infty} \sigma_s(\boldsymbol{x}). \tag{8}$$

Our purpose it to characterize these quantities and to point out how they can be achieved in principle.

3 Main Result

Incremental parsing [30] of a string x^n is a sequential procedure of parsing x^n into distinct phrases, where each new parsed phrase is the shortest string that has not been encountered before as a phrase of x^n, with the possible exception of the last phrase that might be incomplete. Let $c(x^n)$ denote the number of phrases in LZ incremental parsing of x^n. The LZ complexity of x^n is defined as

$$\rho_{LZ}(x^n) \stackrel{\Delta}{=} \frac{c(x^n)\log c(x^n)}{n}. \tag{9}$$

The finite–state compressibility, $\rho(\boldsymbol{x})$, of the infinite sequence $\boldsymbol{x} = (x_1, x_2, \ldots)$ is defined, in [30], as the best compression ratio achieved by IL finite–state encoders, analogously to the above definition of finite–state encryptability. From Theorems 1, 2 and 3 of [30], it follows that $\rho_{LZ}(\boldsymbol{x}) \stackrel{\Delta}{=} \limsup_{n\to\infty} \rho_{LZ}(x^n)$ is equal to $\rho(\boldsymbol{x})$.

The following theorem establishes a lower bound on $\sigma_s(x^n)$ in terms of $\rho_{LZ}(\boldsymbol{x}^n)$ and hence a lower bound of $\sigma(\boldsymbol{x})$ in terms of $\rho(\boldsymbol{x})$.

Theorem 1 *(Converse to a coding theorem): For every x^n,*

$$\sigma_s(x^n) \geq \rho_{LZ}(x^n) - \delta_s(n), \tag{10}$$

where $\delta_s(n)$ is independent of x^n and behaves according to

$$\delta_s(n) = O\left(\frac{s\log(\log n)}{\sqrt{\log n}}\right). \tag{11}$$

Consequently, $\sigma(\boldsymbol{x}) \geq \rho(\boldsymbol{x})$.

Discussion. A few comments on Theorem 1 are in order.

1. It is readily observed that a compatible direct theorem holds, simply by applying the LZ '78 algorithm followed by one–time pad encryption of the compressed bits. The resulting key–rate needed is then upper bounded by $\frac{1}{n}[c(x^n)+1]\log[2\alpha(c(x^n)+1)]$, following [30, Theorem 2], which is, within negligible terms, equal to $\rho_{LZ}(x^n)$. Thus, $\sigma(\boldsymbol{x})=\rho(\boldsymbol{x})$.

2. Consider the difference between the upper bound pertaining to the direct part (as mentioned in item no. 1 above) and the lower bound of the converse part. The behavior of this difference is $O(\alpha s\log(\log n)/\sqrt{\log n})$. This behavior is different from the behavior of the corresponding gap in compression (Theorems 1 and 2 in [30]), which is $O([\log(2\alpha)]\log(8\alpha s^2)/\log n)$. The guaranteed convergence to optimality is therefore considerably slower in the encryption problem.

3. As will be seen in the proof of Theorem 1, $\sigma_s(x^n)$ is first lower bounded in terms of the m–th order empirical entropy associated with x^n (namely, the entropy associated with the relative frequency of non–overlapping m–blocks of x^n), where m is a large positive integer, and then this empirical entropy in turn is further lower bounded in terms of $\rho_{LZ}(x^n)$. The reason for the latter passage is to get rid of the dependence of the main term of the lower bound on the parameter m, which is arbitrary. This also helps to select the optimum growth rate of m as a function of n.

4. We already mentioned that the definition of the IL property here is somewhat more relaxed than in [30] (see footnote no. 2). Moreover, it is possible to relax this requirement even further by allowing a relatively small uncertainty in x^n given $(z_1,k^n,f(z_1,x^n,k^n),g(z_1,x^n))$ (see Subsection 4.2), at the possible cost of further slowing down the convergence of $\delta_s(n)$.

Proof. Let m divide n and consider the partition of x^n into n/m non–overlapping m–vectors $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{n/m}$, where $\boldsymbol{x}_i=x_{(i-1)m+1}^{im}$. Recall that for a given $z_{(i-1)m+1}$ and $\boldsymbol{x}_i$, the length $\boldsymbol{l}_i$ of $\boldsymbol{k}_i=k_{(i-1)m+1}^{im}$ is uniquely determined as $\boldsymbol{l}_i=\Delta(z_i,x_{(i-1)m+1}^{im})$. Let us now define a joint empirical distribution of several variables. For every $a^m\in\mathcal{X}^m$, $z,z'\in\mathcal{Z}$, and every positive integer l, let

$$P_{X^mZZ'L}(a^m,z,z',l)=\frac{m}{n}\sum_{i=1}^{n/m}1\{x_{(i-1)m+1}^{im}=a^m,z_{(i-1)m+1}=z,z_{im+1}=z',\Delta(z,a^m)=l\}. \quad (12)$$

Now, define

$$P_{K^mX^mY^mZZ'L}(\kappa^m, a^m, b^m, z, z', l) = 2^{-l} P_{X^mZZ'L}(a^m, z, z', l) \cdot 1\{b^m = f(z, a^m, \kappa^m)\} \tag{13}$$

Throughout this proof, all information measures are defined w.r.t. $P_{K^mX^mY^mZZ'L}$. Consider the following chain of equalities for the given x^n and an arbitrary encrypter $E \in \mathcal{E}(s)$:

$$\begin{aligned}
\sigma_E(x^n) &= \frac{\ell(k^n)}{n} \\
&= \frac{1}{m} \cdot \frac{m}{n} \sum_{i=1}^{n/m} \ell(k_{(i-1)m+1}^{im}) \\
&= \frac{1}{m} \cdot \frac{m}{n} \sum_{i=1}^{n/m} H(K_{(i-1)m+1}^{im}) \\
&= \frac{H(K^m|L)}{m}.
\end{aligned} \tag{14}$$

Note that the length of the key for the i-th m–block, $\boldsymbol{l}_i = \ell(\boldsymbol{k}_i) = \Delta(z_{(i-1)m+1}, x_{(i-1)m+1}^{im}) = \sum_{t=(i-1)m+1}^{im} \Delta(z_t, x_t)$, is a variable that may take on no more than $(m+1)^{\alpha s-1}$ different values,[3] and hence the same is true concerning the random variable L, and so, $H(L) \leq (\alpha s - 1)\log(m+1)$. Thus,

$$\begin{aligned}
\sigma_E(x^n) &= \frac{1}{m} H(K^m|L) \\
&= \frac{1}{m}[H(K^m) - I(K^m; L)] \\
&\geq \frac{1}{m}[H(K^m) - H(L)] \\
&\geq \frac{1}{m}[H(K^m) - (\alpha s - 1)\log(m+1)].
\end{aligned} \tag{15}$$

Now, for all large m,

$$\begin{aligned}
H(K^m) &\geq H(K^m|Y^m) \\
&\geq I(K^m; X^m|Y^m) \\
&= H(X^m|Y^m) - H(X^m|Y^m, K^m) \\
&= H(X^m) - H(X^m|Y^m, K^m)
\end{aligned}$$

[3] To see why this is true, observe that the sum that defines $\boldsymbol{l}_i$ depends on $\boldsymbol{x}_i = x_{(i-1)m+1}^{im}$ and $\boldsymbol{z}_i = z_{(i-1)m+1}^{im}$ only via the joint type class of pairs $(x, z) \in \mathcal{X} \times \mathcal{Z}$, associated with $(\boldsymbol{x}_i, \boldsymbol{z}_i)$. Thus, the number of different values that $\boldsymbol{l}_i$ may take cannot exceed the total number of such type classes, which in turn is upper bounded by $(m+1)^{\alpha s-1}$.

$$
\begin{aligned}
&\geq H(X^m) - H(X^m|Y^m,K^m,Z,Z') - I(Z,Z';X^m|Y^m,K^m) \\
&= H(X^m) - 0 - I(Z,Z';X^m|Y^m,K^m) \\
&\geq H(X^m) - H(Z,Z'|Y^m,K^m) \\
&\geq H(X^m) - 2\log s, \qquad (16)
\end{aligned}
$$

where the second equality is due to the perfect security assumption and the third equality is due to the IL property, assuming that m is sufficiently large. Thus, combining eqs. (15) and (16), we obtain

$$\sigma_E(x^n) \geq \frac{H(X^m)}{m} - \frac{2\log s}{m} - (\alpha s - 1)\cdot\frac{\log(m+1)}{m}. \qquad (17)$$

Now, the main term, $H(X^m)/m$, is nothing but the normalized m–th order empirical entropy associated with x^n. Next, as discussed earlier, we further lower bound $H(X^m)/m$ in terms of $\rho_{LZ}(x^n)$ at the (small) price of reducing the bound further by additional terms that will be shown later to be negligible. In particular, in the sequel, we prove the following inequality:

$$\frac{H(X^m)}{m} \geq \frac{c(x^n)\log c(x^n)}{n} - \frac{2m(\log\alpha+1)^2}{(1-\epsilon_n)\log n} - \frac{2m\alpha^{2m}\log\alpha}{n} - \frac{1}{m}. \qquad (18)$$

where $\epsilon_n \to 0$ as $n \to \infty$. Combining this with eq. (17), we get

$$\sigma_E(x^n) \geq \frac{c(x^n)\log c(x^n)}{n} - \delta_s(n,m) \qquad (19)$$

where

$$\delta_s(n,m) = \frac{2\log s}{m} + (\alpha s-1)\cdot\frac{\log(m+1)}{m} + \frac{2m(\log\alpha+1)^2}{(1-\epsilon_n)\log n} + \frac{2m\alpha^{2m}\log\alpha}{n} + \frac{1}{m}. \qquad (20)$$

We now have the freedom to let $m = m_n$ grow slowly enough as a function of n such that $\delta_s(n) = \delta_s(n,m_n)$ will vanish for every fixed s. By letting m_n be proportional to $\sqrt{\log n}$, $\delta_s(n)$ becomes $O(s\log(\log n)/\sqrt{\log n})$. Note that the first two terms of $\delta_s(n,m)$ come from considerations pertaining to encryption, whereas the other terms appear also in compression. The second term turns out to be the dominant one, which means that in the encryption problem we end up with slower decay of the redundancy. If we compare the difference between the upper bound and the lower bound in compression (coding them and converse in [30]), this difference is dominated by a term that is $O(([\log(2\alpha)]\log(8\alpha s^2)/\log n)$, whereas in encryption the difference is $O(\alpha s\log(\log n)/\sqrt{\log n})$, namely, a significantly slower decay rate.

It remains then to establish eq. (18). To this end, let us first recall the analogous setup of lossless compression of individual sequences using finite–state machines [30]. A q-state encoder C is defined by a quintuplet $(\Sigma, \mathcal{B}, \mathcal{X}, f, g)$, where Σ is the state set of size q, $\mathcal{B}$ is a finite set of binary words (possibly of different lengths, including the null word for idling), $\mathcal{X}$ is the finite alphabet of the source to be compressed, $f : \Sigma \times \mathcal{Y} \to \mathcal{B}$ is the encoder output function, and $g : \Sigma \times \mathcal{Y} \to \Sigma$ is the next–state function. When an input sequence $(x_1, x_2, ...)$ is fed sequentially into $C = (\Sigma, \mathcal{X}, \mathcal{B}, f, g)$, the encoder outputs a sequence of binary words $(b_1, b_2, ...)$, $b_i \in \mathcal{B}$, while going through a sequence of states $(\sigma_1, \sigma_2, ...)$, according to

$$b_i = f(\sigma_i, x_i), \quad \sigma_{i+1} = g(\sigma_i, x_i), \quad i = 1, 2, ... \tag{21}$$

where σ_i is the state of C at time instant i. A finite–state encoder C is said to be *information lossless* (IL) if for all $\sigma_i \in \Sigma$ and all $x_i^{i+j-1} \in \mathcal{X}^n$, $j \geq 1$, the triple $(\sigma_i, \sigma_{i+j}, \boldsymbol{b})$ uniquely determines x_i^{i+j-1}, where σ_{i+j} and $\boldsymbol{b} = (b_i, ..., b_{i+j-1})$ are obtained by iterating eq. (21) with initial state σ_i and x_i^{i+j-1} as input. The length function associated with C is defined as $\ell_C(x^n) = \sum_{i=1}^n \ell(b_i)$, where $\ell(b_i)$ is the length of the binary string $b_i \in \mathcal{B}$.

Consider the incremental parsing of x^n and let $c(x^n)$ be defined as above. According to [30, Theorem 1], for any q-state IL encoder and for every $x^n \in \mathcal{X}^n$, $n \geq 1$,

$$\ell_C(x^n) \geq [c(x^n) + q^2] \log \frac{c(x^n)}{4q^2}. \tag{22}$$

Consider next the Shannon code, operating on x^n by successively encoding its m–blocks, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$,$\ldots$, $\boldsymbol{x}_{n/m}$, using an arbitrary probability distribution Q. According to this code, $\boldsymbol{x}_i$ is encoded using $\lceil -\log Q(\boldsymbol{x}_i) \rceil$ bits, and so, its length function is given by

$$\begin{aligned}\ell_S(x^n) &= \sum_{i=1}^{n/m} \lceil -\log Q(\boldsymbol{x}_i) \rceil \\ &= \frac{n}{m} \sum_{a^m} P_{X^m}(a^m) \lceil -\log Q(a^m) \rceil \\ &\leq \frac{n}{m} \sum_{a^m} P_{X^m}(a^m) [-\log Q(a^m) + 1] \\ &= -\frac{n}{m} \sum_{a^m} P_{X^m}(a^m) \log Q(a^m) + \frac{n}{m}. \end{aligned} \tag{23}$$

It is easy to see that this code can be implemented by a finite–state encoder in the following manner: At the beginning of each block (t mod $m = 1$), the encoder is always at some fixed initial state σ_0. At time instant $t = (i-1)m + j$, $1 < j \leq m$, the state σ_t is defined as $x_{(i-1)m+1}^{(i-1)m+j-1}$. The encoder outputs the null string whenever $t \bmod m \neq 1$; when $t \bmod m = 1$, the encoder emits the Shannon codeword of the block just terminated. The total number of states is therefore $q = \sum_{j=0}^{m-1} \alpha^j = (\alpha^m - 1)/(\alpha - 1)$. It is also easy to see that the Shannon code is IL. For given positive integers i and j, suppose we are given σ_i, σ_{i+j}, and $(b_i, b_{i+1}, ..., b_{i+j-1})$. Then $(x_i, x_{i+1}, ..., x_{i+j-1})$ can be reconstructed as follows. If time instants i and $i+j$ fall in the same m-block then σ_{j+1} conveys full information on $(x_i, x_{i+1}, ..., x_{i+j-1})$. Otherwise, we use the following procedure: The segment from time i until the end of the current block is reconstructed by decoding the codeword emitted at the end of this block. Similarly, if there are any additional blocks that are fully contained in the segment from i to $i+j$, they can also be reconstructed by decoding. Finally, the portion of the last block until position $i+j-1$ can be recovered again from the final state.

It now follows that the length function of the Shannon code must satisfy the lower bound (22) with $q = q_m \stackrel{\Delta}{=} (\alpha^m - 1)/(\alpha - 1) \leq \alpha^m$, and so,

$$-\frac{n}{m}\sum_{a^m} P_{X^m}(a^m) \log Q(a^{ml}) + \frac{n}{m} \geq [c(x^n) + q_m^2] \log \frac{c(x^n)}{4q_m^2}. \tag{24}$$

Since this holds for every Q while the right–hand side is independent of Q, we may minimize the left–hand side w.r.t. Q and obtain

$$\begin{aligned}
\frac{n}{m}H(X^m) + \frac{n}{m} &\geq [c(x^n) + q_m^2] \log \frac{c(x^n)}{4q_m^2} \\
&\geq c(x^n) \log c(x^n) - c(x^n) \log(4q_m^2) - q_m^2 \log(4q_m^2) \\
&\geq c(x^n) \log c(x^n) - c(x^n) \log(4\alpha^{2m}) - \alpha^{2m} \log(4\alpha^{2m}) \\
&\geq c(x^n) \log c(x^n) - 2mc(x^n)(1 + \log\alpha) - 2m\alpha^{2m}(1 + \log\alpha) \\
&\geq c(x^n) \log c(x^n) - \frac{2mn(1 + \log\alpha)^2}{(1 - \epsilon_n)\log n} - 2m\alpha^{2m}(1 + \log\alpha), \qquad (25)
\end{aligned}$$

where the last inequality follows from [30, eq. (6)]. Eq. (18) is now obtained by normalizing both sides by n. This completes the proof of Theorem 1.

4 Extensions

In this section, we extend Theorem 1 in two directions, availability of SI and lossy reconstruction. As described in the Introduction, we first consider each one of these directions separately, and then jointly.

4.1 Availability of Side Information

Consider the case where SI is available at the encrypter/decrypter/eavesdropper. Suppose that, in addition to the source sequence $\boldsymbol{x}$, there is an (individual) SI sequence $\boldsymbol{s} = (s_1, s_2, \ldots)$, $s_i \in \mathcal{S}$, $i = 1, 2, \ldots$, where $\mathcal{S}$ is a finite alphabet. Let us assume first that all three parties (encoder, decoder, and eavesdropper) have access to $\boldsymbol{s}$. In the formal model definition, a few modifications are needed:

1. In eqs. (1), (3), and (4), the functions Δ, f and g should be allowed to depend on the additional argument s_i,

2. The definition of perfect security should allow conditioning on $\boldsymbol{s}$, in addition to the present conditioning on $\boldsymbol{x}$. I.e., $\Pr\{Y_n^m = y_n^m | \boldsymbol{x}, \boldsymbol{s}\}$ is independent of $\boldsymbol{x}$ for all positive integers n, m (but it is allowed to depend on $\boldsymbol{s}$).

3. In the definition of an IL encrypter, the quadruple $(z, k^n, f(z, x^n, k^n), g(z, x^n))$ should be extended to be the quintuple $(z, k^n, s^n, f(z, x^n, k^n), g(z, x^n))$.

In Theorem 1, the LZ complexity of x^n, should be replaced by the conditional LZ complexity of x^n given s^n, denoted $\rho_{LZ}(x^n|s^n)$, which is an empirical measure of conditional entropy (or conditional compressibility), that is defined as follows (see also [11], [28]): Given x^n and s^n, let us apply the incremental parsing procedure of the LZ algorithm to the sequence of pairs $((x_1, s_1), (x_2, s_2), \ldots, (x_n, s_n))$. According to this procedure, all phrases are distinct with a possible exception of the last phrase, which might be incomplete. Let $c(x^n, s^n)$ denote the number of distinct phrases. For example,[4] if

$$
\begin{aligned}
x^6 &= 0 \mid 1 \mid 0\,0 \mid 0\,1| \\
s^6 &= 0 \mid 1 \mid 0\,1 \mid 0\,1|
\end{aligned}
$$

then $c(x^6, s^6) = 4$. Let $c(s^n)$ denote the resulting number of distinct phrases of s^n, and let $\boldsymbol{s}(l)$ denote the lth distinct $\boldsymbol{s}$–phrase, $l = 1, 2, ..., c(s^n)$. In the above example, $c(s^6) = 3$. Denote by $c_l(x^n|s^n)$ the number of occurrences of $\boldsymbol{s}(l)$ in the parsing of s^n, or equivalently, the number of

[4] The same example appears in [28].

distinct $\boldsymbol{x}$-phrases that jointly appear with $\boldsymbol{s}(l)$. Clearly, $\sum_{l=1}^{c(s^n)} c_l(x^n|s^n) = c(x^n, s^n)$. In the above example, $\boldsymbol{s}(1) = 0$, $\boldsymbol{s}(2) = 1$, $\boldsymbol{s}(3) = 01$, $c_1(x^6|s^6) = c_2(x^6|s^6) = 1$, and $c_3(x^6|s^6) = 2$. Now, the conditional LZ complexity of x^n given s^n is defined as

$$\rho_{LZ}(x^n|s^n) = \frac{1}{n}\sum_{l=1}^{c(s^n)} c_l(x^n|s^n)\log c_l(x^n|s^n). \tag{26}$$

The proof of Theorem 1 extends quite straightforwardly: The definition of $P_{K^mX^mY^mZZ'L}$ should be extended to $P_{K^mS^mX^mY^mZZ'L}$ in account of the empirical distribution that includes the m–blocks of s^n. In (16), all the conditionings should include S^m in addition to all existing conditionings, resulting in the inequality

$$H(K^m) \geq H(X^m|S^m) - 2\log s. \tag{27}$$

Finally, $H(X^m|S^m)$ is further lower bounded in terms of $\rho_{LZ}(x^n|s^n)$ since the latter is essentially a lower bound on the the compression ratio of x^n given s^n using finite–state encoders (see [11, eq. (13)]). The direct is obtained by first, compressing x^n to about $n \cdot \rho_{LZ}(x^n|s^n)$ bits using the conditional parsing scheme [28, Lemma 2, eq. (A.11)] and then applying one–time pad encryption.

The same performance can be achieved even if the encrypter does not have access to s^n, by using a scheme in the spirit of Slepian–Wolf coding: Randomly assign to each member of $\mathcal{X}^n$ a bin, selected independently at random across the set $\{1, 2, \ldots, 2^{nR}\}$. The encrypter applies one–time pad to the (nR)–bit binary representation of the bin index of x^n. The decrypter, first decrypts the bin index using the key and then seeks a sequence $\hat{x}^n$ within the given bin, which satisfies $\rho_{LZ}(\hat{x}^n|s^n) < R-\epsilon$. If there is one and only one such sequence, then it becomes the decoded message, otherwise an error is declared. This scheme works, just like the ordinary SW coding scheme, because the number of $\{\hat{x}^n\}$ for which $\rho_{LZ}(\hat{x}^n|s^n) < R - \epsilon$ does not exceed $2^{n[R-\epsilon+O(\log(\log n)/\log n)]}$ [28, Lemma 2]. The weakness of this is that prior knowledge of (a tight upper bound on) $\rho_{LZ}(x^n|s^n)$ is required. If, for example, it is known that x^n is a noisy version of s^n, generated, say, by a known additive channel, then R should be essentially the entropy rate of the noise.

The case where the legitimate receiver and the eavesdropper have access to different SI's will be discussed in Subsection 4.3, where we also extend the scope to lossy reconstruction.

4.2 Lossy Reconstruction

Suppose that we are content with a lossy reconstruction, $\hat{x}^n$, at the legitimate receiver. In general, this reconstruction may be a random vector due to possible dependence on the random key bits. It is required, however, that $d(x^n, \hat{x}^n) \le nD$ with probability one, for some distortion measure d. Then, in Theorem 1, $\rho_{LZ}(x^n)$ should be replaced by the "LZ rate–distortion function" of x^n, which is defined as

$$r_{LZ}(D; x^n) \stackrel{\Delta}{=} \min_{\{\hat{x}^n:\ d(x^n,\hat{x}^n)\le nD\}} \rho_{LZ}(\hat{x}^n). \tag{28}$$

In the proof of Theorem 1, the joint distribution $P_{K^m X^m \hat{X}^m Y^m Z Z' L}$ should be defined as the expectation (w.r.t. the randomness of the key) of the m–th order empirical distribution extracted from the sequences $(k^n, x^n, \hat{x}^n, y^n)$ and the resulting states $\{z_{(i-1)m+1}\}_{i=1}^{n/m}$ and key lengths $\{\boldsymbol{l}_i\}_{i=1}^{n/m}$. The definition of the IL property can be slightly relaxed to a notion of "nearly IL" (NIL) property, which allows recovery with small uncertainty for all large enough n. In particular, we shall assume that given $w \stackrel{\Delta}{=} (z_i, k_i^{i+n}, f(z_i, x_i^{n+i-1}, k_i^{n+i-1}), g(z_i, x_i^{n+i-1}))$, $\hat{x}_i^{n+i-1}$ must lie, with probability one, in a subset $\mathcal{A}_n(w) \subset \hat{\mathcal{X}}^n$, where[5]

$$\eta_n \stackrel{\Delta}{=} \lim_{n\to\infty} \frac{1}{n} \log \max_w |\mathcal{A}_n(w)| = 0. \tag{29}$$

Perfect security should be defined as statistical independence between the cryptogram and both the source and reconstruction, i.e., the probability of any segment of $\{y_i\}$ should not depend on either $\boldsymbol{x}$ or $\hat{\boldsymbol{x}}$.

In the proof of the converse part, in eq. (16), X^m should be replaced by $\hat{X}^m$ in all places, and we get

$$H(K^m) \ge H(\hat{X}^m) - 2\log s - m\eta_m, \tag{30}$$

as $H(\hat{X}^m|Y^m, K^m, Z, Z')/m$ would be upper bounded by η_m. Then, $H(\hat{X}^m)/m$ is further lower bounded in terms of $\boldsymbol{E}\rho_{LZ}(\hat{x}^n)$, essentially in the same way as before, where here we have also used the fact that, due to the concavity of the entropy functional, $H(\hat{X}^m)$ is lower bounded by the expected m–th order conditional empirical entropy pertaining to the realizations of $\hat{x}^n$. Finally,

[5] This might be the case if unambiguous reconstruction of $\hat{x}_i^{n+i-1}$ requires additional information from times later than $t = n+i-1$. For example, if the encrypter works in blocks of fixed size m, $\hat{x}^n$ is deterministic, and $n \gg m$, then by viewing the block code as finite–state machine as before, there might be uncertainty in not more than the m last symbols of $\hat{x}^n$ in case the last block is incomplete (e.g., when m does not divide n or the n–block considered is not synchronized to the m–blocks). In this case, $|\mathcal{A}_n(w)| \le |\hat{\mathcal{X}}|^m$, which is fixed, independent of n, and so $\eta_n = O(1/n)$.

since we require $d(x^n, \hat{x}^n) \le nD$ with probability one, then $\boldsymbol{E}\rho_{LZ}(\hat{x}^n)$ is trivially further lower bounded by $r_{LZ}(D; x^n)$.

Again, the direct is obvious, and it implies that at least asymptotically, there is nothing to gain from randomizing the reconstruction: The best choice of $\hat{x}^n$ is the one with minimum LZ complexity within the sphere of radius nD around x^n. This conclusion is not obvious a–priori as one might speculate that a randomized reconstruction, depending on the key, may potentially be more secure than a deterministic one.

Note that we have not assumed anything on the distortion measure d, not even additivity. Another difference between Theorem 1 of the lossless case and its present extension to the lossy case, is that we are know longer able to characterize the rate of convergence of $\delta_s(n)$, as it depends on the rate of decay of η_m. In fact, we could have replaced the IL property we assumed in the lossless case by the NIL property there too, but again, the cost would be the loss the ability to specify the behavior of δ_n.

4.3 Lossy Reconstruction With Side Information

The simultaneous extension of Theorem 1, allowing both distortion D and SI s^n leads, with the obvious modifications, to $\min\{\rho_{LZ}(\hat{x}^n|s^n) : \; d(x^n, \hat{x}^n) \le nD\}$, whose achievability is conceptually straightforward when all parties have access to s^n, including the encrypter. But what if the encrypter does not have access to s^n?

In this case, since the setting has the nature of the Wyner–Ziv problem, we need an auxiliary random variable, and there is no longer an apparent way to continue to characterize performance in terms of LZ complexities. Instead, we will resort to statistics of m–blocks, like in [13]. Assume that s^n is generated from x^n by a known memoryless channel

$$P(x^n|s^n) = \prod_{i=1}^{n} P(x_i|s_i). \tag{31}$$

Assume further that the distortion measure is additive. Consider again the m–th order joint distribution of all RV's involved, where now the expectation is taken both w.r.t. the randomness of k^n and the randomness of s^n given x^n. Let W be an auxiliary random variable with the following properties:

(i) Given X^m, W is independent of all other random variables.

(ii) The alphabet $\mathcal{W}$ of W is of size not larger than $\alpha^m + 1$.

(iii) There exists a function $f : \mathcal{S}^m \times \mathcal{W} \to \hat{\mathcal{X}}^m$ such that $\boldsymbol{E}d(X^m, f(S^m, W)) \leq mD$, where the expectation is induced by the joint statistics of all RV's.

(iv) $H(W|K^m, Y^m, S^m, Z, Z') = 0$.

We also assume

$$H(X^m|S^m, Y^m) = H(X^m|S^m), \tag{32}$$

i.e., perfect security, which means that $X^m \to S^m \to Y^m$ is a Markov chain. Hence by construction of W, $W \to S^m \to Y^m$ is also a Markov chain, and therefore

$$H(W|S^m, Y^m) = H(W|S^m). \tag{33}$$

We argue that the minimum achievable key rate is given by

$$\liminf_{m\geq 1} \limsup_{n\to\infty} \min_{P_{W|X^m}\in\mathcal{P}(D)} \frac{H(W|S^m)}{m}, \tag{34}$$

where $\mathcal{P}(D)$ is the set of channels $\{P_{W|X^m}\}$ that satisfy conditions (i)–(iii) above.

As for the converse part, replacing X^m by an arbitrary RV W (satisfying (i)–(iv)) in all places of eq. (16), and using (iv), as well as the perfect security assumption, we end up with the inequality

$$H(K^m) \geq H(W|S^m) - 2\log s. \tag{35}$$

The lower bound on the minimum key rate is then the minimum of $H(W|S^m)/m$ over all channels $\{P_{W|X^m}\}$ and functions $\{f\}$ such that (i)-(iv) hold. This minimum is now further lower bounded by the minimum of $H(W|S^m)/m$ subject to conditions (i)–(iii), with condition (iv) being dropped.

In the direct part, given the empirical distribution of m–vectors of $\boldsymbol{x}$, P_{X^m}, and the channel $P_{S^m|X^m}$, find the optimum channel $P^*_{W|X^m}$ and the function f^* that minimize $H(W|S^m)$ subject to conditions (i)–(iii). Encode as a header the optimum channel $P^*_{W|X^m}$ with a reasonably high resolution (the length of this header is a function of m but it is transmitted only once in an n–block). Now apply the channel $P^*_{W|X^m}(\cdot|\boldsymbol{x}_i)$ to generate w_i, $i = 1, 2, \ldots, n/m$. Then, compress the sequence $w^{n/m} = (w_1, \ldots, w_{n/m})$ using Slepian–Wolf (SW) coding with $s^n = (s_1^m, s_{m+1}^{2m}, \ldots, s_{n-m+1}^n)$ as SI at the decoder. Finally, encrypt the resulting compressed bits at rate $H(W|S^m)/m$ using one–time padding. At the receiver, first decrypt Y^m using K^m and then decode W based on s^n using the compatible SW decoder. Finally, apply

$$\hat{x}_{(i-1)m+1}^{im} = f^*(w_i, s_{(i-1)m+1}^{im}), \qquad i = 1, 2, \ldots, n/m. \tag{36}$$

It should be noted that this characterization is similar to the one in [13], except that here the rate is given by the conditional entropy $H(W|S^m)/m$, whereas in [13], it is the unconditional entropy $H(W)/m$ (in the present notation). In fact, in the direct part of [13], this conditioning on S^m could have been exercised as well, using SW coding (and trivially, in the converse part too), which means that we have two different characterizations of the rate–distortion performance, one with and one without conditioning on S^m. Since these characterizations must be equivalent, this actually means that for the optimum $P_{W|X^m}$, the RV's W and S^m are essentially independent ($H(W|S^m) = H(W)$ up to negligible terms). This is not surprising, as in optimum Wyner–Ziv encoding, the compressed information and the SI must be (nearly) independent, because otherwise there is some waste on common information that is conveyed to the decoder twice and hence redundant.[6] This discussion implies then that SW coding of W w.r.t. S^m is not really necessary here either and one may replace it by ordinary unconditional entropy coding at rate $H(W)/m$ without essential loss of optimality. This essential independence also means that from the point of view of the eavesdropper, the SI is essentially useless in the sense of not not reducing the uncertainty concerning the transmission W (though it still reduces the equivocation concerning X^m). As a side remark, it is also observed that Wyner–Ziv coding in long blocks requires no binning. This point was discussed in [13].

So far we discussed the case where the eavesdropper and the legitimate receiver have access to the same SI s^n. What happens if the eavesdropper has access to SI $\tilde{s}^n$ that may be different from the SI s^n of the legitimate receiver? Here we can model the SI generation mechanism by a single–input double–output channel, i.e., $P(s^n, \tilde{s}^n|x^n) = \prod_{i=1}^n P(s_i, \tilde{s}_i|x_i)$, or we can, much more generally, keep the same single-input single–output channel, from x^n to s^n, as before and allow $\tilde{s}^n$ be an individual sequence.

[6] This independence can easily be seen also in ordinary Wyner–Ziv coding. One of the first steps in the converse to the Wyner–Ziv coding theorem (see [1, p. 583, eq. (15.301)] lower bounds the entropy of compressed message by its conditional entropy given the SI. For this inequality to be tight (which is a necessary condition to achieve the Wyner–Ziv rate–distortion function), the message and the SI should be essentially independent.

It turns out that the minimum achievable key rate remains essentially the same. As for the converse part, define the joint distribution $P_{K^m L S^m \tilde{S}^m W X^m Y^m Z Z'}$ of a given $E \in \mathcal{E}_s$ where W satisfies conditions (i)–(iv) as before, and where this time, we make the obvious extension so as to include also $\tilde{S}^m$). The difficulty here is that in the chain of inequalities that is parallel to (16), we must condition all entropies and mutual informations on S^m in order to use the IL property (iv), but then we remain with the term $H(W|S^m, Y^m)$, a quantity that we cannot further lower bound independently of the specific encryption scheme.

In order to circumvent this difficulty, the idea is to consider, in addition to the arbitrary given encrypter E, also an alternative (hypothetical) encrypter $\hat{E} \in \mathcal{E}_s$, whose output, designated by the RV $\hat{Y}^m$, is independent of all other relevant RV's and hence $H(W|S^m, \hat{Y}^m) = H(W|S^m)$, resulting in a lower bound which is independent of the specific scheme. Specifically, it is assumed that $\hat{E}$ induces the same joint distribution of $(K^m, L, X^m, S^m, \tilde{S}^m, W, Z, Z')$, but may have a different random variable $\hat{Y}^m$, which is independent of $(X^m, W, S^m, \tilde{S}^m)$. The independence of $\hat{Y}^m$ can always be achieved by defining the output of $\hat{E}$ as one–time pad encryption of a certain variable (say, a Slepian–Wolf encoded version of W) that is generated by the encrypter. Now, the modified version of (16) would be

$$\begin{aligned} H(K^m) &\geq H(K^m|\hat{Y}^m, S^m) \\ &\geq I(K^m; W|\hat{Y}^m, S^m) \\ &= H(W|\hat{Y}^m, S^m) - H(W|\hat{Y}^m, S^m, K^m) \\ &= H(W|S^m) - H(W|\hat{Y}^m, S^m, K^m) \\ &= H(W|S^m) - H(W|\hat{Y}^m, S^m, K^m, Z, Z') - I(Z, Z'; W|\hat{Y}^m, K^m, S^m) \\ &= H(W|S^m) - 0 - I(Z, Z'; W|\hat{Y}^m, K^m, S^m) \\ &\geq H(W|S^m) - 2\log s. \end{aligned} \tag{37}$$

Now, as explained earlier, the last line of this chain of inequalities, no longer depends on the alternative encrypter $\hat{E}$. It depends only on the joint distribution of X^m, S^m and W. which is induced by the original encrypter E. Then once again, the best one can do is minimize $H(W|S^m)$ (or just $H(W)$ in view of the earlier discussion) subject to the same conditions as before, namely, the minimum achievable key rate is the same as before, and hence so is the direct part. The equivocation, of course, cannot be larger than $H(X^m|\tilde{S}^m)/m$, but this maximum is achieved since Y^m of the direct part is independent of $(X^m, \tilde{S}^m)$.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Hoboken, New Jersey, U.S.A., 2006.

[2] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Trans. Inform. Theory*, vol. 38, no. 4, pp. 1258–1270, July 1992.

[3] D. Haussler, J. Kivinen, and M. K. Warmuth, "Sequential prediction of individual sequences under general loss functions," *IEEE Trans. Inform. Theory*, vol. IT–44, no. 5, pp. 1906–1925, September 1998.

[4] M. E. Hellman, "An extension of the Shannon theory approach to cryptography," *IEEE Trans. Inform. Theory*, vol. IT–23, no. 3, pp. 289–294, May 1977.

[5] J. C. Kieffer and E.-h. Yang, "Sequential codes, lossless compression of individual sequences, and Kolmogorov complexity," Technical Report 1993–3, Information Theory Research Group, University of Minnesota.

[6] A. Lempel, "Cryptology in transition," *Computing Surveys*, vol. 11, no. 4, pp. 285–303, December 1979.

[7] Y. Liang, H. V. Poor and S. Shamai (Shitz), "Information theoretic security," *Foundations and Trends in Communications and Information Theory*, vol. 5, no. 4–5 pp. 355–580. 2009.

[8] Y. Lomnitz and M. Feder, "Universal communication over modulo–additive channels with an individual noise sequence," arXiv:1012.2751v1 [cs.IT] 13 Dec 2010.

[9] A. Martín, N. Merhav, G. Seroussi, and M. J. Weinberger, "Twice–universal simulation of Markov sources and individual sequences," *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4245–4255, September 2010.

[10] J. L. Massey, "An introduction to contemporary cryptology," *Proc. IEEE*, vol. 76, vol. 5, pp. 533–549, May 1988.

[11] N. Merhav, "Universal detection of messages via finite–state channels," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 2242–2246, September 2000.

[12] N. Merhav and M. Feder, "Universal schemes for sequential decision from individual data sequences," *IEEE Trans. Inform. Theory*, vol. 39, no. 4, pp. 1280–1291, July 1993.

[13] N. Merhav and J. Ziv, "On the Wyner–Ziv problem for individual sequences," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 867–873, March 2006.

[14] A. Reani and N. Merhav, "Efficient on–line schemes for encoding individual sequences with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 57, no. 10, pp. 6860–6876, October 2011.

[15] G. Seroussi, "On universal types," *IEEE Trans. Inform. Theory*, vol. 52, no. 1, pp. 171–189, January 2006.

[16] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 27, pp. 479–523, 1948 (Part I); pp. 623–656, 1948 (Part II).

[17] O. Shayevitz and M. Feder, "Communicating using feedback over a binary channel with arbitrary noise sequence," *Proc. ISIT 2005*, pp. 1516–1520, Adelaide, Australia, September 2005.

[18] M. J. Weinberger, N. Merhav, and M. Feder, "Optimal sequential probability assignment for individual sequences," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 384-396, March 1994.

[19] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. J. Weinberger, "Universal denoising: known channel," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 5–28,January 2005.

[20] T. Weissman and N. Merhav, "Universal prediction of binary individual sequences in the presence of noise," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2151–2173, September 2001.

[21] T. Weissman and N. Merhav, "On limited–delay lossy coding and filtering of individual sequences," *IEEE Trans. Inform. Theory*, vol. 48, no. 3, pp. 721–733, March 2002.

[22] T. Weissman, N. Merhav, and A. Somekh-Baruch, "Twofold universal prediction schemes for achieving the finite–state predictability of a noisy individual binary sequence," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 1849-1866, July 2001.

[23] H. Yamamoto, "Information theory in cryptology," *IEICE Trans.*, vol. E74, no. 9, pp. 2456–2464, September 1991.

[24] J. Ziv, "Coding theorems for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT–24, no. 4, pp. 405–412, July 1978.

[25] J. Ziv, "Perfect secrecy for individual sequences," unpublished manuscript, 1978.

[26] J. Ziv, "Distortion–rate theory for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT–26, no. 2, pp. 137–143, March 1980.

[27] J. Ziv, "Fixed-rate encoding of individual sequences with side information", *IEEE Transactions on Information Theory*, vol. IT–30, no. 2, pp. 348–452, March 1984.

[28] J. Ziv, "Universal decoding for finite-state channels," *IEEE Trans. Inform. Theory*, vol. IT–31, no. 4, pp. 453–460, July 1985.

[29] J. Ziv, "Compression, tests for randomness, and estimating the statistical model of an individual sequence," *Proc. Sequences*, R. M. Capocelli Ed., New York: Springer Verlag, pp. 366–373, 1990.

[30] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inform. Theory*, vol. IT–24, no. 5, pp. 530–536, September 1978.

[31] J. Ziv and N. Merhav, "A measure of relative entropy between individual sequences with application to universal classification," *IEEE Trans. Inform. Theory*, vol. 39, no. 4, pp. 1270–1279, July 1993.

[32] J. Ziv and N. Merhav, "On context–tree prediction of individual sequences," *IEEE Trans. Inform. Theory*, vol. 53, no. 5, pp. 1860–1866, May 2007.