A Provably-Secure Strongly-Randomized Cipher

Ueli M. Maurer

Institute for Signal and Information Processing
Swiss Federal Institute of Technology
CH-8092 Zurich [1]

Abstract. Shannon's pessimistic theorem, which states that a cipher can be perfect only when the entropy of the secret key is at least as great as that of the plaintext, is relativized by the demonstration of a randomized cipher in which the secret key is short but the plaintext can be very long. This cipher is shown to be "perfect with high probability". More precisely, the enemy is unable to obtain any information about the plaintext when a certain security event occurs, and the probability of this event is shown to be arbitrarily close to one unless the enemy performs an infeasible computation. This cipher exploits the existence of a publicly-accessible string of random bits whose length is much greater than that of all the plaintext to be encrypted before the secret key and the randomizer itself are changed. Two modifications of this cipher are discussed that may lead to practical provably-secure ciphers based on either of two assumptions that appear to be novel in cryptography, viz., the (sole) assumption that the enemy's memory capacity (but not his computing power) is restricted and the assumption that an explicit function is, in a specified sense, controllably-difficult to compute, but not necessarily one-way.

[1] The author is presently with the Dept. of Computer Science, Princeton University, Princeton, NJ 08540.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 361-373, 1991.

1. Introduction

One of the most important practical and theoretical open problems in cryptography is to devise a cipher that is both provably-secure and practical. The significance of a result on provable security crucially depends on the definition of security used, on the assumptions about the enemy's knowledge and resources, and on the practicality of the cipher. Excluding approaches that are based on an unproven hypothesis such as the intractability of a certain problem (e.g., factoring), one observes that every approach to provable security that has previously been proposed is either impractical or is based on a generally unrealistic assumption about the enemy's *a priori* and/or obtainable knowledge. To list a few examples: the one-time pad [7] is, because of its large key size, impractical in most applications; perfect local randomizers [4] are based on the generally unrealistic assumption that an enemy can only obtain a small number of ciphertext bits; Wyner's wire-tap channel [8] is based on the generally unrealistic assumption that the enemy's channel is noisier than the main channel; and the Rip van Winkle cipher proposed by Massey and Ingemarsson [1,2] is completely impractical since the legitimate receiver's deciphering delay is on the order of the square of the time the enemy must spend in order to break the cipher. Finally, the result that a cascade of additive stream ciphers is at least as secure as any of its component ciphers [5] yields provably-secure ciphers only when a set of additive stream ciphers can be constructed that provably contains at least one computationally secure cipher.

In this paper, we present a new approach to provable security that was motivated by [2] and is based on the availability of a very large publicly-accessible string of random bits. The need for this public randomizer is the only (but serious) detriment to the practicality of the proposed cipher. The randomizer could, for instance, be stored on a high-density storage medium, copies of which are publicly available, or it could be broadcast by a satellite.

The enemy's computational effort needed to break the cipher is measured in terms of the number of randomizer bits that he must examine. A very general way of modeling algorithms is by execution trees, where each branch corresponds to one or more operations and where the branching points correspond to decisions to be made during the execution of the algorithm. Because every examination of a randomizer bit corresponds to a branching point, the average depth of the execution tree, which is a lower bound on the average number of operations performed, is lower bounded by the average number of examined bits.

The basic idea of our approach is to prove that, even if he uses an optimal strategy for examining randomizer bits, an enemy obtains no information in Shannon's

sense about the plaintext with probability very close to one unless he accesses a substantial fraction of all the randomizer bits. More precisely, we prove that if a certain event occurs, then the enemy's entire observation, consisting of the cryptogram and the examined randomizer bits, is statistically independent of the plaintext. The probability of this event is lower bounded by a quantity that depends on the number of bits examined by the enemy, and it is very close to one unless the enemy examines a substantial fraction (e.g., 2/3) of the entire randomizer. It is obviously impossible to prove that the number of bits that the enemy must examine is greater than the total number of randomizer bits, and thus our result is close to optimum within our framework of provable security. Note that we prove that the size of the necessary input of every algorithm breaking the cipher is infeasibly large rather than that the enemy must perform any operation on the input in addition to examining it.

Since the effort to examine a random bit is in current technology roughly equal to that required to generate one, our lower bound on the enemy's computational effort appears to be on the same order as the effort needed to generate the randomizer. Therefore, our strongly-randomized cipher is truly practical only when either an existing source of randomness can be used (for example, a deep-space radio source or the surface of the moon) or should a much easier way of generating large amounts of random data be discovered (e.g., by generating identical copies of a very complicated quasi-crystal). It is not the purpose of this paper to discuss further the technical problem of generating a huge amount of publicly available random data. Rather, our interest is in exploring the question whether provable security is possible in such a model. Note, however, that when the randomizer is broadcast before the transmission of the actual cryptogram, an enemy must store essentially the whole randomizer if his chances of receiving any information about the plaintext from the succeeding ciphertext are to be non-negligible. Therefore, the amount of random data needed to achieve an acceptable level of security, even when the enemy has infinite computing power, is only somewhat larger than the enemy's memory capacity. This "broadcast" version of our cipher may be more practical than the original one.

The results of this paper appear to be somewhat surprising for two reasons. First, they demonstrate that, although perfect secrecy can be achieved only when the entropy of the secret key is at least equal to that of the plaintext (see [6]), relaxing the notion of perfectness only slightly allows one to build a provably-secure cipher whose secret key is very short compared to the length of the plaintext. Second, although information-theoretic security usually implies that the enemy has infinite computing power, our proposed cipher is secure for an information-theoretic notion of security only when the enemy is computationally restricted.

In Section 2, our model of a cipher with public randomizer is introduced, and a particular randomized cipher is presented. After describing a general model of attacks against randomized ciphers, a proof of security of our cipher against all feasible attacks is given in Section 3. In Section 4, techniques are suggested for basing the (provable) security of ciphers on either one of two assumptions, viz., that the enemy's memory capacity is restricted or that a certain function is difficult to compute in a specified sense, but not necessarily one-way.

2. Description of the Randomized Cipher

Throughout this paper, random variables are denoted by capital letters, whereas the corresponding small letters denote specific values taken on by these random variables. Underlined capital letters or superscripted capital letters denote random vectors. Our model of a strongly-randomized cipher is as follows. As in a conventional symmetric cryptosystem, the communicating parties share a short randomly-selected secret key. The randomizer $\underline{R}$ is a binary random string of length L, whose bits can be read in a random-access manner by the legitimate parties as well as by all potential opponents, i.e., $\underline{R}$ is assumed to be publicly accessible. The cryptogram is a function of the plaintext, the secret key and the randomizer such that, given the cryptogram, the key and the randomizer, the plaintext is uniquely determined. The goal of the design of a randomized cipher is to devise an encryption transformation such that the cryptogram depends on only a few randomizer bits whose positions in turn depend on the secret key in such a manner that without the secret key it is impossible to determine any of the plaintext without examining a very large number of randomizer bits.

We now describe our specific strongly-randomized cipher. It is a binary additive stream cipher in which the plaintext $\underline{X} = [X_1, \ldots, X_N]$, the cryptogram $\underline{Y} = [Y_1, \ldots, Y_N]$ and the keystream $\underline{W} = [W_1, \ldots, W_N]$ are binary sequences of length N. The cryptogram $\underline{Y}$ is obtained by adding $\underline{X}$ and $\underline{W}$ bitwise modulo 2:

$$Y_n = X_n \oplus W_n \quad \text{for } 1 \leq n \leq N.$$

The publicly-accessible binary random string $\underline{R}$ consists of K blocks of length T and thus has total length $L = KT$ bits. These blocks are denoted by $R[k,0], \ldots, R[k,T-1]$ for $1 \leq k \leq K$, i.e., the randomizer can be viewed as a two-dimensional array of binary random variables (see Figure 1). The secret key $\underline{Z} = [Z_1, \ldots, Z_K]$, where $Z_k \in \{0, \ldots, T-1\}$ for $1 \leq k \leq K$, specifies a position within each block of $\underline{R}$, and is chosen to be uniformly distributed over the key space $S_{\underline{Z}} = \{0, \ldots, T-1\}^K$. Thus the number of bits needed to represent the key is $K \log_2 T$.

$R[1,0]$	$R[1,1]$	...	$R[1,T-1]$
$R[2,0]$	$R[2,1]$	...	$R[2,T-1]$
.	.		.
.	.		.
$R[K,0]$	$R[K,1]$	...	$R[K,T-1]$

Figure 1. The randomizer $\underline{R}$, viewed as a two-dimensional array

The keystream $\underline{W}$, which is a function of the secret key $\underline{Z}$ and the randomizer $\underline{R}$, is the bitwise modulo 2 sum of the K subsequences of length N within the randomizer starting at the positions specified by the key, where each block (row) of $\underline{R}$ is considered to be extended cyclically, i.e., the second index is reduced modulo T:

$$W_n = \sum_{k=1}^{K} R[k,\ (n-1+Z_k) \bmod T] \tag{1}$$

for $1 \leq n \leq N$, where $\sum$ denotes summation modulo 2. The sub-array of the randomizer that determines $\underline{W}$ is denoted by $R^{\underline{Z}}$ and is depicted in Figure 2. A diagram of the sending site of the cipher system is shown in Figure 3. Note that the legitimate receiver who knows the secret key needs to examine only KN of the L random bits, i.e., a very small fraction N/T of all bits when $T \gg N$ as we shall assume.

$R[1,Z_1]$	$R[1,Z_1+1]$	...	$R[1,Z_1+N-1]$
$R[2,Z_2]$	$R[2,Z_2+1]$	...	$R[2,Z_2+N-1]$
.	.		.
.	.		.
$R[K,Z_K]$	$R[K,Z_K+1]$	...	$R[K,Z_K+N-1]$

Figure 2. The sub-array $R^{\underline{Z}}$ of the randomizer $\underline{R}$ is selected by the secret key $\underline{Z}$. All second indices are to be reduced modulo T. The keystream $\underline{W} = [W_1, \ldots, W_N]$ is formed by adding the K rows of $R^{\underline{Z}}$ bitwise.

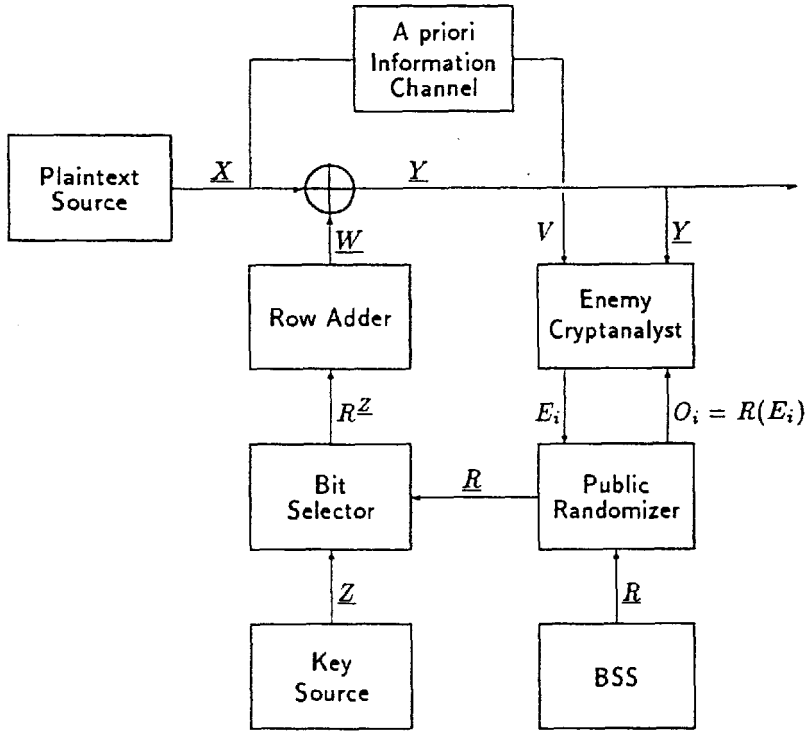

Figure 3. A block diagram of the specific strongly-randomized cipher investigated in Section 3. The public randomizer $\underline{R}$ is an array of independent and completely random binary random variables. The keystream $\underline{W}$ is formed by letting the key $\underline{Z}$ select the sub-array $R^{\underline{Z}}$ of bits of $\underline{R}$ consisting of K rows of length N, and adding these rows bitwise modulo 2. The enemy uses an arbitrary, possibly probabilistic, sequential strategy to determine the addresses $E_1, E_2, \ldots$ of the randomizer bits $O_1, O_2, \ldots$ that he examines.

3. Model of Attacks and Main Results

An enemy trying to break the cipher may have (possibly partial) knowledge of the plaintext statistics and may also have some other *a priori* information about the plaintext. Let $P_{\underline{X}}$ be the probability distribution of the plaintext and let V be a random variable, jointly distributed with $\underline{X}$ according to $P_{\underline{X}V}$, that summarizes the enemy's other *a priori* information about $\underline{X}$. Since precise knowledge of $P_{\underline{X}V}$ and thus also of $P_{\underline{X}}$ can only help the enemy and because we assume that he

precisely knows these distributions, our proof of security remains valid when the enemy actually has only partial knowledge about $P_{\underline{X}V}$.

Our model of the enemy's *attack* is described in the sequel. We allow the enemy to use an arbitrary, possibly probabilistic, sequential strategy for selecting the positions of the randomizer bits that he examines. At each step of the attack, the enemy can make use of the entire available information, i.e., the cryptogram $\underline{Y}$, the side-information V, and the positionss and values of the bits observed so far. Let $E_i = [A_i, B_i]$ denote the address of the i-th randomizer bit examined by the enemy, where A_i and B_i satisfy $1 \leq A_i \leq K$ and $0 \leq B_i \leq T-1$ for $i = 1, 2, \ldots$. Let further $O_i = R(E_i) = R[A_i, B_i]$ denote the observed value of the randomizer bit at address E_i that is examined by the enemy at the i-th step of his attack. Note that the randomizer bit O_i is a binary random variable whose address E_i is a random variable rather than a constant. However, we will make use several times of the fact that, given $E_i = e_i$, O_i corresponds to the randomizer bit $R(e_i)$ at the specific address e_i. We use the notation $E^m = [E_1, \ldots, E_m]$ and $O^m = [O_1, \ldots, O_m]$ for all $m \geq 1$. For a particular sequence $e^m = [e_1, \ldots, e_m]$ of m bit addresses, where $e_i = [a_i, b_i]$ with $1 \leq a_i \leq K$ and $0 \leq b_i \leq T-1$ for $1 \leq i \leq m$, $R(e^m) = [R(e_1), \ldots, R(e_m)]$ denotes the corresponding sequence of randomizer bits. Correspondingly, we have $O^m = R(E^m)$ for $m \geq 1$.

For $m \geq 1$, the bit position E_m is determined by the enemy as a (possibly randomized) function of the entire information he possesses at this time, i.e., the cryptogram $\underline{Y}$, the values O^{m-1} of all previously examined bits together with their addresses E^{m-1}, and the *a priori* information V. The enemy's strategy is hence completely specified by the sequence of conditional probability distributions $P_{E_1|\underline{Y}V}$, $P_{E_2|\underline{Y}VE_1O_1}$, $P_{E_3|\underline{Y}VE_1E_2O_1O_2}$, etc.. The following theorem is the main result of this paper.

Theorem: *There exists an event $\mathcal{E}$ such that, for all joint probability distributions $P_{\underline{X}V}$ and for all (possibly probabilistic) strategies for examining bits $O_1, \ldots, O_M$ of $\underline{R}$ at addresses $E_1, \ldots, E_M$,*

$$I\left(\underline{X}; \underline{Y}E^M O^M \,|\, V, \mathcal{E}\right) = 0 \quad \text{and} \quad P(\mathcal{E}) \geq 1 - N\delta^K,$$

where $\delta = M/KT$ is the fraction of randomizer bits examined by the enemy.

Here $I(\underline{X}; \underline{Y}E^M O^M \mid V, \mathcal{E})$ denotes the (mutual) information that $\underline{Y}, E^M$ and O^M together give about $\underline{X}$, given that V is known and given that the event $\mathcal{E}$ occurs. The theorem states that if the event $\mathcal{E}$ occurs, then the enemy's total observation $[\underline{Y}, E^M, O^M]$ gives no information about the plaintext $\underline{X}$ beyond the information already provided by V. Clearly, if the enemy knew the value of a

random variable V that uniquely determines $\underline{X}$, i.e., such that $H(\underline{X}|V) = 0$, it would make little sense to use a cipher at all. But the point is that, no matter what *a priori* information about the plaintext the enemy has, this does not help him to obtain any *additional* information. For instance, even if the enemy knew all but one bit of the plaintext, he would still get no information about this remaining bit if $\mathcal{E}$ occurs, and the probability of $\mathcal{E}$ could not be reduced by exploiting his virtually complete knowledge about the plaintext. Note that the theorem asserts the existence of a high-probability event $\mathcal{E}$, but does not specify it. However, in the proof we will specify such an event.

Example: Assume $K = 50$, $T = 10^{20}$ and let the plaintext be one gigabit, i.e., $N = 2^{30} \approx 10^9$. The key size of this cipher is $50 \cdot \log_2 10^{20} \approx 3320$ bits. The legitimate users need to examine only 50 randomizer bits per plaintext bit. An enemy, however, even if he used an optimal strategy for examining a fraction $\delta = 1/4$ of all bits, i.e., $M = KT/4 = 1.25 \cdot 10^{21}$ bits in total or $1.16 \cdot 10^{12}$ bits per plaintext bit, would have a chance of obtaining any new information about the plaintext not greater than $2^{30} \cdot (1/4)^{50} < 10^{-21}$.

The proof of the above theorem is divided into a sequence of four lemmas. The complete proofs of Lemmas 1 to 3 are given in [3].

Definition: The sequence $e^M = [e_1, \ldots, e_M]$ of $M \geq 1$ bit positions yields a *consistency check* for the key $\underline{z} = [z_1, \ldots, z_K]$ if and only if there exists an interger $n \in [1, N]$ and a subset $\{[1, t_1], [2, t_2], \ldots, [K, t_K]\}$ of $\{e_1, \ldots, e_M\}$ such that

$$t_k - z_k \equiv n - 1 \pmod{T} \quad \text{for } 1 \leq k \leq K.$$

In other words, e^M yields a consistency check for $\underline{z}$ if and only if $R(e^M)$ and $\underline{Z}$ together determine at least one (the n-th) bit of the keystream $\underline{W} = [W_1, \ldots, W_N]$ or, equivalently, if and only if $R(e^M)$ completely determines at least one column of $R^{\underline{Z}}$ (cf. Figure 2). Furthermore, let $\mathcal{Z}(e^M) \subseteq S_{\underline{Z}}$ denote the set of keys for which e^M yields at least one consistency check, i.e.,

$$\mathcal{Z}(e^M) = \left\{ \underline{z} \in S_{\underline{Z}} : e^M \text{ yields at least one consistency check for } \underline{z} \right\}.$$

The idea behind this definition is that if the enemy knew the plaintext (and hence also the keystream because he knows the ciphertext) and the set $R(e^M)$ of randomizer bits, then, for every key $\underline{z} \in \mathcal{Z}(e^M)$, he could perform one consistency check per keystream bit that he could compute from $R(e^M)$, by comparing the computed keystream bit for the key $\underline{z}$ with the actual keystream bit. If all computed (for key $\underline{z}$) keystream bits agree with the actual keystream bits, the key $\underline{z}$ is still a possible

candidate, but if any of the computed keystream bits differs from the corresponding actual keystream bit, then $\underline{z}$ cannot be the actual key. Note that when e^M consists of one bit in each block of $\underline{R}$, then e^M yields exactly one consistency check for N different keys. In general, if e^M consists of m_k bits in the k-th block for $1 \leq k \leq K$, then e^M yields a total number $N \prod_{k=1}^{K} m_k$ of consistency checks, but in general several of these checks will be for the same key. The event $\mathcal{E}$ introduced in the main theorem will later be defined as the event that the actual key does not belong to the set of keys for which the enemy's set E^M of observed bits yields a consistency check.

Lemma 1: *For all joint probability distributions $P_{\underline{X}V}$, for every sequence $e^M = [e_1, \ldots, e_M]$ of $M \geq 1$ bit positions, and for all $\underline{x}, v, \underline{y}, r^M \in \{0,1\}^M$ and $\underline{z} \notin \mathcal{Z}(e^M)$ such that the conditioning event has non-zero probability,*

$$P\left[\underline{X}=\underline{x} \,|\, V=v, \underline{Y}=\underline{y}, E^M=e^M, O^M=r^M, \underline{Z}=\underline{z}\right] \;=\; P\left[\underline{X}=\underline{x} | V=v\right].$$

Idea of proof: (A formal proof is given in [3].) Every bit of the keystream $\underline{W}$ is the sum of K randomizer bits (see equation (1)). The crucial observation is that when $\underline{Z} \notin \mathcal{Z}(e^M)$, then, for every n satisfying $1 \leq n \leq N$, at least one of the K randomizer bits contributing to W_n is not contained in the sequence $R(e^M)$ of randomizer bits. Therefore, given that the event $\underline{Z} \notin \mathcal{Z}(E^M)$ occurs, the keystream $\underline{W}$ is completely random and statistically independent of $\underline{X}, V, O^M = R(E^M)$ and $\underline{Z}$. Thus, also the plaintext $\underline{X}$ is statistically independent of $\underline{Y}, E^M, O^M$ and $\underline{Z}$. □

Lemma 2: *For all probability distributions $P_{\underline{X}V}$ and for all (possibly probabilistic) strategies for examining $M \geq 1$ bits $O_1, \ldots, O_M$ of $\underline{R}$ at addresses $E_1, \ldots, E_M$, we have*

$$I\left(\underline{X}; \underline{Y} E^M O^M \underline{Z} \,|\, V, \underline{Z} \notin \mathcal{Z}(E^M)\right) \;=\; 0.$$

This lemma establishes the first part of the main theorem when $\mathcal{E}$ is defined as the event that $\underline{Z} \notin \mathcal{Z}(E^M)$. It states that if the enemy does not succeed in choosing the bit positions E^M such that $\underline{Z} \in \mathcal{Z}(E^M)$, then he does not obtain any information whatsoever about the plaintext beyond the information already conveyed by V, even if an oracle would give him the key $\underline{Z}$ for free after he has finished his observation. Note that it is crucial, however, that the enemy does not know the key while selecting bits.

Proof: The conditional mutual information of Lemma 2 can be written as a difference of conditional uncertainties:

$$I\left(\underline{X};\underline{Y}E^M O^M \underline{Z} \,|\, V, \underline{Z} \notin \mathcal{Z}(E^M)\right)$$

$$= H\left(\underline{X} \,|\, V, \underline{Z} \notin \mathcal{Z}(E^M)\right) - H\left(\underline{X} \,|\, V\underline{Y}E^M O^M \underline{Z}, \underline{Z} \notin \mathcal{Z}(E^M)\right).$$

It is an immediate consequence of Lemma 1 that both uncertainties are equal. □

It remains to prove the second part of the theorem, i.e., to lower bound the probability of the event $\mathcal{E}$ that $\underline{Z} \notin \mathcal{Z}(E^M)$. Let $|S|$ denote the cardinality of the set S.

Lemma 3: *For all probability distributions $P_{\underline{X}V}$ and for all (possibly probabilistic) strategies for examining $M \geq 1$ bits $O_1, \ldots, O_M$ of $\underline{R}$ at addresses $E_1, \ldots, E_M$,*

$$P\left[\underline{Z} \notin \mathcal{Z}(E^M)\right] \geq 1 - \frac{\max_{e^M} |\mathcal{Z}(e^M)|}{T^K}.$$

Idea of proof: The proof is based on the observation that no matter which bits the enemy examines, all keys $\underline{z}$ that are not in the set $\mathcal{Z}(E^M)$ for which the enemy's sequence of observed bit positions yields a consistency check, are still equally likely candidates. More precisely, it is proved in [3] that

$$P\left[\underline{Z}=\underline{z} \,|\, \underline{Y}=\underline{y}, V=v, E^M=e^M, O^M=r^M\right] = T^{-K} \tag{2}$$

for all $e^M, \underline{y}, v, r^M$ and $\underline{z} \notin \mathcal{Z}(e^M)$. This result appears to be somewhat counter-intuitive, since it states that the *a posteriori* probabilities of the keys $\underline{z} \notin \mathcal{Z}(e^M)$ are equal to the *a priori* probabilities even when there exists a key $\underline{z} \in \mathcal{Z}(e^M)$ that satisfies many consistency checks and therefore appears to be the correct key. Equation (2) implies that

$$P[\underline{Z}=\underline{z} | E^M=e^M] = T^{-K}$$

for all e^M and $\underline{z} \notin \mathcal{Z}(e^M)$. Summing these probabilities over all keys $\underline{z} \notin \mathcal{Z}(e^M)$, i.e., over $T^K - |\mathcal{Z}(e^M)|$ terms, we obtain

$$\begin{aligned} P\left[\underline{Z} \notin \mathcal{Z}(E^M) \,|\, E^M=e^M\right] &= \sum_{\underline{z} \notin \mathcal{Z}(e^M)} P\left[\underline{Z}=\underline{z} \,|\, E^M=e^M\right] \\ &= 1 - \frac{|\mathcal{Z}(e^M)|}{T^K}. \end{aligned} \tag{3}$$

Since $P[\underline{Z} \notin \mathcal{Z}(E^M)]$ is equal to the average of $P[\underline{Z} \notin \mathcal{Z}(E^M) | E^M = e^M]$ over all values of e^M, we immediately have

$$P\left[\underline{Z} \notin \mathcal{Z}(E^M)\right] \geq 1 - \frac{\max_{e^M} |\mathcal{Z}(e^M)|}{T^K}. \quad \square$$

Equation (3) demonstrates that the enemy's optimal strategy for making the event $\mathcal{E}$ that $\underline{Z} \notin \mathcal{Z}(E^M)$ as unlikely to occur as possible is simply to make the set $\mathcal{Z}(E^M)$ as large as possible. Notice that, surprisingly, this strategy is independent of $\underline{Y}, O^M$ and V. In other words, letting the selected bit positions $E_1, \ldots, E_M$ depend on the observed bits $O_1, \ldots, O_M$, the cryptogram $\underline{Y}$ and on the *a priori* information V cannot help the enemy in reducing the probability of the event $\mathcal{E}$. However, to base the strategy on $\underline{Y}, O^M$ and V can increase the amount of information that the enemy gets about the plaintext in case that $\mathcal{E}$ does not occur, i.e., in case that $\underline{Z} \in \mathcal{Z}(E^M)$. Note that although $P[\underline{Z} \notin \mathcal{Z}(E^M) | E^M = e^M]$ equals the number of keys that are not in $\mathcal{Z}(e^M)$ divided by the total number of keys, equation (3) is non-trivial because E^M is a random variable that, because it depends on $\underline{Y}$, also depends on $\underline{Z}$.

Lemma 4: *For every sequence $e^M = [e_1, \ldots, e_M]$ of $M \geq 1$ bit positions,*

$$|\mathcal{Z}(e^M)| \leq N \left(\frac{M}{K}\right)^K .$$

Proof: Let m_k, for $1 \leq k \leq K$, be the number of randomizer bits specified by e^M that belong to the k-th block of $\underline{R}$, i.e., whose first address component is equal to k. Every subset of elements of e^M of the form $\{[1,t_1],[2,t_2],\ldots,[K,t_K]\}$ yields a consistency check for exactly N keys, namely for the keys $z = [(t_1 - x) \bmod T, (t_2 - x) \bmod T, \ldots, (t_K - x) \bmod T]$ for $0 \leq x \leq N-1$. There are exactly $\prod_{k=1}^K m_k$ different subsets of the described form and hence there are at most $N \prod_{k=1}^K m_k$ keys for which e^M yields a consistency check. $\prod_{k=1}^K m_k$ is maximized for real m_k under the restriction $\sum_{k=1}^K m_k = M$ by the choice $m_1 = \cdots = m_K = M/K$ for which $\prod_{k=1}^K m_k = (M/K)^K$. Clearly, this maximum is also an upper bound on $\prod_{k=1}^K m_k$ under the restriction that $m_1, \ldots, m_K$ must be integers satisfying $\sum_{k=1}^K m_k = M$. □

Proof of the Theorem: Lemma 2 shows that if we define $\mathcal{E}$ as the event that $\underline{Z} \notin \mathcal{Z}(E^M)$, then $I(\underline{X}; \underline{Y} E^M O^M \underline{Z} | V, \mathcal{E}) = 0$ and therefore also $I(\underline{X}; \underline{Y} E^M O^M | V, \mathcal{E}) = 0$. This last step follows from the two basic facts that mutual information is always non-negative and that giving additional random variables (here $\underline{Z}$) to the information-giving set cannot reduce the information about $\underline{X}$. Lemmas 3 and 4 finally give

$$P[\mathcal{E}] \geq 1 - \frac{\max_{e^M} |\mathcal{Z}(e^M)|}{T^K} \geq 1 - N \left(\frac{M}{KT}\right)^K = 1 - N\delta^K . \quad \square$$

4. Conclusions

In this section, we suggest two modifications of the randomized cipher presented in Section 2 that are more practical in that the size of the public randomizer required to achieve a sufficient level of security is much smaller. A rigorous proof of security for the first suggested modification would lead to the first cipher that is provably-secure under the sole assumption that the enemy's memory capacity, but not necessarily his computing power, is restricted. The second suggested modification has the potential of leading to an existence proof for secure cryptosystems without necessarily leading to a specific realization.

We first discuss a version of our strongly-randomized cipher in which instead of having the randomizer *stored* in a publicly-accessible way, it is *broadcast* by a sender (e.g., a satellite), i.e., the randomizer evolves in time rather than in space. There may exist natural sources of randomness, such as a deep-space radio source, that could be used. Alternatively, the randomizer could be transmitted as a burst of random data over the (insecure) communication channel prior to the transmission of the actual cryptogram. Because in this version of our cipher, the randomizer is not available at the time that the ciphertext is transmitted, an enemy must not only examine but also *store* a substantial fraction of the randomizer in order to be able to obtain any information about the plaintext from the ciphertext. Thus, if the enemy's memory capacity is not more than δ times the number of randomizer bits, then there exists no strategy for storing randomizer bits such that these will later be of any use to the enemy with probability more than $N\delta^K$, where N is the length of the plaintext. Note, however, that in general an enemy is not restricted to storing randomizer bits. Rather, he can store the values of boolean functions applied to the randomizer. We conjecture that a result similar to the above theorem holds even for such an extended model of the enemy's attack.

A second modification of our cipher is based on the observation that the size of the randomizer can be greatly reduced if the bit access operation can be made more difficult. In [3], a version of our cipher is discussed that is based on a function that is difficult to compute in a specified sense, but not necessarily one-way.

Finally, we would like to point out that randomization techniques similar to those presented in this paper may be useful for the construction of practical ciphers, even when the randomizer is not sufficiently long to guarantee a reasonable lower bound on the enemy's computational effort required to break the cipher or when the randomizer is replaced by a pseudo-random sequence.

Acknowledgement

I would like to thank Jim Massey for motivating this research and for many helpful discussions.

References

[1] J.L. Massey, *An introduction to contemporary cryptology*, Proceedings of the IEEE, vol. 76, no. 5, pp. 533-549, May 1988.

[2] J.L. Massey and I. Ingemarsson, *The Rip van Winkle cipher - a simple and provably computationally secure cipher with a finite key*, in IEEE Int. Symp. Info. Th., Brighton, England, (Abstracts), p. 146, June 24-28, 1985.

[3] U.M. Maurer, *Conditionally-perfect secrecy and a provably-secure randomized cipher*, to appear in Journal of Cryptology, special issue EUROCRYPT'90.

[4] U.M. Maurer and J.L. Massey, *Local randomness in pseudo-random sequences*, to appear in Journal of Cryptology, special issue CRYPTO'89.

[5] U.M. Maurer and J.L. Massey, *Cascade ciphers: the importance of being first*, presented at the 1990 IEEE Int. Symp. Inform. Theory, San Diego, CA, Jan. 14-19, 1990 (submitted to J. of Cryptology).

[6] C.E. Shannon, *Communication theory of secrecy systems*, Bell Syst. Tech. J., vol. 28, pp. 656-715, Oct. 1949.

[7] G.S. Vernam, *Cipher printing telegraph systems for secret wire and radio telegraphic communications*, J. American Inst. Elec. Eng., vol. 55, pp. 109-115, 1926.

[8] A. Wyner, *The wire-tap channel*, Bell Systems Technical Journal, vol. 54, no. 8, pp. 1355-1387, Oct. 1975.