

# Reduction of the secret key length in the perfect cipher by data compression and randomisation

Boris Ryabko

Federal Research Center for Information and Computational Technologies  
Novosibirsk State University

## Abstract

Perfect ciphers have been a very attractive cryptographic tool ever since C. Shannon described them. Note that, by definition, if a perfect cipher is used, no one can get any information about the encrypted message without knowing the secret key. We consider the problem of reducing the key length of perfect ciphers, because in many applications the length of the secret key is a crucial parameter. This paper describes a simple method of key length reduction. This method gives a perfect cipher and is based on the use of data compression and randomisation, and the average key length can be made close to Shannon entropy (which is the key length limit). It should be noted that the method can effectively use readily available data compressors (archivers).

**Keywords:** *cryptography, perfect cipher, data compression, randomisation, Shannon entropy.*

## 1 Introduction

Perfect ciphers are very attractive to cryptography, and they have found many applications since C. Shannon published his famous work [1] where he described such ciphers and proved that the so-called one-time pad (or Vernam

cipher) is perfect. The concept refers to secret-key cryptography involving three participants Alice, Bob and Eve, where Alice wants to send a message to Bob in secret from Eve, who has the ability to read all correspondence between Alice and Bob. To do this, Alice and Bob use a cipher with a secret key  $k$  (i.e. a word from some alphabet), which is known to them in advance (but not to Eve). When Alice wants to send some message  $m$ , she first encrypts  $m$  using key  $k$  and sends it to Bob, who in turn decrypts the received encrypted message using the key  $k$ . Eve also receives the encrypted message and tries to decrypt it without knowing the key. The system is called perfect, if Eve, with computers and other equipment of unlimited power and unlimited time, cannot obtain any information about the encrypted message. Not only did C. Shannon provide a formal definition of perfect (or unconditional) secrecy, but he also showed that the so-called one-time pad (or Vernam cipher) is such a system. One of the specific properties of this system is the equivalence of the length of the secret key and the message (or its entropy). Quite often this property has limited practical application as many modern telecommunication systems forward and store megabytes of information.

The natural idea is to compress messages before encryption by lossless data compression to reduce the message length and hence the secret key, and then apply a one-time pad. Note that, on the one hand, data compression before encryption is “folk wisdom” in cryptography, but on the other hand, naïve approaches to joint compression and encryption have led to well-documented security breaches in real-world systems [2].

To illustrate the non-triviality of this problem, consider a toy example. Suppose Alice wants to send Bob a message from the set  $M$ , wanting to use a one-time pad. Suppose Alice wants to save the bits of the secret key and decides to first encode the message using a data compressor and then encrypt the compressed sequence using the one-time pad and send it to Bob. Generally speaking, she will save the secret key, but Eve gets extra information (the length of the compressed message). Indeed, she knows the data compressor being used and therefore gets some information about the original message, especially for small  $M$ .

In this paper we propose a perfect cipher that uses data compression and randomisation. Schematically, the proposed cipher can be divided into three steps. Denote all possible messages by  $M$  and describe the first step as follows: the original message  $m \in M$  is compressed by some data compressor  $\phi$  (let the resulting word  $\phi(m) = x = x_1 \dots x_s$  and  $l = \max_{m \in M} |\phi(m)|$ ).

Second, the resulting word  $x$  is encrypted with a one-time pad  $\psi$  using a secret key of length  $s$  bits (let  $\psi(x) = y = y_1y_2\dots y_s$  and, third, word  $y$  is expanded into  $l - s$  random bits, which are independent and obey the Bernoulli distribution with  $p(0) = p(1) = 1/2$ . (So, the resulting word is  $y_1y_2\dots y_sr_{s+1}\dots r_l$ , where  $r_i$  are the random bits).

Thus, the goal of this cipher is to reduce the length of the secret key used and hence it makes sense to use this cipher if the value of the secret bits is much greater than the value of the random bits. It is worth noting that the proposed cipher is simple and, in fact, its complexity is determined by the data compression algorithm and/or the random bit generator. In addition, any practically usable archiver (e.g. ZIP) can be used in conjunction with this cipher. The average length of the secret key can be made close to the limit (i.e. Shannon entropy), in case the probability distribution of the encrypted messages is known or unknown. In the former case, the well-known Huffman code can be applied, in the latter, a universal code (or standard archiver).

## 2 Data compression

### 2.1 Prefix-free and trimmed codes

Let  $A$  be a finite alphabet and  $A^n$  be the sets of  $n$ -letter words ( $n \geq 1$ ) and  $A^* = \cup_{n=1}^{\infty} A^n$ . For some  $n \geq 1$ , a code is a map  $\phi : A^n \rightarrow \{0, 1\}^*$ . A code  $\phi$  is lossless if there exists  $\phi^{-1}$  such that  $\phi^{-1}(\phi(m)) = m$  for any  $m \in A^n$ . We will consider so-called prefix-free codes. It means that the set of all codewords  $\{\phi(m) : m \in A^n\}$  is prefix-free. (Recall that, by definition, some set of words  $U$  is prefix-free if for all words  $u, v \in U$  neither  $u$  is a prefix of  $v$  nor  $v$  is a prefix of  $u$ .)

Trimmed codes are designed to convert any code that has several very long code words to shorten them so that the prefix-free code is translated into a prefix-free new code. More precisely, let  $\lambda$  be a code for elements from  $M = m_1, \dots, m_L, M \subset A^*$ . For some code and  $m \in M$ , the codeword length  $|\lambda(m)|$  can be about  $L$ . The average length of the secret key is determined by the codeword lengths as well as complexity of the cipher proposed and depends on the lengths of the codewords. Thus, there are situations where it is convenient to use codes for which the codeword length of all letters does not exceed  $\lceil \log L \rceil + 1$ . (instead of about  $L$ ). We call such codes trimmed,

and define one specific code with this property as follows: if  $\lambda$  is a code then

$$\lambda^{tr}(a_i) = \begin{cases} 0 \lambda(a_i) & \text{if } |\lambda(a_i)| \leq \lceil \log L \rceil \\ 1 \text{ bin}_{\lceil \log L \rceil}(i) & \text{if } |\lambda(a_i)| > \lceil \log L \rceil, \end{cases} \quad (1)$$

where  $\text{bin}_{\lceil \log L \rceil}(i)$  is a binary presentation of  $i$  whose length is  $\lceil \log L \rceil$ . (For example,  $\text{bin}_4(3) = 0011$ ). We see that the maximal codeword length is not greater than  $\lceil \log L \rceil + 1$ . (Also, note that there are prefix-free codes for which the maximal codeword length is  $\lceil \log L \rceil$ .)

Let us explain how to decode  $\lambda^{tr}$ . First, the decoder reads the first binary letter. If it is 0, the decoder uses the codeword of the code  $\lambda$  in order to find the encoded letter. If the first letter is 1, the next  $\lceil \log L \rceil$  letters contain the binary decomposition of  $i$ , i.e. the letter is  $m_i$ .

## 2.2 Data compression and the Shannon entropy

Ever since C. Shannon published his famous paper [3], it has been known that Shannon entropy is a lower bound on the average length of any lossless prefix-free code. In the proposed cipher, the average length of the secret key used is equal to the average length of the code word used to compress the data before encryption. Therefore, let us briefly consider data compression methods separately for the cases of known and unknown statistics.

Suppose there is a set  $A^n$ ,  $n \geq 1$ , and a probability distribution  $p$  on  $A^n$ . The famous Huffman code has a minimum average codeword length lying in the interval  $[h(p), h(p) + 1)$ , where  $h(p) = -\sum_{v \in A^n} p(v) \log_2 p(v)$  is Shannon entropy [4]. For some distribution  $p$  the codeword length can be  $|A^n| - 1$ . There may be a situation where this is inconvenient and then a truncated code can be used. In this case, the average codeword length can be bounded from above by  $h(p) + 2$ .

Let us consider the data compression methods (or universal codes) for the case of unknown statistics. Note that nowadays there are many universal codes which are based on different ideas and approaches, among which we note the PPM universal code [5], the arithmetic code [6]), the Lempel-Ziv (LZ) codes [7], the Burrows-Wheeler transform [8] which is used along with the book-stack (or MTF) code [9–11], the class of grammar-based codes [12, 13] and some others [14–17]. These codes are universal. This means that, asymptotically, the length of the compressed file goes to the smallest possible value, i.e. the Shannon entropy ( $h(\nu)$ ) per letter. It is worth noting

that modern archivers are based on these codes and are also universal. (As far as a real computer program can satisfy the asymptotic properties.)

When some real data compressor  $\phi$  is applied to long messages (say  $m \in A^n$  and  $n \geq 1000$ ), it may be difficult to find the value

$$l = \max_{u \in A^n} |\phi(u)| \quad (2)$$

that is used for encryption. In this case, a truncated code  $\phi^{tr}$  can be used, since for this code the maximum codeword length is  $\lceil n \log |A| \rceil + 1$  and is known in advance, see (1).

### 3 Description of the cipher

. Let Alice want to send the word  $m = m_1 \dots m_n \in A^n$  to Bob and apply a cipher to hide  $m$  from Eve. Alice and Bob are going to use the prefix-free data compressor  $\phi$  and some bits of secret key  $k$  whose length is not less than  $l$  in (2).

Alice finds  $\phi(m)$  (let it be  $x_1 \dots x_s$ ) and calculates  $y_1 = x_1 \oplus k_1, y_2 = x_2 \oplus k_2, \dots, y_s = x_s \oplus k_s$ . Alice then generates  $l - s$  of random independent binary digits  $r_1 r_2 \dots r_{l-s}$  and constructs an encrypted word

$$E(m, k, r) = y_1 \dots y_s r_1 r_2 \dots r_{l-s}. \quad (3)$$

Alice then sends Bob the word  $E(m, k, r)$ . While decoding it, Bob calculates  $x_1 = y_1 \oplus k_1, x_2 = y_2 \oplus k_2, \dots$  and stops this process as soon as it gets a word of  $v, v \in \{\phi(m), m \in A^n\}$ . (Note that such a word  $v$  is a single word, since  $\phi$  is prefix-free.) We denote this cipher by  $C\&R$  because it is based on compression and randomisation.

The following statement describes the properties of the cipher  $C\&R$ .

**Theorem 1** . *Let the cipher  $C\&R$  be applied to messages from some  $M$  and  $E$  be a set of all possible encrypted words (3). Then,  $C\&R$  is a perfect cipher, that is*

$$P(m|e) = P(m)$$

*for all  $m \in M, e \in E$  (where  $P(m)$  and  $P(m|e)$  are probability and conditional probability, correspondingly and the equality is as in the definition of a perfect cipher [3]).*

*Proof of Theorem.* Let  $\phi$  be a data compressor used in the considered cipher  $C\&R$  and let  $m$  and  $e$  be any elements from  $M$  and  $E$ ,  $\phi(m) = x_1x_2\dots x_s$ , where  $s = |\phi(m)|$ .

C. Shannon showed that the cipher is perfect if and only if

$$P(e|m) = P(e)$$

for all  $m \in M, e \in E$  (where  $P(m)$  and  $P(m|e)$ , see [3]). We will use this theorem and first estimate  $P(e|m)$  as follows:

$$P(e|m) = \prod_{i=1}^s P\{k_i = x_i \oplus e_i\} \left( \prod_{j=s+1}^l P\{r_j = e_j\} \right),$$

where  $l$  is defined in (2). Indeed, this equation shows that  $x_i \oplus k_i = x_i \oplus (x_i \oplus e_i) = e_i$ , and the random variables  $x_i$  and  $r_j$  obey the Bernoulli distribution, hence  $P(e|m) = 2^{-l}$ . Now we estimate  $P(e)$ .

$$P(e) = \sum_{m \in M} P(m) \left( \prod_{i=1}^{|\phi(m)|} P\{k_i = x_i \oplus e_i\} \left( \prod_{j=|\phi(m)|+1}^l P\{r_j = e_j\} \right) \right),$$

where  $\phi(m) = x_1\dots x_{|\phi(m)|}$ . Repeating the estimates from the previous case, we see that  $P(e) = \sum_{m \in M} P(m) 2^{-l}$ . Hence,  $P(e) = 1 2^{-l} = 2^{-l}$ . So,  $P(e) = P(e|m)$  and the theorem is proven.

## References

- [1] Shannon C. E. Communication theory of secrecy systems. The Bell system technical journal. 1949 Oct; 28(4):656-715.
- [2] Shkel YY, Poor HV. A compression perspective on secrecy measures. IEEE Journal on Selected Areas in Information Theory. 2021 Feb 2;2(1):163-76.
- [3] Shannon CE. A mathematical theory of communication. The Bell system technical journal. 1948 Jul;27(3):379-423.
- [4] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 2006.
- [5] J. Cleary and I. Witten, “Data compression using adaptive coding and partial string matching,” *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.
- [6] J. Rissanen and G. G. Langdon, “Arithmetic coding,” *IBM Journal of research and development*, vol. 23, no. 2, pp. 149–162, 1979.
- [7] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on information theory*, vol. 23, no. 3, pp. 337–343, 1977.
- [8] M. Burrows and D. J. Wheeler, “A block-sorting lossless data compression algorithm,” 1994.
- [9] B. Y. Ryabko, “Data compression by means of a “book stack”,” *Problems of Information Transmission*, vol. 16, no. 4, pp. 265–269, 1980.
- [10] J. Bentley, D. Sleator, R. Tarjan, and V. Wei, “A locally adaptive data compression scheme,” *Communications of the ACM*, vol. 29, no. 4, pp. 320–330, 1986.
- [11] B. Ryabko, N. R. Horspool, G. V. Cormack, S. Sekar, and S. B. Ahuja, “Technical correspondence,” *Communications of the ACM*, vol. 30, no. 9, pp. 792–797, 1987.

- [12] J. C. Kieffer and E.-H. Yang, “Grammar-based codes: a new class of universal lossless source codes,” *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 737–754, 2000.
- [13] E.-H. Yang and J. C. Kieffer, “Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform. I. without context models,” *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 755–777, 2000.
- [14] M. Drmota, Yu. Reznik, and W. Szpankowski, “Tunstall code, Khodak variations, and random walks,” *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2928–2937, 2010.
- [15] B. Ryabko, “Twice-universal coding,” *Problems of Information Transmission*, vol. 3, pp. 173–177, 1984.
- [16] G. Louchard, W. Szpankowski, “Average profile and limiting distribution for a phrase size in the Lempel-Ziv parsing algorithm”, *IEEE Transactions on Information Theory*. vol. 41, no. 2, pp. 478-488, 1995.
- [17] Reznik YA, Li X, Lillevold KO, Jagannath A, Greer J. Optimal multi-codec adaptive bitrate streaming. In 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) 2019 Jul 8 (pp. 348-353). IEEE.