Преглед НЦД 13 (2008), 22–26

Miodrag Živković
(Matematički fakultet, Beograd)

SELECTIVE ENCYPTION OF HUFFMAN COMPRESSED TEXT

Abstract. The security of multimedia data in digital distribution networks is commonly provided by encryption, i.e., the mathematical process that transforms a plaintext message into unintelligible ciphertext. *Selective encryption* is a recent approach to reduce the computational requirements for huge volumes of multimedia data in distribution networks with different client device capabilities. We propose a method for selective encryption that can be applied to compressed data – for example to Huffman compressed text.

Keywords: selective encryption, ciphertext-only attack, randomness test.

1. Introduction

It is easy to make compressed text unreadable, because it is susceptible to errors. For example, changing a single bit in jpeg image file can cause significant changes in a picture. This fact suggests that it is easier to encrypt compressed text. Shannon [3] shows that the *unicity distance* (minimum length of ciphertext needed to uniquely determine the cipher key) is given by $H(K)/D$, where $H(K)$ is the number of key bits, and $1 - D$ is information content of one ciphertext bit. In other words, theoretical security of enciphered text depends on its redundancy. If ciphertext is totally irredundant, i.e. it looks as a sequence of independent random bits, and then it is absolutely secure.

Stream ciphers are based on using *pseudo random number generators* (PRNG). Input to a PRNG is a parameter called *key,* and output is a binary sequence $k_0,k_1,\ldots,k_{n-1,\ldots}$ which behaves as a realization of a sequence of independent uniformly distributed (0,1) random variables. PRNG is usually constructed using recurrent sequences, for example linear recurrent sequences (see [1] for example). Let $p_0,p_1,\ldots,p_{n-1}$ denote binary plaintext, and let $k_0,k_1,\ldots,k_{n-1}$ denote the binary key sequence – the output of PRNG. Standard way to obtain ciphertext $c_0,c_{1,}\ldots,c_{n-1}$ is to apply XOR operation (addition modulo 2) to corresponding bits of plaintext and the key sequence: $c_i = p_i \circ k_i$, $i = 0,1,\ldots,n-1$. Therefore, to encipher n bits of plaintext we generally need n bits of key sequence.

Our goal is to improve enciphering efficiency by using less than n bits of key sequence to encipher plaintext of length n. One obvious solution is to chose k, $1 < k < n$, and set $c_i = p_i \circ k_i$ if i is a multiple of k, and $c_i = p_i$ otherwise. If plaintext is a result of compression, then ciphertext is partially distorted plaintext. *Ciphertext-only attack* (determining plaintext knowing only ciphertext) is equivalent to decompression starting from ciphertext – which is hard or impossible. Reconstruction of the plaintext is still possible if we have long enough ciphertext. The reconstruction is based on (still remaining) redundancy of compressed text.

2. Proposed encryption method

The problem stated above can be solved more elegantly. Having chosen k, $1 < k < n$, as above, divide plaintext in blocks of length k. Let $[x]$ denote integral part of the real number x. The ith plaintext block is $P_i = (p_{ik},p_{ik+1},\ldots,p_{ik+k-1})$, $i = 0,1,\ldots,[n/k]$. The last block is filled with zeros: we suppose $p_{n+j} = 0$ if $j \geq 0$. The first k bits of the key sequence $M = (c_0,c_1,\ldots,c_{n-1})$ are used as a mask. An alternative mask is $M \circ I$, where I is a k-tuple $(1,1,\ldots,1)$ and $\circ$ denotes bitwise XOR operation. The ith ciphertext block $C_i = (c_{ik},c_{ik+1},\ldots,c_{ik+k-1})$ is obtained from P_i using the mask M or $M \circ I$, depending on the key sequence bit p_{k+i}:

$$C_i = P_i \circ p_{k+i}I \circ M, \qquad I = 0,1,\ldots,[n/k] \tag{1}$$

(the multiplication in $p_{k+i}I$ is componentwise). The number of key sequence bits used is $k + [n/k]$. For small k that is approximately k times less than if the standard encryption is used. In exchange, we have used the same key sequence block (M or $I \circ M$) many times; still, it is not seen from ciphertext which block is used at which positions. The optimal saving factor is $O(n^{1/2})$, obtained for $k = O(n^{1/2})$. If $k = 1$ or $k = n$, then encryption is in fact standard encryption: there is no saving in key sequence bits, but there is no security compromise potentially induced.

3. Analysis – randomness tests

To evaluate proposed encryption scheme, we carried out a number of experiments, testing randomness of ciphertexts.

Plaintexts are obtained applying Huffman encoding to a collection of ASCII texts in English language from Gutenberg project [2].

Key sequence is obtained using DES [3] algorithm. Let DES(X,K) denote the 64-bit block obtained by enciphering input 64-bit block X using the 56-bit key K, and let 3DES(X,K) denote the result of triple encipherment of X using the triple key $K = (K_1,K_2,K_3)$:

$$3\mathrm{DES}(X,K) = \mathrm{DES}(\mathrm{DES}(\mathrm{DES}(X,K_1),K_2),K_3).$$

The ith 64-bit key sequence block X_i is obtained iteratively,

$$X_i = 3\mathrm{DES}(X_{i\text{-}1},K), \qquad i \geq 0,$$

where

$$X_{i-1} = (0123456789abcdef)$$

and

$$K = (0123456789abcdef, 123456789abcdef\,0, 23456789abcdef\,01).$$

Here hexadecimal digits denote the corresponding 4-bit groups.

Ciphertext is obtained as explained above.

The encipherment quality is measured testing ciphertext randomness. It is expected not only that the entropy of ciphertext bit is 1 (i.e., that the number of 1s in ciphertext is close to $n/2$), but even more, that ciphertext successfully passes every conceivably randomness test. Here we chose a particular randomness test: χ^2 test of uniform distribution of ciphertext pentagrams. Let $n' = n - 4$ and let

$$c'_i = 16c_i + 8c_{i+1} + 4c_{i+2} + 2c_{i+3} + c_{i+4}, 0 \leq i \leq n',$$

denote the 5-bit numbers corresponding to ciphertext. We count occurrences

$$f_i = |\{j \mid 0 \le j < n', c_j' = i\}|$$

of pentagrams i, $0 \le i < 32$, and then we compute χ^2 statistics

$$Y = \sum_{i=0,31} (f_i - n'/32)^2 / (n'/32).$$

If c_i is a sequence of independent, uniformly distributed random (0,1) variables, then random variable Y has χ^2 probability distribution with 31 degrees of freedom. For each Y we compute the probability p of the event that χ^2 (with 31 degrees of freedom) distributed random variable exceeds Y. We consider that ciphertext passes the test (or we say it is random) if $p < 0.999$. The test is repeated for $n = 2^l$, $l = 8,9,\ldots,23$, and for $k = 2^b$, $b = 0,1,2,\ldots,14$. The values of p obtained for each combination of l and b are listed in Table 1, where the cells corresponding to $p \ge 0.999$ are shaded.

E	H	Text enciphered using table of size 2^b bits; b:														
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
8	0.095	0.000	0.26	0.55	0.85	0.00	0.06	0.14	0.13	0.03	0.05	0.05	0.05	0.05	0.05	0.05
9	0.691	0.151	0.12	0.79	0.09	0.02	0.02	0.33	0.12	0.00	0.08	0.08	0.08	0.08	0.08	0.08
10	0.990	0.312	0.74	0.84	0.73	0.10	0.44	0.84	0.52	0.23	0.85	0.03	0.03	0.03	0.03	0.03
11	0.999	0.523	0.60	0.99	0.50	0.46	0.31	0.46	0.04	0.55	0.96	0.00	0.03	0.05	0.05	0.05
12	1.000	0.709	0.43	0.99	0.08	0.98	0.77	0.27	0.40	0.77	0.89	0.04	0.12	0.64	0.57	0.57
13	1.000	0.020	0.86	1.00	0.16	1.00	0.89	0.69	0.86	0.40	0.95	0.25	0.24	0.37	1.00	1.00
14	1.000	0.323	0.99	1.00	0.48	0.86	1.00	0.77	0.24	0.23	0.63	0.50	0.03	0.95	0.89	1.00
15	1.000	0.013	0.99	1.00	0.75	0.98	1.00	0.12	0.00	0.78	0.50	0.29	0.03	0.67	1.00	0.75
16	1.000	0.845	1.00	1.00	1.00	0.99	0.98	0.42	0.08	0.86	0.98	0.02	0.07	0.96	1.00	0.80
17	1.000	0.985	1.00	1.00	1.00	1.00	1.00	0.89	0.57	0.34	0.97	0.11	0.74	0.01	0.36	0.16
18	1.000	0.449	1.00	1.00	1.00	1.00	1.00	1.00	0.65	0.01	0.80	0.99	0.94	0.02	0.01	0.53
19	1.000	0.287	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.41	1.00	0.77	0.34	0.00	0.46	0.07
20	1.000	0.120	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.89	0.99	0.70	0.86	0.74
21	1.000	0.602	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.66	0.22	0.04
22	1.000	0.997	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.84	0.43
23	1.000	0.260	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.85	0.01

Table 1. χ^2 probabilities corresponding to pentagram frequencies distribution
(column E: encoded text of size 2^l bytes, values for l; column H: Huffman encoded text)

Looking at Table 1, we see that

1) the plaintext (the column Huffman encoded text in Table 1) is not random if $n > 2^{11}$;
2) ciphertext is random for all n if $k = 1$;
3) if $k > 1$, then ciphertext is random for k large, roughly if $k > n/8000$.

It is interesting that when k increases, the number of key sequence bits used decreases, but the randomness of ciphertext also increases! These seemingly paradoxical results can be explained as follows. The randomness test we used is 'local', it is checking dependence of ciphertext bits which are at distance at most 5. Since that dependence is destroyed as better as k is larger, it follows that the number of key bit sequence decreases *and* the encryption quality (measured as explained above) increases.

Since the results of randomness tests were unexpected, we carried out alternative tests. Before counting pentagram occurrences, we perform 'decorrelation' of ciphertext bits at distance k: instead of the sequence c_i we consider the sequence $c_i \circ c_{k+I}$, $i = 0,1,\ldots$, to which we apply the same randomness test. The new results are shown in Table 2.

E	H	Text enciphered using table of size 2^b bits; b:														
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
8	0.095	0.000	0.26	0.42	0.43	0.07	0.34	0.44	0.01	0.58	0.37	0.04	0.70	0.34	0.23	0.94
9	0.691	0.151	0.12	0.95	0.74	0.02	0.01	0.10	0.00	0.88	0.70	0.06	0.24	0.04	0.20	0.79
10	0.990	0.312	0.74	0.96	0.21	0.02	0.10	0.57	0.09	0.51	0.88	0.00	0.01	0.09	0.69	0.03
11	0.999	0.523	0.60	0.32	0.44	0.21	0.06	0.10	0.43	0.99	1.00	0.07	0.01	0.86	0.62	0.20
12	1.000	0.709	0.43	0.34	0.16	0.19	0.00	0.43	0.70	0.74	1.00	0.07	0.37	0.08	0.28	0.09
13	1.000	0.020	0.86	0.02	0.89	0.65	0.00	0.94	0.00	0.70	0.87	0.10	0.17	0.95	0.37	0.16
14	1.000	0.323	0.99	0.94	0.94	0.52	0.00	0.76	0.46	0.86	0.35	0.61	0.09	0.71	0.25	0.09
15	1.000	0.013	0.99	1.00	0.97	1.00	0.25	0.54	0.31	1.00	0.00	0.49	0.15	0.73	0.38	0.42
16	1.000	0.845	1.00	1.00	1.00	0.92	0.79	0.71	0.94	0.99	0.01	0.88	0.16	0.00	0.58	0.69
17	1.000	0.985	1.00	1.00	1.00	1.00	0.83	1.00	1.00	1.00	0.19	0.94	0.81	0.60	0.49	0.20
18	1.000	0.449	1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.99	0.69	0.98	0.04	0.22	0.25	0.17
19	1.000	0.287	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.86	1.00	0.01	0.04	0.10	0.02
20	1.000	0.120	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.01	0.88	0.01	0.48
21	1.000	0.602	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.62	0.32	0.00	0.92
22	1.000	0.997	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.39	0.05	0.96
23	1.000	0.260	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.68	0.37	0.92

Table 2. χ^2 probabilities corresponding to pentagram frequencies distribution in decorrelated ciphertext (column E: encoded text of size 2^l bytes, values for l; column H: Huffman encoded text)

Now

1) for $k < 1000$ ciphertext is random if and only if (roughly) $n < 100000$, for all $k > 1$;
2) for $k > 1000$ ciphertext is generally random if $n < 8000000$ (which is the upper bound of length in our experiments). This seemingly strange behavior is explained by the lack of correlation in plaintext at distances greater than 1000.

Summarizing, randomness tests we have chosen do not impose any upper bound on k. In order to minimize the use of computationally demanding PRNG it would be optimal to chose k close to $n^{1/2}$, therefore enabling to encrypt n bits of plaintext using only $O(n^{1/2})$ bits from PRNG output. Next we show that choosing large k implies the possibility to mount a ciphertext-only attack.

4. Ciphertext-only attack

It is reasonable to expect that the systematic use of two masks, even in an obfuscated manner, must cause some problems. We now demonstrate that ciphertext-only attack is possible if k is large enough. From (1) it follows that the block $C_i \circ C_{i+1}$ is simply connected to plaintext:

$$C_i \circ C_{i+1} = P_i \circ P_{i+1} \circ (p_{k+i} \circ p_{k+i+1})I,$$

which equals to $P_i \circ P_{i+1}$ if $p_{k+i} = p_{k+i+1}$, and $P_i \circ P_{i+1} \circ I$ otherwise. If k is large enough, then $P_i \circ P_{i+1}$ and $P_i \circ P_{i+1} \circ I$ have different probability distributions, characterizing the particular plaintext. Using this fact, an attacker can decide if $p_{k+i} = p_{k+i+1}$ for each $i \geq 0$. For fixed $p_k = 0$ or $p_k = 1$ all bits p_i, $i \geq k$, are uniquely determined. Afterwards it remains to break enciphering algorithm similar to Viegener cipher: now the attacker knows the sequence of blocks $C_i' = C_i \circ p_{k+i}I = P_i \circ M$. This makes it possible to deduce statistically the mask M. Note that such an attack is much easier to perform if the plaintext *is not compressed*.

Consequently, in order to avoid the described attack, one must not choose large k. The upper bound on k depends on plaintext statistics, i.e., on the compression quality.

Bibliography

[1] R. A. Mollin, *An Introduction to Cryptography*, Chapman & Hall/CRC, New York, 2007.
[2] Project Gutenberg, http://www.gutenberg.org
[3] C. E. Shannon, *Communication theory of secrecy systems*, Bell System Technical J. **28** (1949), 656–715.

ezivkovm@matf.bg.ac.yu