Entropy of Selectively Encrypted Strings

Reine Lundin and Stefan Lindskog

Department of Computer Science
Karlstad University
Sweden
{reine.lundin|stefan.lindskog}@kau.se

Abstract. A feature that has become desirable for low-power mobile devices with limited computing and energy resources is the ability to select a security configuration in order to create a trade-off between security and other important parameters such as performance and energy consumption. Selective encryption can be used to create this trade-off by only encrypting chosen units of the information. In this paper, we continue the investigation of the confidentiality implications of selective encryption by applying entropy on a generic selective encryption scheme. By using the concept of run-length vector from run-length encoding theory, an expression is derived for entropy of selectively encrypted strings when the number of encrypted substrings, containing one symbol, and the order of the language change.

Keywords: computer security, security measures, selective encryption, entropy.

1 Introduction

The ability to select a security configuration is a feature that has become desirable for low-power mobile devices acting in heterogeneous wireless network environments with limited computing and energy resources. A selective security service is a service that provides various security configuration at run-time to create a trade-off between security and other important parameters such as performance and energy consumption. Selective security is also a way to comply with the principle of adequate security, which states that resources should only be protected to a degree consistent with their value and only until they lose their value.

The concept of selective encryption was introduced in 1995 and 1996 for the purpose of reducing the amount of encrypted MPEG data in a video stream while still providing an acceptable level of confidentiality [9]. Selective encryption has also been used to save energy and processing time for H.264/AVC video streams [7], JPEG images [6], speech compressed with the G.729 speech encoding standard [10], and a wireless video camera [3]. Previous work on selective encryption has mainly focused on performance and/or energy saving issues and on making selectively encrypted information perceptively secure to a certain protection level: that is, to determine which parts of the information to encrypt to distort its perception beyond a desired threshold. In this paper, we continue the investigation in [5] of the confidentiality implications of selective encryption by applying entropy on the generic selective encryption scheme presented in [4]. Using the concept of run-length vector from run-length encoding theory, an expression is derived for entropy of selectively encrypted strings when the number of encrypted substrings, containing one symbol, and the order of the language change.

The remainder of the paper is organized as follows. Sect. 2 introduces terminology and definitions of languages and entropy. Selective encryption is discussed in Sect. 3, and this section also presents the concept of run-length vector from run-length encoding. The expression for entropy of selectively encrypted strings is derived in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Terminology and Definitions

Terminology and definitions of languages and entropy are introduced in this section.

2.1 Languages

In language theory an alphabet Σ is a finite non-empty set of symbols and a string s over Σ is a finite sequence of symbols drawn from that alphabet. The length of a string, $|s|$, is the number of symbols in the string. If no symbol is drawn from the alphabet, the empty string ϵ is created, having $|\epsilon| = 0$. The concatenation operator $|$ is used to join two strings together by appending. Hence, the string $s_1|s_2$ is produced by appending s_2 to s_1. This is often written as $s_1 s_2$ without the concatenation operator. Concatenation of a string with the empty string yields the string itself, $s\epsilon = \epsilon s = s$; thus ϵ is the identity string during concatenation.

The set of all strings over an alphabet Σ is called the transitive closure Σ^* and every set $L \subset \Sigma^*$ is called a language. The size of a language, $|L|$, is the number of strings in the language. An n-language, L^n, is a subset of a language L containing the strings of length n, hence

$$L^n = \{s \in L; |s| = n\} \tag{1}$$

Note that the union of all n languages constitutes the whole language, hence $L = \bigcup_{n=0}^{k} L^n$, where k is an arbitrarily large integer. Furthermore, $L^0 = \{\epsilon\}$ and $L^1 = \Sigma$. Thus, L^1 can both refer to a language with strings of length one and the constructing alphabet.

The symbols in a language will normally have different probabilities that depend on preceding symbols. Orders of languages, ω, to approximate the originally language were proposed in [8]. The idea is shown in the following list.

L_0 Zero-order language, symbols are independent and uniformly distributed.

L_1 First-order language, symbols are independent and distributed as in L.

L_2 Second-order language, symbols are dependent on one preceding symbol with probabilities as in L.

L_n n-order language, symbols are dependent on $n-1$ preceding symbols with probabilities as in L.

2.2 Entropy

Entropy $H(X)$ [8] is a measure that gives the average amount of information of a discrete random variable X. However, entropy can also be seen as a measure giving the average number of guesses in an optimal binary search attack. The discrete random variable X is a variable that attains values from finite sample space $\mathcal{X} = \{x_1, \ldots, x_n\}$ with probability distribution $p_i = p(X = s^i) = p(X^i)$. From this, entropy is defined as follows.

Definition 1. *The entropy $H(X)$ of a random variable X with probability distribution p_i is defined as*

$$H(X) = -\sum_i p_i \log_2 p_i \tag{2}$$

Definition 1 can be extended to joint and conditional entropy [1].

Definition 2. *The joint entropy $H(X_0, X_1)$ of a pair of random variables (X_0, X_1) with joint probability distribution p_{ij} is defined as*

$$H(X_0, X_1) = -\sum_{i,j} p_{ij} \log_2 p_{ij} \tag{3}$$

Definition 3. *The conditional entropy $H(X_1|X_0)$ of the random variable X_1 given the random variable X_0 with conditional probability distribution $p_{j|i}$ is defined as*

$$H(X_1|X_0) = \sum_i p_i H(X_1|X_0^i) = -\sum_{i,j} p_{ij} \log_2 p_{j|i} \tag{4}$$

Definition 2 can be generalized to n random variables that are related in the chain rule as follows.

$$H(X_0, \ldots, X_{n-1}) = H(X_0) + \sum_{i=1}^{n-1} H(X_i|X_0, \ldots, X_{i-1}) = \sum_{i=0}^{n-1} H^i \tag{5}$$

3 Selective Encryption

As stated above, the main idea of selective encryption is to create a trade-off between confidentiality and performance by encrypting chosen substrings of a string while leaving the remaining substrings unencrypted, compressed or encrypted with another encryption algorithm. In this paper, the substrings are assumed to be of equal size, containing one symbol, and the remaining substrings are assumed to be unencrypted and given in position.

The generic selective encryption scheme presented in [4] consists of three basic entities: the string s to be selectively encrypted, the bit vector b that controls which substrings of s to encrypt and the selectively encrypted message $E(s)$. In the scheme, s is divided into n equally sized substrings s_i, $0 \leq i < n$, hence

$$s = \mathop{\Big|}\limits_{i=0}^{n-1} s_i \tag{6}$$

Furthermore, s_i is encrypted if $b_{i \bmod |b|} = 1$ and left unencrypted if $b_{i \bmod |b|} = 0$. Without a loss of generality it can be assumed that $n = |b|$, hence the modulus operator can be removed. The selectively encrypted string $E(s)$ is now constructed as follows.

$$E(s) = \mathop{\Big|}\limits_{i=0}^{n-1} \begin{cases} s_i & \text{if } b_i = 0 \\ E(s_i) & \text{if } b_i = 1 \end{cases} \tag{7}$$

From the number of encrypted substrings in $E(s)$, controlled by b, the encryption level is defined as

$$EL = \frac{\sum_{i=0}^{n-1} b_i}{n} \tag{8}$$

The concept of run-length vector from run-length encoding theory [1] is used in this paper in order to capture the distribution of zeros and ones in the bit vector. In run-length encoding, information is stored as a run-length value and a single instance of the corresponding data entity, where a run is the longest substring from the current position containing identical data entities. Thus the description length of information containing long runs will decrease. However, if the information does not contain long runs, the description length of the information might instead increase. The sequence of run-length values of the information is called the run-length vector r. For instance, the bit vector $(0, 0, 0, 1, 1, 0, 0)$ can be written as (302120), with the corresponding run-length vector $r = (3, 2, 2)$. Note how r captures the distribution of runs of zeros and ones in the bit vector. By using the convention of letting the first element in the run-length vector express the run-length of zeros at the beginning of the bit vector, even if there are none, r_{2j} will then give the run-length of zeros and r_{2j+1} will give the run-length of ones in the bit vector. The elements in r will thus alternate between giving the run-lengths of zeros and ones of the bit vector, starting with zeros.

From the notation of the 1-norm [2], also called the taxicab geometry or Manhattan distance, the partial cumulative sum of a vector v will be denoted

$$||v_{[k,l]}|| = \sum_{i=k}^{l} |v_i| \tag{9}$$

where k is the starting position and l the ending position of the vector. Note that $||v_{[k,l]}|| = 0$ if $k > l$, and if $|v| = n$ then $||v_{[0,n-1]}|| = ||v||_1$. From this notation the run-length vector can be calculated from the bit vector as

$$r_i = \max\{k+1\,;\, ||\neg^i b_{[||r_{[0,i-1]}||,||r_{[0,i-1]}||+k]}|| = 0\} \tag{10}$$

where $\neg^i$ is the negation operator to the power of i. In a similar way, the bit vector can be calculated from the run-length vector as

$$b_i = \min\{k\,;\, ||r_{[0,k]}|| > i\} \mod 2 \tag{11}$$

The elements in the bit vector will be one when k is an odd integer. By setting $\alpha_j = ||r_{[0,2(j-1)]}||$ and $\beta_j = ||r_{[0,2j-1]}||$, this will happen for the index sets $I_j = [\alpha_j, \beta_j)$ where $J = [1, \lfloor \frac{|r|}{2} \rfloor]$. Since $I_{j_1} \cap I_{j_2} = \emptyset$ if $j_1 \neq j_2$, the union of all index sets

$$I = \bigcup_{j \in J} I_j \tag{12}$$

indexes all ones in the bit vector while still preserving the uniqueness of the indexing.

4 Confidentiality of Selective Encryption

To investigate how the entropy changes for selectively encrypted strings, let each of the n equally sized substrings of a selectively encrypted string be associated with a random variable as

$$X_0, \ldots, X_{n-1} = E(s) = \mathop{\Big|}_{i=0}^{n-1} \begin{cases} X_i = s_i & \text{if } b_i = 0 \\ X_i = E(s_i) & \text{if } b_i = 1 \end{cases} \tag{13}$$

Since the entropy is affected only when $b_i = 1$, unencrypted substrings only affect the entropy indirectly; it is sufficient to use the index set I in (12) to describe how entropy changes. Hence, by using (5), (12) and (13), the entropy of selectively encrypted strings can be written as

$$H_\omega(X_0, \ldots, X_{n-1}) = \sum_{i \in I} H_\omega^i = \sum_{j \in J} \sum_{i \in I_j} H_\omega^i = \sum_{j \in J} H_\omega^{I_j} = H_\omega^I \tag{14}$$

4.1 Zero- and First-order Languages

The random variables are independent for L_1 languages. Hence, the conditional entropies in (14) becomes

$$H_1^i = H(X_i) \tag{15}$$

By using (15) in (14)

$$H_1^I = \sum_{i \in I} H(X_i) = |I| H(X_0) \tag{16}$$

where the last step comes from the fact that the random variables are identically distributed. In [5] it was shown for first-order languages that the entropy is given by the expression $H(X_0) \sum_{i=0}^{n-1} b_i$. However, since $|I| = \sum_{i=0}^{n-1} b_i$, the expressions are equal. The symbols are also uniformly distributed for L_0 languages, hence (16) transforms to

$$H_0^I = |I| \log_2 |L_0^1| \tag{17}$$

From the derived expression, no confidentiality can be achieved for L_1 or L_0-languages if the number of encrypted units is zero, $|I| = 0$, or if the alphabet contains only one symbol, $|L^1| = 1$. Furthermore, intuitively and obviously, encrypting more substrings or having a larger alphabet will increase the level of confidentiality. Note also that the entropy tends to infinity as the number of encrypted substrings or the number of symbols in the alphabet tends to infinity.

4.2 Second-order Languages

For L_2 languages the probability distribution of the symbols depends on one preceding symbol. Thus, when deriving an expression for H_2^I, the symbol preceding a run of ones must be taken into consideration. In [5], two cases were shown to affect the expression of H_2^I. The first case deals with a run of ones starting at the beginning of the bit vector, $i \in I_1 = [0, \beta_1)$, and the second case deals with runs of ones not starting at the beginning of the bit vector, $i \in I_j = [\alpha_j, \beta_j)$ with $\alpha_j \neq 0$ and $X_{\alpha_j - 1}^{h_j}$. However, a single expression for H_2^I combining the two cases was not derived in the paper.

The conditional entropies in Definition 3 is defined as the average of the entropies of the conditional distributions, averaged over the conditioning distribution. Hence, in the first case, the conditional entropies in (14) can for L_2 languages be written as

$$H_2^i = p(X_0, \ldots, X_{i-1}^l) H(X_i | X_{i-1}^l) = p(L_2^{i-1} X_{i-1}^l) H(X_i | X_{i-1}^l) \tag{18}$$

where $p(L_2^{i-1} X_{i-1}^l)$ is a row vector of the second-order probabilities of the strings ending with the substring X_{i-1}^l and $H(X_i | X_{i-1}^l)$ is a column vector of the conditional entropies. By using (18) in (14), the expression of the first case becomes

$$H_2^{I_1} = H(X_0) + \sum_{i=1}^{\beta_1 - 1} p(L_2^{i-1} X_{i-1}^l) H(X_i | X_{i-1}^l) \tag{19}$$

In the second case, the conditional entropies in (14) can for L_2 languages be written as

$$\begin{aligned} H_2^i &= p(X_{\alpha_j-1},\ldots,X_{i-1}^l|X_{\alpha_j-1}^{h_j})H(X_i|X_{i-1}^l) \\ &= p(L_2^{i-\alpha_j}X_{i-1}^l|X_{\alpha_j-1}^{h_j})H(X_i|X_{i-1}^l) \end{aligned} \tag{20}$$

By using (20) in (14), the expression of the second case becomes

$$H_2^{I_j} = \sum_{i\in I_j} p(L_2^{i-\alpha_j}X_{i-1}^l|X_{\alpha_j-1}^{h_j})H(X_i|X_{i-1}^l) \tag{21}$$

If $\alpha_1 \neq 0$, then it is only necessary to sum over all I_j in (21) to derive H_2^I. However, if $\alpha_1 = 0$, then (19) needs to be included in the sum. To combine the two cases, the alphabet L^1 must be extended with a new special symbol δ to $\mathbb{L}^1 = L^1 \bigcup\{\delta\}$. Language $\mathbb{L}^2$ is constructed as an extension to language L_2 by setting $p(\delta) = 1$, $p(\delta|\gamma_i) = 0$ and $\forall\gamma_i \in L^1$ $p(\gamma_i|\delta) = p(\gamma_i)$. Thus, all strings in $\mathbb{L}_2$ start with the to L_2 independent substring δ and then continue as in L_2. Now, by setting $X_{-1}^{h_1} = \delta$, it is possible to rewrite (19) as follows.

$$\begin{aligned} H_2^{I_1} &= p(X_{-1}^l|X_{-1}^{h_1})H(X_0|X_{-1}^l) + \sum_{i=1}^{\beta_1-1} p(L_2^{i-1}X_{i-1}^l|X_{-1}^{h_1})H(X_i|X_{i-1}^l) \\ &= \sum_{i\in I_1} p(L_2^{i-\alpha_1}X_{i-1}^l|X_{\alpha_1-1}^{h_1})H(X_i|X_{i-1}^l) \end{aligned} \tag{22}$$

Note that (22) is a special case of (21) with $j = 1$, hence

$$H_2^I = \sum_{j\in J}\sum_{i\in I_j} p(L_2^{i-\alpha_j}X_{i-1}^l|X_{\alpha_j-1}^{h_j})H(X_i|X_{i-1}^l) \tag{23}$$

For L_1 languages (23) transforms as

$$H_2^I = \sum_{j\in J}\sum_{i\in I_j} p(L_1^{i-\alpha_j}X_{i-1})H(X_i) = \sum_{i\in I} H(X_i) = H_1^I \tag{24}$$

Furthermore, H_2^I can be larger or smaller than H_1^I. For instance, the probability distribution in Table 1 gives $H_1^I < H_2^I$ if X_0^1 and $H_2^I < H_1^I$ if X_0^2. In Fig. 1 the eight different states of a selectively encrypted string containing three substrings are illustrated with encrypted substrings colored gray and unencrypted

Table 1. The probability distribution that gives $H_1^I < H_2^I$ if X_0^1 and $H_2^I < H_1^I$ if X_0^2.

p_{ij}	X_1^1	X_1^2
X_0^1	0.2	0.2
X_0^2	0.2	0.4

substrings colored white. The states are grouped into columns according to the encryption level, with arrows pointing towards the next state containing the encrypted substrings of the current state. By using (23) the entropy of the different states becomes

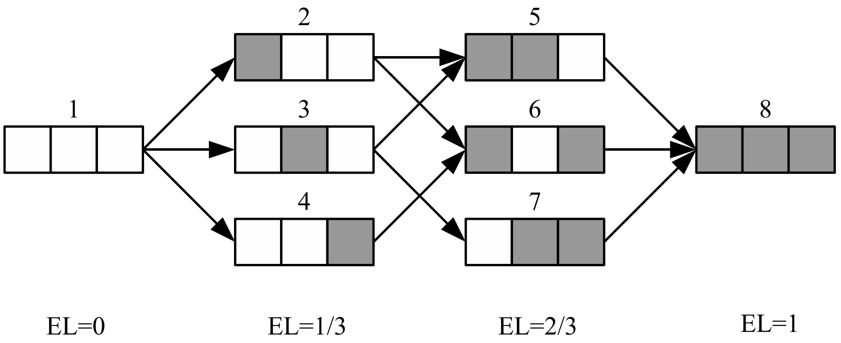

Fig. 1. The eight states of a selectively encrypted string containing three substrings.

1. $H_2^I = 0$
2. $H_2^I = H(X_0)$
3. $H_2^I = H(X_1|X_0^{h_1})$
4. $H_2^I = H(X_2|X_1^{h_1})$
5. $H_2^I = H(X_0, X_1)$
6. $H_2^I = H(X_0) + p(X_1^l|X_1^{h_2})H(X_2|X_1^l)$
7. $H_2^I = p(X_0^l|X_0^{h_1})H(X_1|X_0^l) + p(L_2^1 X_1^l|X_0^{h_1})H(X_2|X_1^l)$
8. $H_2^I = H(X_0, X_1, X_2)$

4.3 Third-order Languages

For L_3 languages, the probability distribution of the symbols depends on the two preceding symbols. Thus, when deriving an expression for H_3^I, two symbols preceding a run of ones must be taken into consideration. Note that it is only the first preceding symbol of a run of ones that is known with certainty to be unencrypted. The second preceding symbol could either be encrypted or unencrypted; this will be denoted $X_i^\dagger$.

From the alphabet $\mathbb{L}^1 = L^1 \bigcup \{\delta\}$, language $\mathbb{L}_3$ is constructed as an extension of language L_3 by setting $p(\delta^2) = 1$, $p(\delta|\gamma_i) = 0$ and $\forall \gamma_i \in L^1$, $p(\gamma_i|\delta^2) = p(\gamma_i)$. Thus, all strings in $\mathbb{L}_3$ start with the independent substring δ^2 and then continue as in L_3. The conditional entropies in (14) can now be written for L_3 languages as

$$\begin{aligned} H_3^i &= p(X_{\alpha_j-2}, \ldots X_{i-2}^{l_1}, X_{i-1}^{l_2} | X_{\alpha_j-2}^{\dagger}, X_{\alpha_j-1}^{h_j}) H(X_i | X_{i-2}^{l_1}, X_{i-1}^{l_2}) \\ &= p(L_3^{i-\alpha_j} X_{i-2}^{l_1}, X_{i-1}^{l_2} | X_{\alpha_j-2}^{\dagger}, X_{\alpha_j-1}^{h_j}) H(X_i | X_{i-2}^{l_1}, X_{i-1}^{l_2}) \end{aligned} \tag{25}$$

Hence,

$$H_3^I = \sum_{j \in J} \sum_{i \in I_j} p(L_3^{i-\alpha_j} X_{i-2}^{l_1}, X_{i-1}^{l_2} | X_{\alpha_j-2}^{\dagger}, X_{\alpha_j-1}^{h_j}) H(X_i | X_{i-2}^{l_1}, X_{i-1}^{l_2}) \tag{26}$$

By using (26), the entropy of the different states in Fig. 1 becomes

1. $H_3^I = 0$
2. $H_2^I = H(X_0)$
3. $H_2^I = p(X_0^{l_2} | X_0^{h_1}) H(X_1 | X_0^{l_2})$
4. $H_2^I = p(X_0^{l_1}, X_1^{l_2} | X_0^{\dagger}, X_1^{h_1}) H(X_2 | X_0^{l_1}, X_1^{l_2})$
5. $H_2^I = H(X_0, X_1)$
6. $H_2^I = H(X_0) + p(X_0^{l_1}, X_1^{l_2} | X_0^{\dagger}, X_1^{h_2}) H(X_2 | X_0^{l_1} X_1^{l_2})$
7. $H_2^I = p(X_0^{l_2} | X_0^{h_1}) H(X_1 | X_0^{l_2}) + p(X_0^{l_1}, X_1^{l_2} | X_0^{h_1}) H(X_2 | X_0^{l_1}, X_1^{l_2})$
8. $H_2^I = H(X_0, X_1, X_2)$

4.4 *n*-order Languages

For L_n languages the probability distribution of the symbols depends on $n-1$ preceding symbol. Thus, as before, from the alphabet $\mathbb{L}^1 = L^1 \bigcup \{\delta\}$, language $\mathbb{L}_n$ is constructed as an extension of language L_n by setting $p(\delta^{n-1}) = 1$, $p(\delta|\gamma_i) = 0$ and $\forall \gamma_i \in L^1$, $p(\gamma_i | \delta^{n-1}) = p(\gamma_i)$ Thus, all strings in $\mathbb{L}_n$ start with the independent substring δ^{n-1} and then continue as in L_n.

To shorten the notation in the following expressions

$$\mathbb{X}_{i-1}^{n-1} = X_{i-(n-1)}^{l_1}, \ldots, X_{i-1}^{l_{n-1}} \tag{27}$$

and

$$\mathbb{Y}_{\alpha_j-1}^{n-1} = X_{\alpha_j-(n-1)}^{\dagger}, \ldots, X_{\alpha_j-1}^{h_j} \tag{28}$$

By using (27) and (28) the conditional entropies in (14) can now be written for L_n languages as

$$H_n^i = p(L_n^{i-\alpha_j} \mathbb{X}_{i-1}^{n-1} | Y_{i-1}^{n-1}) H(X_i | \mathbb{X}_{i-1}^{n-1}) \tag{29}$$

Hence,

$$H_n^I = \sum_{j \in J} \sum_{i \in I_j} p(L_n^{i-\alpha_j} \mathbb{X}_{i-1}^{n-1} | \mathbb{Y}_{\alpha_j-1}^{n-1}) H(X_i | \mathbb{X}_{i-1}^{n-1}) \tag{30}$$

5 Concluding Remarks

We have in this paper continued the investigation of the confidentiality implications of selective encryption by applying entropy on a generic selective encryption scheme. By using the concept of run-length vector from run-length encoding theory, an expression was derived for entropy of selectively encrypted strings when the number of encrypted substrings, containing one symbol, and the order of the language change.

To further understand the confidentiality implication of selective encryption we will investigate how entropy changes when the substrings are of different sizes larger than one. Moreover, the conditional probabilities in the paper are used left to right. That is, if $b = (0, 1)$, then the first string gives information about the second string. However, if $b = (1, 0)$, then the second string also gives information about the first string. In our future work we will also aim to investigate how the entropy changes for different sources of information and appearances of the bit vector.

References

1. T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, NY, USA, 1991.
2. G. B. Folland. *Real Analysis, Modern Techniques and Their Applications.* John Wiley & Sons, New York, NY, USA, 1999.
3. J. Goodman and A. P. Chandrakasan. Low power scalable encryption for wireless systems. *Wireless Networks*, 4(1):55–70, 1998.
4. S. Lindskog, R. Lundin, and A. Brunstrom. Middleware support for tunable encryption. In *Proceedings of the 5th International Workshop on Wireless Information Systems (WIS 2006)*, May 23, 2006.
5. R. Lundin and S. Lindskog. Security implications of selective encryption. In *Proceedings of the 6th International Workshop on Security Measurements and Metrics (MetriSec 2010)*, Bolzano, Italy, September 15, 2010.
6. M. Podesser, H. P. Schmidt, and A. Uhl. Selective bitplane encryption for secure transmission of image data in mobile environments. In *Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02)*, Tromsø/Trondheim, Norway, October 4–6, 2002.
7. Z. Shahid, M. Chaumont, and W. Puech. Fast protection of H.264/AVC by selective encryption of CABAC for I & P frames. In *Proceedings of the 17th European Signal Processing Conference (EUSIPCO 2009)*, pages 2201–2205, Glasgow, Scotland, August 24–28, 2009.
8. C. E. Shannon. *Claude Elwood Shannon: Collected Papers.* IEEE Press, Piscataway, NJ, USA, 1993.
9. G. A. Spanos and T. B. Maples. Performance study of a selective encryption scheme for security of networked, real-time video. In *Proceedings of the 4th International Conference on Computer Communications and Networks (ICCCN'95)*, pages 72–78, Las Vegas, Nevada, USA, September 1995.
10. Z. Su, J. Jiang, S. Lian, G. Zhang, and D. Hu. Hierarchical selective encryption for G.729 speech based on bit sensitivity. *Journal of Internet Technology*, 10(5):599–608, 2010.