Security of Compressing Encrypted Sources

G. Jakimoski and K. P. Subbalakshmi

Abstract

When transmitting redundant data over an insecure channel, it is customary to first compress the data and then encrypt it. Johnson et al have investigated a cryptosystem where the order of these steps is reversed, and they have shown that this reversal is possible. We formally study the security of the encrypt-then-compress cryptosystems and show that the compression does not compromise the security of the system in both the information theoretic case (perfectly secure encryption) and the complexity theoretic case (computationally secure encryption).

I. Introduction

A common approach to secure transmission of redundant data is to first compress the data, and then encrypt it. However, there are some situations where the reversed system might be preferable. Consider for example a content distribution scenario where the content owner and the network operator are two distinct entities, and do not trust each other. The content owner is interested in protecting the privacy of the content via encryption, but he has no incentive to use his limited computational resources to run a compression algorithm before encrypting the data. Since the content owner does not trust the network operator, he will not supply the cryptographic key that was used to encrypt the data. So, in order to maximize the network utilization, the network owner is forced to compress the data after it has been encrypted.

The previous example was used by Johnson et al [1] as a motivation to study the problem of compressing encrypted data. Johnson et al show that the reversal of order is possible in some settings of interest without loss of either optimal coding efficiency or Wyner-sense [2] perfect secrecy. They also remark that Shannon-sense [5] perfect secrecy is also achievable.

We formally study the security of the encrypt-then compress schemes. We show that the application of a compression transformation after encryption does not compromise the security of the system. We use the stronger Shannon-sense perfect secrecy definition in the case of unconditional security, and we use left-or-right (LOR) indistinguishability [6] in the case of computational security.

The paper is organized as follows. In Section 2, we describe the security notions used in our analysis. Section 3 deals with the security of encrypt-then-compress schemes. The paper ends with concluding remarks.

II. Preliminaries

In this section, we describe some common security notions for encryption schemes.

A. Symmetric encryption schemes

A *symmetric encryption scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a triple of a *key generation* algorithm $\mathcal{K}$, an *encryption* algorithm $\mathcal{E}$ and a *decryption* algorithm $\mathcal{D}$. The key generation algorithm $\mathcal{K}$ is randomized. It takes as input a security parameter $k \in \mathsf{N}$ and returns a key K. The encryption algorithm $\mathcal{E}$ can be either randomized or stateful. It takes as input a *plaintext* M and the secret key K, and outputs a *ciphertext* C. The decryption algorithm $\mathcal{D}$ is deterministic and stateless. It takes as input a ciphertext C and the secret key K. It outputs either the corresponding plaintext M or a special symbol $\mathsf{Invalid}$. We require that $\mathcal{D}_K(\mathcal{E}_K(M)) = M$ for any message M.

B. Perfect secrecy

In the case of unconditional security, the key produced by the key generation algorithm is used only once. In this case, the encryption scheme has perfect secrecy if given a ciphertext C even a computationally unbounded adversary cannot learn anything about the plaintext M. That is, the a posteriori probability that the plaintext is M, given the ciphertext C is observed, is identical to the a priori probability that the plaintext is M, i.e.:

$$\Pr[M|C] = \Pr[M],$$

for all $M \in \mathcal{P}$, $C \in \mathcal{C}$, where $\mathcal{P}$ is the set of all possible plaintexts, and $\mathcal{C}$ is the set of all possible ciphertexts.

The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA, e-mail: ksubbala@stevens.edu, goce.jakimoski@stevens.edu.

This work was supported in part by the National Science Foundation under the grant NSF 0627688.

C. Left-or-right indistinguishability

We use the notion of left-or-right indistinguishability [6] to define a secure symmetric encryption scheme. We consider an adversary that has access to a *left-or-right encryption oracle* $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$, $b \in \{0,1\}$, which takes as input a pair of plaintexts (x_0, x_1) and does the following: if $b = 0$ it returns $\mathcal{E}_K(x_0)$; otherwise, it returns $\mathcal{E}_K(x_1)$. The adversary is allowed to make a query (x_0, x_1) only if x_0 and x_1 are equal-length plaintexts. Its goal is to guess the bit b. We say the encryption scheme is secure in an IND-CPA (indistinguishability under chosen-plaintext attacks) sense if no efficient adversary can distinguish the cases $b = 0$ and $b = 1$. See [6] for a more formal definition.

III. Security of the encrypt-then-compress schemes

Figure 1 depicts a cryptosystem where the encryption precedes compression. The plaintext X is first encrypted using a secret key K into a ciphertext Y. Then, the ciphertext Y is transformed using a compression transformation to get the compressed ciphertext Z. The decompression and decryption are done jointly at the receiver's side using the secret key K. Using the security notions presented in Section 2, we show that the overall system remains secure if the encryption scheme that encrypts X into Y is secure.

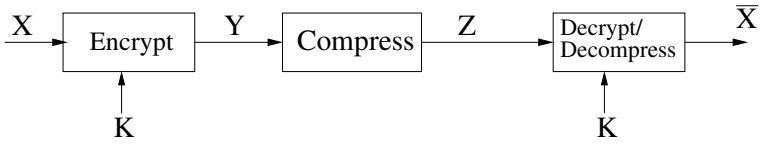

Fig. 1. A cryptosystem where the compression is done after the encryption. The plaintext X is encrypted using a key K into a ciphertext Y. The ciphertext Y is compressed using a linear compression transformation L to get the compressed ciphertext Z. At the receiver, the decompression and decryption are performed jointly using the secret key K.

The unconditional security case. We note that if the encryption is done using one-time pad, then the resulting encrypt-then-compress scheme has perfect secrecy as well.

Theorem 1: If the encryption scheme that is used to encrypt X into Y has perfect secrecy, then the overall encrypt-then-compress cryptosystem of Figure 1 has perfect secrecy too.

Proof. We use $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ to denote random variables taking values in the set of possible plaintexts, the set of possible ciphertexts and the set of possible compressed ciphertexts respectively. For the a posteriori probability $\Pr[\mathbf{X} = \mathbf{X}|\mathbf{Z} = \mathbf{Z}]$ that the plaintext is X, given the compressed ciphertext Z is observed, we get:

$$\begin{aligned}
\Pr[\mathbf{X} = \mathbf{X}|\mathbf{Z} = \mathbf{Z}] &= \sum_{Y} \Pr[\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}|\mathbf{Z} = \mathbf{Z}] \\
&= \sum_{Y} \Pr[\mathbf{X} = \mathbf{X}|\mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}] \Pr[\mathbf{Y} = \mathbf{Y}|\mathbf{Z} = \mathbf{Z}] \\
&= \sum_{Y} \Pr[\mathbf{X} = \mathbf{X}] \Pr[\mathbf{Y} = \mathbf{Y}|\mathbf{Z} = \mathbf{Z}] \\
&= \Pr[\mathbf{X} = \mathbf{X}] \sum_{\mathbf{Y}} \Pr[\mathbf{Y} = \mathbf{Y}|\mathbf{Z} = \mathbf{Z}] \\
&= \Pr[\mathbf{X} = \mathbf{X}]
\end{aligned}$$

The probabilities are computed over the random keys used by the encryption scheme, and over the randomness used by the compression function if the compression function is probabilistic. We used the observation that $\Pr[\mathbf{X} = \mathbf{X}|\mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}] = \Pr[\mathbf{X} = \mathbf{X}]$, which we prove below.

$$\begin{aligned}
\Pr[\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}] &= \Pr[\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}] \Pr[\mathbf{Z} = \mathbf{Z}|\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}] \\
&= \Pr[\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}] \Pr[\mathbf{Z} = \mathbf{Z}|\mathbf{Y} = \mathbf{Y}] \\
\Pr[\mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}] &= \Pr[\mathbf{Y} = \mathbf{Y}] \Pr[\mathbf{Z} = \mathbf{Z}|\mathbf{Y} = \mathbf{Y}] \\
\Pr[\mathbf{X} = \mathbf{X}|\mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}] &= \frac{\Pr[\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}]}{\Pr[\mathbf{Y} = \mathbf{Y}, \mathbf{Z} = \mathbf{Z}]} = \frac{\Pr[\mathbf{X} = \mathbf{X}, \mathbf{Y} = \mathbf{Y}]}{\Pr[\mathbf{Y} = \mathbf{Y}]} \\
&= \Pr[\mathbf{X} = \mathbf{X}|\mathbf{Y} = \mathbf{Y}] = \Pr[\mathbf{X} = \mathbf{X}]
\end{aligned}$$

Here, we used the fact that the value of Z depends only on the value of Y, and $\Pr[\mathbf{X} = \mathbf{X}|\mathbf{Y} = \mathbf{Y}] = \Pr[\mathbf{X} = \mathbf{X}]$ due to the perfect secrecy of the scheme that encrypts X into Y. □

The computational security case. One can also show that if the encryption used to encrypt X into Y is computationally secure (e.g., a secure stream cipher), then the resulting encrypt-then-compress scheme is computationally secure as well.

Theorem 2: Suppose X is encrypted into Y using an IND-CPA encryption scheme, then the overall encrypt-then-compress cryptosystem of Figure 1 is IND-CPA too.

Proof. We show the previous claim by contradiction. Suppose there is an adversary E that can break the overall encrypt-then-compress scheme in the IND-CPA sense. In that case, we can construct an adversary A that can break the encryption scheme which is used to encrypt X into Y.

A will simulate E's oracle as follows. Whenever, E asks a query (X_0, X_1), A will answer that query by sending a query to its own left-or-right encryption oracle $\mathcal{E}_K(\mathcal{LR}(\cdot, \cdot, b))$. Suppose Y is the output of A's encryption oracle. A will answer E's query by compressing Y into $Z = L(Y)$ and writing Z on the E's query tape. When E halts, it will output its guess about the value of the bit b. E's guess will be A's guess as well.

Suppose E asks q_e encryption queries, runs in time t and has advantage ϵ of distinguishing the cases $b = 0$ and $b = 1$. It is not hard to verify that A will also ask q_e encryption queries, it will run in ct time, where c is a small implementation dependent constant, and it will have advantage ϵ of distinguishing the cases $b = 0$ and $b = 1$. This contradicts our assumption that the encryption scheme used to encrypt X into Y is computationally secure (i.e., IND-CPA using LOR indistinguishability definition). □

IV. Conclusion

We have formally investigated the security of systems where the redundant data is compressed after its encryption. We show that if the original encryption scheme is secure, then the overall system is secure as well. In other words, the compression does not compromise the security of the system.

References

[1] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On Compressing Encrypted Data," IEEE Transactions on Signal Processing, vol. 52, no. 10, pp. 2992–3006, 2004.

[2] A. D. Wyner, The wire-tap channel, Bell Syst. Tech. J., vol. 54, no. 8, pp. 13551387, 1975.

[3] D. Schonberg, S. Draper, and K. Ramchandran, "On Compression of Encrypted Images," International Conference on Image Processing, Atlanta, GA, October 2006.

[4] D. Schonberg, C. Yeo, S. C. Draper, and K. Ramchandran, "On Compression of Encrypted Video," Proceedings of Data Compression Conference 2007, pp. 173 - 182.

[5] C. E. Shannon, Communication theory of secrecy systems, Bell Syst. Tech. J., vol. 28, 1949.

[6] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption," Proceedings of 38th Annual Symposium on Foundations of Computer Science, IEEE, 1997.