Secure Length-Preserving All-or-Nothing Transform

Kuwakado, Hidenori

Tanaka, Hatsukazu

(Citation)
情報処理学会論文誌 : 論文誌ジャーナル(IPSJ Journal),46(8):1843-1851

(Issue Date)
2005-08-15

(Resource Type)
journal article

(Version)
Version of Record

Recommended Paper

Secure Length-Preserving All-or-Nothing Transform

HIDENORI KUWAKADO† and HATSUKAZU TANAKA††

When a hard drive (HDD) is recycled, it is recommended that all files on the HDD are repeatedly overwritten with random strings for protecting their confidentiality. However, it takes a long time to overwrite them. This problem is solved by applying the all-or-nothing transform (AONT) to the filesystem of the HDD. To use the HDD economically, it is desirable to use a length-preserving AONT (LP-AONT). Whereas previous AONTs cause the increase of size of a file, and no LP-AONT is secure under previous security definitions. However, it does not mean that the LP-AONT is useless; previous security definitions are too strict in practical applications. Then, by introducing the ambiguity of a message, we propose more practical security definitions of the AONT. We also show the secure implementation of the LP-AONT under the proposed security definitions. The analysis shows that our implementation is nearly optimal in terms of the success probability of an adversary. It means that the ambiguity of one message block allows us to construct the LP-AONT as secure as previous AONTs.

1. Introduction

The secure management of files on the hard drive (HDD) is one of important problems in the area of information security. In particular all files on the HDD should be completely deleted before the HDD is recycled. To make it harder for even expensive HDD probing to recover files, we usually use special software (e.g., **shred** in the GNU Core Utilities [4]), which repeatedly overwrite files with a random (or fixed) string. The Japan Electronics and Information Technology Industries Association [11] recommends that the HDD is overwritten with the fixed string twice. The Under Secretary of Defense [12] has defined the rule that the HDD is overwritten with a string, its complement, then with a random string. However, it probably takes a long time to overwrite all files on the HDD with the string.

One may think that this problem can be solved by using an encrypted filesystem such that the file is automatically encrypted at writing, and it is automatically decrypted at reading. Since all files on the HDD have been encrypted, it is unnecessary to overwrite them with the random string. However, the encrypted filesystem requires the secure management of the decryption key. If the decryption key is easily guessed, then the encrypted filesystem is useless for protecting files on the HDD.

The use of the all-or-nothing encryption mode in the encrypted filesystem is effective in reducing the overwriting time and avoiding the decryption-key management. The all-or-nothing encryption mode, which consists of the all-or-nothing transform (AONT) and the usual encryption, has been originally proposed for improving the security against brute force attacks [9], and is also applied to the construction of efficient fixed-blocksize encryption schemes [5]. The remarkable property of the all-or-nothing encryption mode is that it is infeasible to find out information about message blocks if even one ciphertext block is lost. Overwriting a part of the encrypted file with the random string can be considered as the loss of the ciphertext blocks. Therefore, if a small part of the encrypted file is overwritten with the random string, then an adversary cannot obtain any information about the file even if the adversary knows the decryption key.

Since the remarkable property is achieved by the AONT, a simplified scheme without the decryption key is possible. Namely, only the AONT is applied to the filesystem. In usual use, at the recording, the file is automatically transformed by the AONT, and the resulting data are recorded on the HDD, and at the reading, the file is automatically recovered by the inverse AONT. When a user recycles the HDD, it is sufficient that the user overwrites some parts of files on the HDD.

In both cases we see that the AONT plays an

† Kobe University
†† Kobe Institute of Computing

The initial version of this paper was presented at Computer Security Symposium (CSS2004) held on Oct. 2004, which was sponsored by SIGCSEC. This paper was recommended to be submitted to IPSJ Journal by the chairperson of SIGCSEC.

essential role in reducing the overwriting time. We hence focus on the AONT. In this paper, the AONT such that the size of data increases is called a length-increasing AONT (LI-AONT), and the AONT such that it does not increase is called a length-preserving AONT (LP-AONT). In the application of the filesystem, it is desirable to utilize the LP-AONT for the economical use of the HDD.

The concept of the AONT has been originally introduced by Rivest [9]. After then, the AONT has been studied in several models. Boyko [2] has studied the AONT in the random oracle model. Stinson [10] has done it in the information-theoretic model. Canetti, Dodis, Halevi, Kushilevitz, and Sahai [3] have done it in the standard model.

Boyko [2] has shown that the optimal asymmetric encryption padding (OAEP) [1] is the most secure AONT under the random oracle model. Note that the use of the OAEP causes the increase of the size of data, i.e., the OAEP is the LI-AONT. The AONT proposed by Canetti, Dodis, Halevi, Kushilevitz, and Sahai [3] is also the LI-AONT. Stinson's AONT [10] is the LP-AONT, but the security problem has been pointed out by Boyko. Thus, no secure LP-AONT has been proposed in spite of the fact that the LP-AONT is better than the LI-AONT in terms of the size of data.

Boyko [2] has provided security definitions of the AONT under the random oracle model, which are considered as the formal version of Rivest's definition [9]. Surprisingly, no LP-AONT is secure under the Boyko's definitions. However, it probably suggests that Byoko's definitions are too strict rather than the LP-AONT is totally useless.

Hence, we discuss security definitions and implementations of the LP-AONT. We give new security definitions under the random oracle model suitable for evaluating the security of the LP-AONT in practical applications. Our security definitions explicitly introduce the ambiguity of the message. It follows that our security definitions are slightly weaker than Boyko's definitions, but we believe that they provide sufficient security in practical applications. Boyko discussed the relation between the number of lost *bits* of the output and the size of leaked information about the whole message. In contrast, we discuss the size of leaked information about the message when one *block* of the output is lost. The reason for considering the loss of the block is that a user can easily overwrite the file with the random string in blocks (e.g., the byte of the file, the sector on the HDD) rather than in bit. Moreover, we show a secure implementation of the LP-AONT under the new security definitions. We prove that our implementation is nearly optimal in terms of the success probability of the adversary.

The AONT is closely related to a threshold secret sharing scheme [9]. For example, the LI-AONT proposed by Rivest [9] is an s-out-of-s computational secret sharing scheme. It is known that the size of a share in the s-out-of-s computational secret sharing scheme is asymptotically $1/s$ of the size of the secret information [6]. In other words, the size of a share in previous s-out-of-s computational secret sharing schemes is always larger than $1/s$ of the size of the secret information. The LP-AONT, which is discussed in this paper, is the s-out-of-s secret sharing scheme based on the random oracle such that the size of a share is precisely $1/s$ of the size of the secret information. Hence, the LP-AONT is useful for applications of the s-out-of-s secret sharing scheme.

This paper is organized as follows. In Section 2, we summarize notation. In Section 3, we describe previous definitions and implementations of the AONT. In Section 4, we give new security definitions, which are suitable for estimating the security of the LP-AONT. In Section 5, we propose the implementation of the LP-AONT. In Section 6, we analyze the security of the proposed implementation, and prove the optimality of the implementation, in terms of the success probability of the adversary. In Section 7, we conclude this paper and describe an open problem.

2. Notation

For an algorithm A, we denote by $A()$ the distribution of A's output on inputs. When A and B are algorithms, we denote by $A^B()$ the distribution of A's output on inputs when A uses B as an oracle. We denote by $c \xleftarrow{R} C$ to choose c at random according to the distribution C, and denote by $c \leftarrow C$ to set x to the result of evaluating expression C. For a set S, we denote by $s \xleftarrow{R} S$ to choose s uniformly at random from S. We denote by

$$\Pr[a \xleftarrow{R} A(), \ldots : p(a, \ldots)]$$

the probability that a predicate $p(a, \ldots)$ is true

after $a \stackrel{R}{\leftarrow} A(), \ldots$ We denote by

$$\mathrm{Ev}[a \stackrel{R}{\leftarrow} A(), \ldots : f(a, \ldots)]$$

the expected value of $f(a, \ldots)$ after $a \stackrel{R}{\leftarrow} A(), \ldots$ To specify the distribution of the random oracle G, we write $G \stackrel{R}{\leftarrow} \Omega$ where Ω is the set of all maps from finite strings to the set of infinite strings.

For a block sequence

$$\mathbf{x} = (x_1, x_2, \ldots, x_s)$$

where $x_i \in \{0,1\}^\ell$, we denote by $\mathbf{x}\backslash x_u$ the block sequence such that x_u is replaced with an empty block ψ, i.e.,

$$\mathbf{x}\backslash x_u = (x_1, x_2, \ldots, x_{u-1}, \psi, x_{u+1}, \ldots, x_s).$$

The empty block ψ means that the block at the position is lost (or unknown). We call the sequence containing ψ the incomplete sequence. We let $\|$ denote the concatenation operator on blocks.

3. Previous Definitions and Implementations

Rivest [9] has defined the all-or-nothing transform as follows:

Definition 1 We say that a transform F mapping a message sequence $(x_1, x_2, \ldots, x_s)$ into a pseudo-message sequence $(y_1, y_2, \ldots, y_{s'})$ is an all-or-nothing transform (AONT) if F satisfies the following conditions.

(1) The transform F is invertible.
(2) Both of F and its inverse are efficiently computable.
(3) It is computationally infeasible to compute any function of any message block if any one of the pseudo-message blocks is unknown.

The transform F is called a length-preserving AONT (LP-AONT) if $s = s'$, and it is called a length-increasing AONT (LI-AONT) if $s < s'$.

The OAEP [1] and the package transform [9] are implementations of the LI-AONT. The OAEP is based on the random oracle model, and the package transform is based on the ideal cipher model. The LI-AONT based on the standard model has been also proposed [3].

Stinson [10] has modified Definition 1 to provide unconditional security as follows:

Definition 2 Let $X_1, X_2, \ldots, X_s$ be random variables taking on values in $\{0,1\}^\ell$, and let F denote a function of them. Suppose that random variables $Y_1, Y_2, \ldots, Y_s$ are given by $F(X_1, X_2, \ldots, X_s)$. We say that F is an unconditional AONT if random variables satisfy the following conditions where H denotes the entropy function.

(1) $\mathrm{H}(Y_1, Y_2, \ldots, Y_s | X_1, X_2, \ldots, X_s) = 0$.
(2) $\mathrm{H}(X_1, X_2, \ldots, X_s | Y_1, Y_2, \ldots, Y_s) = 0$.
(3) $\mathrm{H}(X_i | Y_1, Y_2, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_s) = \mathrm{H}(X_i)$ for $i, j = 1, 2, \ldots, s$.

Stinson [10] has shown the unconditional AONT F based on the linear function over a finite field $\mathrm{GF}(q)$. Let D denote an invertible $s \times s$ matrix on $\mathrm{GF}(q)$ such that no entry of D is equal to 0. For a message sequence $\mathbf{x} = (x_1, x_2, \ldots, x_s)$ where $x_i \in \mathrm{GF}(q)$, F is defined as

$$F(x_1, x_2, \ldots, x_s) = \mathbf{x}D^{-1} \text{ over } \mathrm{GF}(q), \quad (1)$$

where $\mathbf{x}$ is regarded as a vector and D^{-1} is the inverse matrix of D.

For the third conditions of Definition 1 and Definition 2, Boyko [2] pointed out that the conditions only considered the amount of information leaked about a particular message block, as opposed to the whole message. To solve this problem, Boyko has given formal security definitions of the AONT for the whole message in terms of semantic security and indistinguishability. As an example, we mention Boyko's definition of the non-adaptive indistinguishability.

Definition 3 Let F be a transform mapping an n-bit message to an n'-bit pseudo message and using the random oracle Γ. Let $\mathcal{L}$ be a set of lost bit positions of the pseudo message. There are the find stage for finding two messages x_0, x_1 and the guess stage for guessing which message was transformed. Auxiliary data d are used for transmitting information from the find stage to the guess stage. Alice A is said to succeed in (T, q_Γ, ϵ)-distinguishing F if there exists $\mathcal{L}$ such that

$$\Pr\Big[\Gamma \stackrel{R}{\leftarrow} \Omega,\ (x_0, x_1, d) \stackrel{R}{\leftarrow} A^\Gamma(\mathcal{L}, \mathsf{find}),\ b \stackrel{R}{\leftarrow} \{0,1\},\ y \stackrel{R}{\leftarrow} F^\Gamma(x_b) : A^\Gamma(d, \widetilde{y}, \mathsf{guess}) = b\Big] \geq \frac{1}{2} + \epsilon,$$

and Alice runs for at most T steps and makes at most q_Γ queries to Γ, where $\widetilde{y}$ is the string such that bits of y are lost according to $\mathcal{L}$.

If ϵ is negligibly small, then it is considered

that F is secure because $\widetilde{y}$ does not give even one bit about the message. Boyko proved that no AONT can achieve substantially better security than the OAEP in the sense of this definition.

Boyko's definition considers the loss in bits, as opposed to the loss in blocks. Even if Boyko's definition is modified to the loss in blocks, no LP-AONT is secure. Since the LP-AONT is deterministic, it is possible to guess b by computing $F(x_0)$ and $F(x_1)$. We also see that no LP-AONT is secure in the sense of Boyko's semantic security.

On the other hand, one may consider that the third condition of Definition 2 is information-theoretically extended to the whole message. The following condition guarantees the security of the whole message: for $j = 1, 2, \ldots, s$,

$$\begin{aligned} &\mathrm{H}(X_1, X_2, \ldots, X_s | \\ &\quad Y_1, Y_2, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_s) \\ &\quad = \mathrm{H}(X_1, X_2, \ldots, X_s). \end{aligned} \tag{2}$$

However, it has been shown that the AONT satisfying Eq. (2) is the LI-AONT [7].

4. New Definitions

The LP-AONT F is a deterministic permutation on $\{0,1\}^{\ell s}$ where ℓ is the length of one block and s is the number of blocks. As described in Section 3, F is not secure in terms of the indistinguishability of Boyko's definition. However, we do not consider that the LP-AONT is completely insecure. Although Boyko's definitions give the formal security of information about the message as a whole, they probably are too strict in practical situation. In other words, the situation that Alice completely knows both of x_0 and x_1 is too advantageous to Alice. Thus, we relax Boyko's definitions by introducing the ambiguity of the message.

Let us consider the following game done by Alice (adversary) and Bob in terms of the indistinguishability. Let F be the LP-AONT based on the random oracle Γ.

Find stage: Alice is given u, v and access to the random oracle Γ. She outputs two incomplete message sequences $\mathbf{x}_0\backslash x_u$, $\mathbf{x}_1\backslash x_u$ and auxiliary data d. She gives the two incomplete message sequences to Bob. Bob chooses a random bit $b \in \{0,1\}$. He chooses $x_u \in \{0,1\}^\ell$ at random and substitutes x_u into the u-th block of $\mathbf{x}_b\backslash x_u$. He computes

$$\mathbf{y}\backslash y_v = F^\Gamma(\mathbf{x}_b)\backslash y_v.$$

and gives $\mathbf{y}\backslash y_v$ to Alice.

Guess stage: Given d and $\mathbf{y}\backslash y_v$, Alice has access to Γ. She has to guess b.

Alice uses the auxiliary data d to transmit information from the find stage to the guess stage. For example, d probably contains the incomplete message sequences and queries to the random oracle Γ. Note that Alice does not completely know two candidates of the message. In this respect Definition 4 differs from Definition 3. If Alice's probability of correctly guessing b is $1/2$, then the AONT F is secure in terms of the indistinguishability.

Using the application described in Section 1, we explain the security given by the above game. The incomplete message sequences $\mathbf{x}_0\backslash x_u$, $\mathbf{x}_1\backslash x_u$ are files in compliance with known formats. For example, Alice knows that the first line of the Portable Document Format (PDF) file is "%PDF-1.2" and that of the PostScript file is "%!PS-Adobe-2.0." The incomplete pseudo-message sequence $\mathbf{y}\backslash y_v$, which was obtained by Alice, is the no-overwritten part of the file on the HDD. Alice wants to know whether the file is the PDF file or the PostScript file, and wants to obtain information about the overwritten part of the file if possible. However, if the probability that Alice's guess is correct is $1/2$, then it means that she cannot distinguish between the PDF file and the PostScript file. Since even the known part $\mathbf{x}_b\backslash x_u$ is completely hidden, it is not easier to extract the unknown part x_u from $\mathbf{y}\backslash y_v$. Namely, the knowledge of file formats is useless to obtain any information about the unknown part of the message sequence from the incomplete pseudo-message sequence.

The formal definition of indistinguishability based on the above game is given as follows:

Definition 4 Let F be a transform mapping $\mathbf{x}=(x_1, x_2, \ldots, x_s)$ into $\mathbf{y}=(y_1, y_2, \ldots, y_s)$ and using the random oracle Γ. Let u, v $(1 \le u, v \le s)$ be integers. Alice A is said to succeed in (T, q_Γ, ϵ)-distinguishing F if there exist u and v such that

$$\mathrm{Pr}\Big[\Gamma \xleftarrow{R} \Omega,$$

$$(\mathbf{x}_0\backslash x_u, \mathbf{x}_1\backslash x_u, d) \xleftarrow{R} A^\Gamma(u, v, \mathsf{find}),$$

$$b \xleftarrow{R} \{0,1\},\ x_u \xleftarrow{R} \{0,1\}^\ell,$$

$$\mathbf{y}\backslash y_v \xleftarrow{R} F^{\Gamma}(\mathbf{x}_b)\backslash y_v : A^{\Gamma}(d, \mathbf{y}\backslash y_v, \mathsf{guess}) = b \Big] \geq \frac{1}{2} + \epsilon,$$

and Alice runs for at most T steps and makes at most q_Γ queries to Γ.

Next, let us consider the following game in terms of the semantic security.

Find stage: Alice is given u, v and access to the random oracle Γ. She outputs $\mathbf{x}\backslash x_u$, and gives it to Bob. Bob chooses $x_u \in \{0,1\}^\ell$ at random, and substitutes x_u into the u-th block of $\mathbf{x}\backslash x_u$. He computes

$$\mathbf{y}\backslash y_v = F^{\Gamma}(\mathbf{x})\backslash y_v,$$

and gives $\mathbf{y}\backslash y_v$ to Alice.

Guess stage: Given u, v and $\mathbf{y}\backslash y_v$, Alice has access to Γ. She has to guess $f(\mathbf{x})$ where f is an arbitrary deterministic function.

Alice cannot transmit any information about $\mathbf{x}\backslash x_u$ from the find stage to the guess stage. Otherwise she will transmit the value of $f(\mathbf{x}\backslash x_u)$ to the guess stage. Note that she outputs the incomplete message sequence in the find stage. This respect is different from the semantic security of Boyko's definitions. If Alice cannot guess the value of $f(\mathbf{x})$ substantially better than always outputting the most probable value of $f(\mathbf{x})$, then she cannot obtain any useful information from the incomplete pseudo-message sequence.

Using the application described in Section 1, we explain the security given by the above game. Suppose that Alice obtains an incomplete digital image file $\mathbf{y}\backslash y_v$ by the HDD proving. Since Alice does not know what the image is, she attempts to recover the image even if the resolution of the recovered image is low. In this case, $f(\mathbf{x})$ means the low-resolution image of the image $\mathbf{x}$. However, if Alice cannot guess $f(\mathbf{x})$ substantially better than always outputting the most probable $f(\mathbf{x})$, then she cannot obtain any new information about the image.

The formal definition of the semantic security based on the above game is given as follows:

Definition 5 Let F be a transform mapping $\mathbf{x}=(x_1, x_2, \ldots, x_s)$ into $\mathbf{y}=(y_1, y_2, \ldots, y_s)$ and using the random oracle Γ. Let u, v $(1 \leq u, v \leq s)$ be integers, and let f denote any deterministic function. Alice A is said to succeed in (T, q_Γ, ϵ)-predicting f if there exist u, v such that

$$\Pr\Big[\Gamma \xleftarrow{R} \Omega,\ \mathbf{x}\backslash x_u \xleftarrow{R} A^{\Gamma}(u, v, \mathsf{find}),\ x_u \xleftarrow{R} \{0,1\}^\ell,\ \mathbf{y} \leftarrow F^{\Gamma}(\mathbf{x}) : A^{\Gamma}(u, v, \mathbf{y}\backslash y_v, \mathsf{guess}) = f(\mathbf{x})\Big] \geq p_f + \epsilon,$$

where

$$p_f = \mathrm{Ev}\big[\Gamma \xleftarrow{R} \Omega : \max_z \Pr\big[\mathbf{x}\backslash x_u \xleftarrow{R} A^{\Gamma}(u, v, \mathsf{find}),\ x_u \xleftarrow{R} \{0,1\}^\ell : f(\mathbf{x}) = z\big]\big]$$

and A runs for at most T steps and makes at most q_Γ queries to Γ.

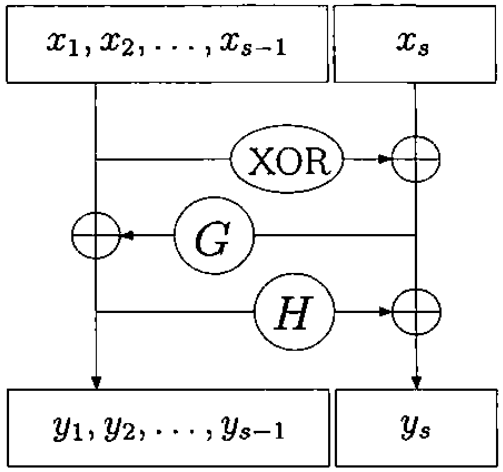

Fig. 1 Proposed LP-AONT.

Stinson's LP-AONT given by Eq. (1) does not use the random oracle, i.e., the linear transform D is explicitly given. Stinson's LP-AONT is not secure in the sense of Definition 4 and Definition 5 because of the linear transform. For given D, it is easy to find $\mathbf{x}_0\backslash x_u$ and $\mathbf{x}_1\backslash x_u$ such that ϵ of Definition 4 is equal $1/2$. It is also easy to construct an algorithm A such that ϵ of Definition 5 is not negligibly small.

5. Implementation

We propose a new LP-AONT as shown in **Fig. 1**. Let $\mathbf{x} = (x_1, x_2, \ldots, x_s)$ be the message sequence where $x_i \in \{0,1\}^\ell$. Let G be a mapping $\{0,1\}^\ell$ into $\{0,1\}^{\ell(s-1)}$, and let H denote a mapping $\{0,1\}^{\ell(s-1)}$ into $\{0,1\}^\ell$. An LP-AONT F is defined as

$$F(\mathbf{x}) = y_1 \| y_2 \| \ldots \| y_s = (x_L \oplus G(r)) \| (r \oplus H(x_L \oplus G(r))),$$

where

$$x_L = x_1 \| x_2 \| \cdots \| x_{s-1}, \quad (3)$$
$$r = x_1 \oplus x_2 \oplus \cdots \oplus x_s. \quad (4)$$

The pseudo-message sequence is given by

$$\mathbf{y} = (y_1, y_2, \ldots, y_s),$$

where $y_i \in \{0,1\}^\ell$. The inverse of F is computed straightforwardly. Thus, F is the efficiently invertible permutation on $\{0,1\}^{\ell s}$.

The proposed LP-AONT is similar to the LI-AONT OAEP, but they are different in the input of G. In the case of the proposed LP-AONT, it is the exclusive-OR result of message blocks. In the case of the OAEP, it is an ℓ-bit random string that is independent of message blocks.

We mention the relationship between the proposed LP-AONT and the s-out-of-s secret sharing scheme. Let us consider $\mathbf{x}$ and $(y_1, y_2, \ldots, y_s)$ as a secret information and s shares, respectively. The size of $\mathbf{x}$ is ℓs bits and that of y_i is ℓ bits, i.e., the size of a share is precisely $1/s$ of that of the secret information. As described in Section 6, $\mathbf{x}$ cannot be recovered without using all the s shares. Therefore, the proposed LP-AONT can be used instead of conventional s-out-of-s secret sharing schemes. Note that the proposed LP-AONT is not an information-theoretically secure s-out-of-s secret sharing scheme.

6. Security Analysis of the Implementation

We assume that G and H are random oracles. Let q_G and q_H be the numbers of queries to G and H, respectively.

Theorem 1 Suppose that Alice A $(T, q_G + q_H, \epsilon)$-distinguishes F where the u-th message block and the v-th pseudo-message block are lost, $q_G \le 2^{\ell-1}$ and $\ell \ge 2$. If $v = s$, then

$$\epsilon \le 2q_G 2^{-\ell},$$

otherwise

$$\epsilon \le 2q_G \left(\frac{e \ln 2^\ell}{\ln \ln 2^\ell} + 1 \right) 2^{-\ell}, \quad (5)$$

where e is the base of the natural logarithm.

We prove the above theorem. Let AC be the event that Alice's guess is correct. Let r_B denote the input of G used by Bob in this game, i.e., Eq. (4). Let AskR_B be the event that Alice asks about the value of $G(r_B)$. Let FAskR_B and GAskR_B be events that such a query is made in the find stage and in the guess stage, respectively. We have

$$\begin{aligned}
\Pr[\mathsf{AC}] &= \Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] \cdot \Pr[\neg\mathsf{AskR}_B] \\
&\quad + \Pr[\mathsf{AC} \mid \mathsf{AskR}_B] \cdot \Pr[\mathsf{AskR}_B] \\
&\le \Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] + \Pr[\mathsf{AskR}_B] \\
&= \Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] + \Pr[\mathsf{FAskR}_B] \\
&\quad + \Pr[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B] \\
&\quad \cdot \Pr[\neg\mathsf{FAskR}_B] \\
&\le \Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] + \Pr[\mathsf{FAskR}_B] \\
&\quad + \Pr[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B]. \quad (6)
\end{aligned}$$

Since Bob uniformly chooses x_u from $\{0,1\}^\ell$ at random, the distribution of r_B is uniform on $\{0,1\}^\ell$ because of Eq. (4). Since $G(r_B)$ is random if Alice does not ask about $G(r_B)$, we have

$$\Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] = \frac{1}{2}.$$

Moreover, since Bob computes r_B after he received two incomplete message sequences, we have

$$\Pr[\mathsf{FAskR}_B] \le q_{FG} 2^{-\ell}$$

where q_{FG} is the number of queries to G in the find stage. Hence, Eq. (6) is written as

$$\begin{aligned}
\Pr[\mathsf{AC}] &\le \frac{1}{2} + q_{FG} 2^{-\ell} \\
&\quad + \Pr[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B]. \quad (7)
\end{aligned}$$

We evaluate the probability $\Pr[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B]$. Let q_{GG} be the number of queries to G in the guess stage, and let GAskR_B^i denote the event that Alice's i-th query to G in the guess stage is r_B.

$$\begin{aligned}
&\Pr[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B] \\
&= \sum_{i=1}^{q_{GG}} \Pr[\mathsf{GAskR}_B^i \mid \neg\mathsf{FAskR}_B] \\
&= \sum_{i=1}^{q_{GG}} \Pr[\mathsf{GAskR}_B^i \mid \bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \wedge \neg\mathsf{FAskR}_B] \\
&\quad \cdot \Pr[\bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \mid \neg\mathsf{FAskR}_B] \\
&\le \sum_{i=1}^{q_{GG}} \Pr[\mathsf{GAskR}_B^i \mid \bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \wedge \neg\mathsf{FAskR}_B] \\
&\quad (8)
\end{aligned}$$

We first consider the case of $v = s$, i.e., the last pseudo-message block y_s is lost. Since x_u

is uniformly distributed on $\{0,1\}^\ell$, r_B and y_s are uniformly done on it. It follows that the value of $H(y_1\|y_2\|\cdots\|y_{s-1})$ is useless to guess r_B. Eq. (8) is bounded as follows:

$$\sum_{i=1}^{q_{GG}} \Pr\Big[\mathsf{GAskR}_B^i \mid \bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \wedge \neg\mathsf{FAskR}_B\Big] = \sum_{i=1}^{q_{GG}} \frac{1}{2^\ell - (q_{FG} + (i-1))} \le \frac{q_{GG}}{2^\ell - q_G}$$

For r asked by Alice in the find stage, Alice does not ask about $G(r)$ to the oracle G in the guess stage because such a $G(r)$ is probably included in the auxiliary data d. Using the assumption of $q_G \le 2^{\ell-1}$, we have

$$\begin{aligned}\Pr[\mathsf{AC}] &\le \frac{1}{2} + q_{FG}2^{-\ell} + \Pr\big[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B\big] \\ &\le \frac{1}{2} + \frac{q_{FG}}{2^\ell} + \frac{q_{GG}}{2^\ell - q_G} \\ &\le \frac{1}{2} + \frac{q_{FG}}{2^\ell} + \frac{2q_{GG}}{2^\ell} \\ &\le \frac{1}{2} + \frac{2q_G}{2^\ell}.\end{aligned}$$

Next, we consider the case of $v \neq s$. Pseudo-message blocks y_i except for y_v have been fixed in the guess stage. We define a function V: $\{0,1\}^\ell \to \{0,1\}^\ell$ as

$$V(z) = H(y_1\|\cdots\|y_{v-1}\|z\|y_{v+1}\|\cdots\|y_{s-1}) \oplus y_s$$

where $z \in \{0,1\}^\ell$. Since H is the random oracle, V is a random function. Note that Alice does not need to guess the lost y_v correctly to guess r_B because a preimage z such that $V(z) = r_B$ is not necessarily unique. The following lemma is useful for computing the number of such preimages [2),8)].

Lemma 1 Let W: $\{0,1\}^\ell \to \{0,1\}^\ell$ be a random function where $\ell \ge 2$. Then,

$$\mathrm{Ev}\Big[\max_{y\in\{0,1\}^\ell} \big|W^{\{-1\}}(y)\big|\Big] \le \frac{e\ln 2^\ell}{\ln\ln 2^\ell} + 1$$

where the expected value is taken on all functions from $\{0,1\}^\ell$ to $\{0,1\}^\ell$ and $\big|W^{\{-1\}}(y)\big|$ is the number of w such that $y = W(w)$.

Let $\gamma = (e\ln 2^\ell / \ln\ln 2^\ell) + 1$. Since $\gamma > 8$ if $\ell \ge 2$, Eq. (8) is bounded as follows:

$$\begin{aligned}\sum_{i=1}^{q_{GG}} \Pr\Big[\mathsf{GAskR}_B^i \mid \bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \wedge \neg\mathsf{FAskR}_B\Big] &= \sum_{i=1}^{q_{GG}} \frac{\gamma}{2^\ell - (q_{FG} + (i-1))} \\ &\le \frac{\gamma\, q_{GG}}{2^\ell - q_G}.\end{aligned}$$

Therefore, we have

$$\begin{aligned}\Pr[\mathsf{AC}] &\le \frac{1}{2} + q_{FG}2^{-\ell} + \Pr\big[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B\big] \\ &\le \frac{1}{2} + \frac{q_{FG}}{2^\ell} + \frac{\gamma\, q_{GG}}{2^\ell - q_G} \\ &\le \frac{1}{2} + \frac{q_{FG}}{2^\ell} + \frac{2\gamma\, q_{GG}}{2^\ell} \\ &\le \frac{1}{2} + \frac{2q_G\gamma}{2^\ell}.\end{aligned}$$

We have proved Theorem 1.

Theorem 2 Suppose that Alice A $(T, q_G + q_H, \epsilon)$-predicts f where the u-th message block and the v-th pseudo-message block are lost, $q_G \le 2^{\ell-1}$ and $\ell \ge 2$. If $v = s$, then

$$\epsilon \le 2q_G 2^{-\ell},$$

otherwise

$$\epsilon \le 2q_G\left(\frac{e\ln 2^\ell}{\ln\ln 2^\ell} + 1\right)2^{-\ell},$$

where e is the base of the natural logarithm.

The bounds of Theorem 2 are the same as those of Theorem 1 because the proof of Theorem 2 is very similar to that of Theorem 1. We have Eq. (6), i.e.,

$$\Pr[\mathsf{AC}] \le \Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] + \Pr[\mathsf{FAskR}_B] + \Pr\big[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B\big]$$

Since $G(r_B)$ is random if Alice does not ask about $G(r_B)$, we have

$$\Pr[\mathsf{AC} \mid \neg\mathsf{AskR}_B] \le p_f.$$

Moreover, we have

$$\Pr[\mathsf{FAskR}_B] = q_{FG}2^{-\ell}.$$

Since no information is transmitted from the find stage to the guess stage, the bound of $\Pr\big[\mathsf{GAskR}_B \mid \neg\mathsf{FAskR}_B\big]$ is slightly different from that of Theorem 1. Namely, values of $G(r)$ obtained in the find stage cannot be transmitted to the guess stage. It follows that for $v = s$

$$\sum_{i=1}^{q_{GG}} \Pr\left[\mathsf{GAskR}_B^i \mid \bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \wedge \neg\mathsf{FAskR}_B\right]$$
$$= \sum_{i=1}^{q_{GG}} \frac{1}{2^\ell - (i-1)}$$
$$\leq \frac{q_{GG}}{2^\ell - q_{GG}}$$
$$\leq \frac{q_{GG}}{2^\ell - q_G},$$

and for $v \neq s$

$$\sum_{i=1}^{q_{GG}} \Pr\left[\mathsf{GAskR}_B^i \mid \bigwedge_{j=1}^{i-1} \neg\mathsf{GAskR}_B^j \wedge \neg\mathsf{FAskR}_B\right]$$
$$= \sum_{i=1}^{q_{GG}} \frac{\gamma}{2^\ell - (i-1)}$$
$$\leq \frac{\gamma\, q_{GG}}{2^\ell - q_{GG}}$$
$$\leq \frac{\gamma\, q_{GG}}{2^\ell - q_G}.$$

Therefore, the bound of $\Pr[\mathsf{AC}]$ is the same as that of Theorem 1.

Theorem 1 and Theorem 2 show that no adversary (Alice) can do substantially better than by the exhaustive search of the lost pseudo-message block because the advantage of the adversary is $O(\ell/\log \ell)$. It follows that no LP-AONT can achieve substantially better security than the proposed implementation.

Using the example of the filesystem on the HDD described in Section 1, we explain the meaning of the above theorems. Suppose that Bob wants to give his HDD to Alice where all files on his HDD have been recorded with the proposed implementation. Bob first overwrites the first blocks of all files with the random string according to appropriate ways [11),12)]. It follows that Alice cannot recover the original blocks from the overwritten blocks even if she uses expensive HDD probing. After Alice got his HDD, she attempts to know files on his HDD. However, the above theorems imply that her attempt fails, i.e., no attempt gives new information about files on his HDD to her.

7. Concluding Remarks

We have given new security definitions of the LP-AONT under the random oracle model, and proposed the secure implementation of the LP-AONT. Since the assumption of the ambiguity of one message block is not so unrealistic, new definitions offer the sufficient security in applications. We have also proved that the proposed implementation is nearly optimal in terms of the success probability of the adversary. In addition, the proposed implementation is practical since it consists of a pseudo-random generators G and a hash function H.

In this paper, we have assumed that the message and the pseudo message are lost in blocks. For the application of the filesystem, it is a reasonable assumption that the pseudo message is lost in blocks since the user usually overwrites the file in sectors of the HDD or bytes of the file. However, from the theoretical viewpoint, the security for the loss of the pseudo message in bits is an interesting open problem.

Applications of the LP-AONT include the secure filesystem on the HDD. The use of the proposed implementation is probably effective if the size of each of files on the HDD are large, for example, the HDD of a video server. Conversely, if the size of each of files on the HDD is very small, the use of the proposed implementation is possibly ineffective because the time for overwriting all the first blocks of files is nearly equal to the time for overwriting the whole HDD. We will evaluate the practical performance of the proposed implementation.

Acknowledgments This research was partially supported by International Communications Foundation. The authors thank to anonymous reviewers for useful comments.

References

1) Bellare, M. and Rogaway, P.: Optimal Asymmetric Encryption, *Lecture Notes in Computer Science Advanced in Cryptology–EUROCRYPT '94*, Vol.950, pp.92–111 (1994).
2) Boyko, V.: On the Security Properties of OAEP as an All-or-Nothing Transform, *Advances in Cryptology—CRYPTO '99, Lecture Notes in Computer Science*, Vol.1666, pp.503–518 (1999).
3) Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E. and Sahai, A.: Exposure-Resilient Functions and All-or-Nothing Transforms, *Advances in Cryptology—EUROCRYPT 2000, Lecture Notes in Computer Science*, Vol.1807, pp.453–469 (2000).
4) Free Software Foundation, Inc.: Coreutils (2003). http://www.gnu.org/software/coreutils/coreutils.html
5) Johnson, D.B., Matyas, S.M. and Peyravian, M.: Encryption of Long Blocks Using a Short-Block Encryption Procedure (1996). http://grouper.ieee.org/groups/1363/P1363a/contributions/peyrav.pdf

6) Krawczyk, H.: Secret Sharing Made Short, *Advances in Cryptology—CRYPTO '93, Lecture Notes in Computer Science*, Vol.773, pp.136–146 (1993).
7) Kuwakado, H. and Tanaka, H.: All-or-Nothing Transform based on a Linear Code, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E85-A, No.5, pp.1084–1087 (2002).
8) Motwani, R. and Raghavan, P.: *Randomized Algorithms*, Cambridge University Press (1995).
9) Rivest, R.L.: All-or-Nothing Encryption and the Package Transform, *Fast Software Encryption FSE '97, Lecture Notes in Computer Science*, Vol.1267, pp.210–218 (1997).
10) Stinson, D.R.: Something about all or nothing (transform), *Designs, Codes and Cryptography*, Vol.22, pp.133–138 (2001).
11) The Japan Electronics and Information Technology Industries Association (2002). http://it.jeita.or.jp/perinfo/committee/pc/HDDdata/refer.html
12) The Under Secretary of Defense: National industrial security program operating manual supplement (1995). http://www.dtic.mil/whs/directives/corres/pdf/522022msup1_0295/cp8.pdf

(Received November 26, 2004)
(Accepted June 9, 2005)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.304–312.)

Editor's Recommendation

We have a security thereat from an unauthorized access to data on a recycled hard disk drive (HDD). With the all-or-nothing transform (AONT) we can decrypt the data only when we have all the encrypted data. AONT is expected for a use in the variety of applications such as protection of recycle HDD. It has a drawback that the size of the encrypted data increases compared to the plain text, hence, not practical. This paper proposes a solution with more relaxed security definitions of the AONT, and shows the proposed scheme works as secure as the previous ones. The work is novel and presented clearly.

(Chairperson of SIGCSEC Yuko Murayama)

Hidenori Kuwakado received the B.E., M.E. and D.E. degrees from Kobe University in 1990, 1992, and 1999 respectively. He worked for Nippon Telegraph and Telephone Corporation from 1992 to 1996. From 1996 to 2002 he was a Research Associate in the Faculty of Engineering, Kobe University. Since 2002, he has been an Associate Professor in the Faculty of Engineering, Kobe University. His research interests are in cryptography and information security.

Hatsukazu Tanaka was born in Hyogo, Japan, on September 30, 1941. He received the B.E. degree from Kobe University, Kobe, Japan in 1964, the M.E. degree in 1966, and the D.E. degree in 1969, both from Osaka University, Osaka, Japan. He was appointed as a Research Associate in the Faculty of Engineering, University of Osaka Prefecture in 1969. From 1973 through 1987 he was an Associate Professor in the Department of Electrical Engineering, Kobe University. From 1988 through 2005 he was a Professor in the Department of Electrical and Electronics Engineering, Kobe University. Since 2005 he has been a President in Kobe Institute of Computing. From 1980 through 1981 he was a member of the Communication Group of the University of Toronto, Toronto, Ontario, Canada, as a Visiting Scientist. His main work is on the basic theory of Information Engineering such as Information Theory, Coding Theory, Cryptography and Information Security, Image Processing, etc. Dr. Tanaka is a Fellow member of IEEE, a Fellow member of IEICE, and a member of IACR.