An Evaluation Framework for Fastest Oblivious RAM

Seira Hidano, Yuto Nakano and Shinsaku Kiyomoto
KDDI Research, Inc., Saitama, Japan

Keywords: Memory Access Pattern, Oblivious RAM, Average Min-Entropy, Collision Entropy.

Abstract: Oblivious RAM (ORAM) is security provable approach for memory access pattern hiding. However, since ORAM incurs high computational overheads due to repeated shuffles of data blocks in a memory, numerous constructions have been proposed to reduce it. While the computational cost has been improved by these constructions as compared to early ones, it is still expensive from the practical point of view. Specifically, in its application to IoT devices, less computational cost is expected for avoiding high energy consumption. We thus focus on an ORAM construction proposed by Nakano et al. in 2012, which we call the *fastest ORAM*. The computational cost of this construction is much less than any other conventional ORAM constructions. However, the security has not been analyzed sufficiently, due to the lack of practical security definitions. Therefore, we formulate a new security definition for the fastest ORAM on the basis of the average min-entropy, and propose a framework for evaluating the security.

1 INTRODUCTION

IoT devices usually adopt Low Power Wide Area networks (LPWAN) for sending data and receiving control command from an administrator. In many LPWAN technologies including LoRa and Sigfox, end devices and servers share secret keys for authentication, confidentiality, and integrity. Therefore, it is critical to protect these keys from adversaries for security of services. One of the challenges is that an adversary can easily gain access to IoT devices as they are widely deployed in wide area. Once the adversary obtains the device, one can try to extract the key by using reverse engineering. In the case of LoRaWAN, the master secret key called AppKey is used to derive two session keys called AppSKey and NwkSKey just after the device joins the network. If the adversary can detect which data is accessed during generating the session keys, one can also obtain the master key. Oblivious RAM (ORAM) can be used to mitigate the risk of leaking the secret key. By using ORAM, we can hide which data is actually used during the session key generation even if the adversary has full access to RAM. The drawback of using ORAM is its relatively high demanding in computation, hence high energy consumption. Since many IoT devices are designed for long-term use with batteries, ORAM for IoT devices should be lightweight.

The first constructions of ORAM was introduced by Goldreichis et al. in (Goldreich, 1987; Goldreich and Ostrovsky, 2007). These ORAM constructions hide memory access patterns by shuffling locations where data are stored on a memory at a certain interval. However, the shuffling process imposes considerable computational overheads. Although there is a rich literature on ORAM constructions devoted to improving the overheads (Pinkas and Reinman, 2010; Goodrich and Mitzenmacher, 2011; Goodrich et al., 2011; Shi et al., 2011; Stefanov et al., 2011; Kushilevitz et al., 2012; Williams and Sion, 2012; Stefanov and Shi, 2013; Stefanov et al., 2013), their computational overheads are still expensive, which means these constructions are not suitable for practical use. Meanwhile, there is a construction to dramatically improve the overheads, instead of involving information leakage (Nakano et al., 2012). This construction has been proposed by Nakano et al., and has the computation overhead of only two times. We call it the *fastest ORAM* in this paper. The fastest ORAM is expected to be applied to IoT devices, yet there remain concerns on its security. Nakano et al. also provided a new security definition called δ-security for the fastest ORAM. However, this concept was different from that of generic security definitions, such as unpredictability and indisguishability. Analysis with a metric incomparable with well-known ones may engender the degradation of security. We thus revisit the security of the fastest ORAM.

Hidano, S., Nakano, Y. and Kiyomoto, S.
An Evaluation Framework for Fastest Oblivious RAM.
DOI: 10.5220/0006690701140122
In *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)*, pages 114-122
ISBN: 978-989-758-296-7

1.1 Our Contribution

In this paper, we propose a framework for evaluating the *fastest ORAM*. The contributions of this work are the following:

- We formulate a new security definition based on the average min-entropy for ORAM constructions involving information leakage, which is called l-leakage access pattern hiding. This definition is one of the metrics to capture security in the worst case scenario, and comparable with the traditional security definition for ORAM, namely, perfect access pattern hiding. This definition allows us to configure the security level by properly evaluating the amount of the leakage, which means that we can optimize some system specific-parameters used in the fastest ORAM according to security requirements.
- We introduce a practical way to evaluate the amount of information leakage in the fastest ORAM. In order to calculate the amount of the leakage, it is usually required to estimate the probability distribution of memory access patterns. However, the means to the estimation is not known in both cases: theoretical and experimental settings. We thus provide an upper bound of amount of the leakage on the basis of the collision entropy, and give an experimental way to estimate the amount of the leakage from the probability distribution of distance between two access patterns.
- We applied the fastest ORAM to a program of AES, and evaluated the average min-entropy and the amount of information leakage. From the results, we confirm that given a security requirement l, the fastest ORAM can achieve l-leakage access pattern hiding by optimizing its specific parameters.

1.2 Paper Organization

The rest of the paper is organized as follows: Section 2 overviews early studies on Oblivious RAM (ORAM). Section 3 gives some definitions on ORAM and its security, as well as the definitions of some entropic metrics. Section 4 formulates a new security definition based on the average min-entropy and gives an upper bound related to the new definition. Section 5 proposes a framework for evaluating the security of the *fastest ORAM*. Section 7 concludes this paper.

2 RELATED WORK

Oblivious RAM (ORAM) is a security probable approach for memory access pattern hiding. The main ORAM construction consists of two algorithms: an initialization algorithm and an execution algorithm. The initialization algorithm takes data blocks as input and initializes the oblivious structure in a memory containing the data blocks. The execution algorithm compiles each logical access into the virtual access(es). In some ORAM constructions, the initialization algorithm is executed repeatedly at certain interval to ensure the security.

The first construction of ORMA has been proposed by Godreich (Goldreich, 1987), and then extended by Goldreich and Ostrovsky (Goldreich and Ostrovsky, 2007). They introduced the following two ORAM constructions: *Square-Root ORAM*, and *Hierarchical ORAM*. The Square-Root ORAM provides the computational overhead of a square root every access, and the Hierarchical ORAM has polylogarithmic computational complexity. Recently, numerous constructions have been proposed to reduce the overheads (Pinkas and Reinman, 2010; Goodrich and Mitzenmacher, 2011; Goodrich et al., 2011; Shi et al., 2011; Stefanov et al., 2011; Kushilevitz et al., 2012; Williams and Sion, 2012; Stefanov and Shi, 2013; Stefanov et al., 2013). In particular, a series of ORAM constructions, beginning with the construction of Shi et al. (Shi et al., 2011), adopted binary trees as the underlying structure of a memory. Such *Tree-based ORAMs* are more efficient than the hierarchical approach, they still have logarithmic computational complexity (Stefanov et al., 2013). The high computational overhead is caused by repeated shuffling processes.

Meanwhile, Nakano et al. have proposed an ORAM construction where the shuffling process is not executed repeatedly. Instead, a client is required to access a memory on a server twice every access. The construction of Nakano et al. has the computational overhead of only two times. While the overheads are dramatically improved, the security has not been analyzed sufficiently. Nakano et al. focused on the repeated accesses to the same data block and provided a security definition for their construction, which is called δ-security. However, they did not mention the relation between δ-security and the definition for security probable ORAM constructions, which means that we cannot compare the security level between different constructions. We thus revisit the security of the construction proposed by Nakano et al., and propose a framework for evaluating the security.

3 PRELIMINARIES

In this section, we give some definitions on oblivious RAM (ORAM) and its security as well as the definitions of some entropic metrics.

Memory Access Pattern. We consider a client-server model as a target one (e.g. in terms of computer architecture, a CPU is a client, and a data memory is a server storage). A server has a memory $\mathcal{M} = \{m_1, \dots, m_N\}$ consisting of N data blocks for managing a data array of a client. Given a program $\mathcal{R}$, a client accesses the memory $\mathcal{M}$ on the server according to $\mathcal{R}$. Each access is denoted by 3-tuple (op, m_i, v). op is one of the following two operations: read or write. When $\mathsf{op} = \mathsf{read}$, v is denoted by $\perp$, and the client accesses the data block $m_i \in \mathcal{M}$ to get the value stored in it. Otherwise, the client accesses the data block m_i to store the value v in it. Let c_{m_i} be the address of a data block $m_i \in \mathcal{M}$ on the server. When a client has an access (op, m_i, v), a server observes $a = (\mathsf{op}, c_{m_i}, v)$. If a client accesses a memory $\mathcal{M}$ on a server n times, the access pattern is defined by a sequences of n accesses, $\mathbf{a} = (a_1, \dots, a_n) \in \mathcal{A}^n$. For the sake of simplicity, we below consider an access as $a = c_{m_i}$.

Oblivious RAM. An oblivious RAM (ORAM) construction ORAM is a tuple of the following two algorithms:

Init: Given a memory $\mathcal{M}$, this algorithm initializes the structure of the memory while shuffling data blocks in $\mathcal{M}$. After the initialization process is completed, the memory $\mathcal{M}$ is used by the algorithm Exec. In some ORAM constructions, the algorithm Init is executed every T executions of the algorithm Exec for ensuring security.

Exec: Let $\mathbf{a} = (a_1, \dots, a_n) \in \mathcal{A}^n$ be an access pattern that a program $\mathcal{R}$ outputs. In order to hide the access pattern $\mathbf{a}$ from a server, this algorithm compiles $\mathbf{a}$ into another access patten $\mathbf{b} = (b_1, \dots, b_q) \in \mathcal{B}^q$, where $\mathcal{B}$ is the same set as $\mathcal{A}$ and $q \geq n$. We below refer to $\mathbf{a}$ and $\mathbf{b}$ as a private access pattern and an observable access pattern, respectively.

ORAM Security. The security of ORAM constructions is generally defined as follows:

Definition 3.1 (Perfect access pattern hiding (Goodrich et al., 2012)). *Let $\mathbf{a} = (a_1, \dots, a_n) \in \mathcal{A}^n$ and $\mathbf{a}' = (a'_1, \dots, a'_n) \in \mathcal{A}^n$ be private access patterns. Let $\mathbf{b} = (b_1, \dots, b_q) \in \mathcal{B}^q$ and $\mathbf{b}' = (b'_1, \dots, b'_q) \in \mathcal{B}^q$ be observable access patterns of the private access patterns $\mathbf{a}$ and $\mathbf{a}'$, respectively. An ORAM construction $\mathsf{ORAM} = (\mathsf{Init}, \mathsf{Exec})$ is perfect access pattern hiding if the observable access patterns $\mathbf{b}$ and $\mathbf{b}'$ are computationally indistinguishable for anyone but the client.*

While most of existing ORAM constructions were built with the objective of achieving perfect access pattern hiding, the fastest ORAM construction proposed by Nakano et al. (Nakano et al., 2012) was aimed at dramatically improving the computational overheads instead of allowing information leakage. In the case of constructions involving information leakage, the amount of the leakage is required to be properly evaluated for achieving security requirements. Nakano et al. thus gave a different security definition for such ORAM constructions as follows:

Definition 3.2 (δ-length security). *Assume that a data block $m_i \in \mathcal{M}$ is accessed twice at i-th access and $(i+\delta)$-th access by a program $\mathcal{R}$, which is called a δ-distance access of m_i. An ORAM construction* ORAM *is δ-length ε-secure if the probability that an adversary identifies any d-distance access in a private access pattern $\mathbf{a} = (a_1, \dots, a_n)$ is at most ε for every $d \leq \delta$.*

The above security definition was introduced in terms that one of the main challenges of ORAM was to hide repeated accesses to the same data block.

Renyi Entropy. Let X be a random variable on a set $\mathcal{X}$ of possible values. The Renyi entropy (Renyi, 1960) of X for a real number $\alpha \geq 0$ is defined as follows:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_{x \in \mathcal{X}} \Pr[X = x]^\alpha. \quad (1)$$

In particular, we refer to the Renyie entropy for $\alpha = 2$ and the one for $\alpha = \infty$ as collision entropy and min-entropy, respectively. Note that the base of the logarithm is 2 throughout this paper.

Conditional Renyi Entropy. While there are different types of formalization of conditional Renyi entropy, we follow the definition introduced by Fehr et al. (Fehr and Berens, 2014). Let X and Y be random variables on different sets $\mathcal{X}$ and $\mathcal{Y}$ of possible values, respectively. The conditional Renyi entropy of X given Y for a real number $\alpha \geq 0$ is defined as follows:

$$\tilde{H}_\alpha(X|Y) = -\log \left(\mathbb{E}_{y \in \mathcal{Y}} R_\alpha(X|Y=y)^{\frac{\alpha-1}{\alpha}} \right)^{\frac{\alpha}{\alpha-1}}, \quad (2)$$

where

$$R_\alpha(X|Y=y) = \left(\sum_{x \in \mathcal{X}} \Pr[X = x|Y=y]^\alpha \right)^{\frac{1}{\alpha-1}}. \quad (3)$$

In the case of this formalization, the following chain rule holds for all $\alpha \geq 0$:

$$\tilde{H}_\alpha(X|Y) = H_\infty(X,Y) - H_0(Y). \quad (4)$$

The average min-entropy (Dodis et al., 2008) of X given Y is the conditional Renyi entropy for $\alpha = \infty$, which can be written as follows:

$$\tilde{H}_\infty(X|Y) = -\log \mathbb{E}_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} \Pr[X = x|Y = y]. \quad (5)$$

4 NEW SECURITY DEFINITION

We here revisit the security of ORAM constructions involving information leakage. The security level of such ORAM constructions is adjustable by some system-specific parameters. However, if these parameters are configured on the basis of δ-security, the ORAM construction may not fully satisfy with the other security requirements. This is because δ-security captures the resistance to a particular attack (which is not the worst case scenario), and the relation between δ-security and the other security definitions is also not obvious. Thus we formulate a new security definition which can capture the worst-case security.

The average min-entropy, formulated by Dodis et al. (Dodis et al., 2008), is one of practical measures corresponding to the difficulty of guessing or predicting a secret in the worst case scenario. We define l-leakage security based on the average min-entropy for ORAM constructions involving information leakage as follows:

Definition 4.1 (l-leakage access pattern hiding)**.** *Suppose that a memory $\mathcal{M}$ consists of N data blocks. Let $\mathcal{A}^n$ and $\mathcal{B}^q$ be sets of possible values of a private access pattern* $\mathbf{a}$ *and an observable access pattern* $\mathbf{b}$*, respectively. Given random variables* $\mathbf{A}$ *and* $\mathbf{B}$*, on the sets $\mathcal{A}^n$ and $\mathcal{B}^q$, respectively, an ORAM construction* ORAM *is l-leakage access pattern hiding if the average min-entropy $\tilde{H}_\infty(\mathbf{A}|\mathbf{B}) \geq -\log\frac{1}{N^n} - l$.*

When an ORAM construction is perfect access pattern hiding, the average min-entropy is $\tilde{H}_\infty(\mathbf{A}|\mathbf{B}) = -\log\frac{1}{N^n}$. Namely, we can configure the security level in the worst case by properly evaluating the amount of the leakage l.

We also introduce the following lemma regarding the average min-entropy, which allow us to easily calculate the amount of information leakage (see Section 5 for details):

Lemma 4.1. *Given random variables* $\mathbf{A}$ *and* $\mathbf{B}$*, on sets $\mathcal{A}^n$ and $\mathcal{B}^q$ of private access patterns and observable access patterns, respectively, we have the following lower bound of the average min entropy $\tilde{H}_\infty(\mathbf{A}|\mathbf{B})$:*

$$\tilde{H}_\infty(\mathbf{A}|\mathbf{B}) \geq \frac{1}{2}\left(\log\frac{1}{N^q} + H_2(\mathbf{A},\mathbf{B})\right). \quad (6)$$

Proof. Since $\max_{\mathbf{a}\in\mathcal{A}^n} \Pr[\mathbf{A} = \mathbf{a}|\mathbf{B} = \mathbf{b}] \leq \sqrt{\sum_{\mathbf{a}\in\mathcal{A}^n} \Pr[\mathbf{A} = \mathbf{a}|\mathbf{B} = \mathbf{b}]^2}$ is obvious for any $\mathbf{b} \in \mathcal{B}^q$, $2\tilde{H}_\infty(\mathbf{A}|\mathbf{B}) \geq \tilde{H}_2(\mathbf{A}|\mathbf{B})$ holds. From this inequality and the cain rule of Equation (4), we have $\tilde{H}_\infty(\mathbf{A}|\mathbf{B}) \geq \frac{1}{2}(H_2(\mathbf{A},\mathbf{B}) - H_0(\mathbf{B}))$. $H_0(\mathbf{B}) = -\log\frac{1}{|\mathcal{B}^q|} = -\log\frac{1}{N^q}$, and hence we obtain Equation (6). □

From Definition 4.1 and Lemma 4.1, we obtain the following theorem:

Theorem 4.1. *Given a real number $l \geq 0$ as a security parameter, when the following inequality holds, an ORAM construction* ORAM *is l-leakage access pattern hiding:*

$$l \leq -\log\frac{1}{N^n} - \frac{1}{2}\log\frac{1}{N^q} - \frac{1}{2}H_2(\mathbf{A},\mathbf{B}). \quad (7)$$

5 EVALUATION FRAMEWORK

We propose a framework to evaluate security of the *fastest ORAM* (Nakano et al., 2012). In this section, we describe the concrete algorithm of the fastest ORAM, and then introduce a practical way to estimate the amount of information leakage in it from the probability distribution of distance between two memory access patterns.

5.1 Algorithms of Fastest ORAM

In the fastest ORAM, the Init algorithm is executed once before the Exec algorithm. These algorithms are given as follows:

Init: On loading data blocks to a memory $\mathcal{M}$, the corresponding addresses are permuted. Then dummy data blocks are added. A total of N data blocks are loaded to the memory. This process partitions the memory $\mathcal{M}$ into the following two regions: a secure buffer $\mathcal{S}$ and an unsecured memory $\mathcal{U}$, consisting of $N_\mathcal{S}$ data blocks and N_u data blocks, respectively, which means $N = N_\mathcal{S} + N_\mathcal{U}$. The secure buffer $\mathcal{S}$ is implemented within client's trusted boundary, and the unsecured memory $\mathcal{U}$ is kept in a server. In addition, a history table $\mathcal{H}$ of size $N_\mathcal{H}$ is required to store addresses of data blocks that has been moved from the secure buffer $\mathcal{S}$ to the unsecured memory $\mathcal{U}$. The history table is implemented as part of the secure buffer $\mathcal{S}$.

Exec**:** The access to a data block $m \in \mathcal{M}$ is proceeded as follows:

1. If m is in $\mathcal{S}$, two random elements (not m) from $\mathcal{S}$ are replaced with a random element from $\mathcal{U}$

and a random element from $\mathcal{U}$ which had already been accessed before (as recorded in the history table $\mathcal{H}$).

2. If m is not in $\mathcal{S}$ and its address c_m is in $\mathcal{H}$, two random elements from $\mathcal{S}$ are replaced with a random element from $\mathcal{U}$ and m.
3. If m is not in $\mathcal{S}$, and its address c_m is not in $\mathcal{H}$, two random elements from $\mathcal{S}$ are replaces with m and a random element from $\mathcal{U}$ which had already been accessed before (as recorded in the history table $\mathcal{H}$).

If the history table $\mathcal{H}$ gets full, $N_{\mathcal{H}}$ elements are selected at random among $N_{\mathcal{H}}+2$ elements to update $\mathcal{H}$. After the above processes, m is accessed. Algorithm 1 is the concrete algorithm of Exec.

Algorithm 1: The Exec algorithm of the fastest ORAM.

```
Scan $\mathcal{S}$ for $m$.
if $m \in \mathcal{S}$ then
    Replace two random elements (not $m$) in $\mathcal{S}$ with
    two random elements in $\mathcal{U}$, one of them is
    chosen from $\mathcal{H}$ and the other is chosen from $\mathcal{U}$.
else
    Scan $\mathcal{H}$ for $c_m$.
    if $c_m \in \mathcal{H}$ then
        Replace two random elements in $\mathcal{S}$ with a
        random element in $\mathcal{U}$ and $m$.
    else
        Replace two random elements in $\mathcal{S}$ with $m$
        and a random element whose address is
        registered in $\mathcal{H}$.
    end if
end if
Choose $N_{\mathcal{H}}$ elements from $N_{\mathcal{H}}+2$ to update $\mathcal{H}$.
Access $m$.
```

Security. Suppose that a client accesses $m \in \mathcal{M}$ at t-th access. After δ accesses, we have $P_S = \Pr[m \in \mathcal{S}] \geq (\frac{N_S-2}{N_S})^\delta$ and $P_H = \Pr[c_m \in \mathcal{H}] \geq (\frac{N_{\mathcal{H}}}{N_{\mathcal{H}}+2})^\delta$. Nakano et al. introduced the following theorem:

Theorem 5.1. *The fastest ORAM construction is δ-length ε-secure, where*

$$\varepsilon \leq \frac{P_S}{(N_S-2)^2} + \frac{(1-P_S)P_H}{(N_S-2)^2} + \frac{(1-P_S)(1-P_H)}{2(N_S-2)}. \quad (8)$$

Although Theorem 5.1 shows the probability of the adversary detecting the repeated accesses to the same block, the amount of infomation leakage is not clear. In the following, we propose a method to evaluate the leakage.

5.2 How to Evaluate l-Leakage

We here provide a practical way to estimate the average min-entropy $\tilde{H}_\infty(\mathbf{A}|\mathbf{B})$ for the fastest ORAM by using Equation (6) in Lemma 4.1. The number of possible values of observable access patterns, N^q, can be calculated as N^{2n} because the Exec algorithm of the fastest ORAM accesses the unsecured memory $\mathcal{U}$ twice every private access. To calculate the collision entropy $H_2(\mathbf{A},\mathbf{B})$, the marginal probability distribution of $\mathbf{A}$ and $\mathbf{B}$, $p_{\mathbf{A},\mathbf{B}}$, is required, yet it is not known to estimate $p_{\mathbf{A},\mathbf{B}}$ theoretically or experimentally for the fastest ORAM. For the theoretical estimation, extremely complex calculation of probabilities would be involved. The experimental estimation is not also an easy means because a vast number of samples of access patterns must be collected. We thus consider estimating $H_2(\mathbf{A},\mathbf{B})$ without the marginal probability distribution $p_{\mathbf{A},\mathbf{B}}$.

Hidano et al. have proposed a way to estimate the collision entropy $H_2(X)$ of a random variable X on a set $\mathcal{X}$ experimentally by using the probability distribution p_D of distance d between two elements in the set $\mathcal{X}$ (Hidano et al., 2010; Hidano et al., 2012). Specifically, the $H_2(X)$ is given as $-\log p_D(0)$. In addition, the distance distribution p_D can be estimated experimentally with fewer samples of the distance d as compared to the probability distribution of X, denoted as p_X, because the number of possible values of the distance is typically far fewer than that of X.

Suppose that the fastest ORAM is applied to a program $\mathcal{R}$. By using the property introduced by Hidano et al., $H_2(\mathbf{A},\mathbf{B})$ can be estimated by the following procedure:

1. Execute the program $\mathcal{R}$ k times ($k \ll N^{2n}$), and obtain the set of observable access patterns $\mathsf{DB}_{\mathbf{B}} = \{\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_k\}$, in which each pattern corresponds with any of private access patterns $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ that the program $\mathcal{R}$ outputs.
2. Calculate their distance $\hat{d} = (\hat{\mathbf{b}}, \hat{\mathbf{b}}')$ for any two samples of observable access patterns $\hat{\mathbf{b}}, \hat{\mathbf{b}}' \in \mathsf{DB}_{\mathbf{B}}$, and have a set $\mathsf{DB}_D = \{\hat{d}_1, \ldots, \hat{d}_{\frac{k(k-1)}{2}}\}$ of samples of the distance.
3. Estimate the probability distribution p_D of the distance d from the samples DB_D (See Section 5.3 for the details).
4. Calculate the probability $p_D(0)$, and have $H_2(\mathbf{A},\mathbf{B}) = -\log \sum_{\mathbf{a} \in \mathcal{A}^n, \mathbf{b} \in \mathcal{B}^q} \Pr[\mathbf{B} = \mathbf{b}|\mathbf{A} = \mathbf{a}]^2 \Pr[\mathbf{A} = \mathbf{a}]^2 = -\log \frac{1}{N^{2n}} - \log p_D(0)$ (assuming that $\Pr[\mathbf{A} = \mathbf{a}] = \frac{1}{N^n}$ for all $\mathbf{a} \in \mathcal{A}^n$).

From the above results, an lower bound of the average min-entropy $\tilde{H}_\infty(\mathbf{A}|\mathbf{B})$ is given as:

$$\tilde{H}_\infty^{\mathsf{low}}(\mathbf{A}|\mathbf{B}) = -\frac{1}{2}\log p_D(0). \quad (9)$$

Then, an upper bound of amount of information leakage in the fastest ORAM is given as:

$$l^{\mathsf{up}} = -\log\frac{1}{N^n} + \frac{1}{2}\log p_D(0). \quad (10)$$

Given a real number $l \geq 0$ as a security parameter, if $l \leq l^{\mathsf{up}}$, we say from Theorem 4.1 that the fastest ORAM is l-leakage access pattern hiding.

5.3 Estimating Distance Distribution

Let us consider the definition of distance between two memory access patterns and a way to estimate its probability distribution. Edit metrics can be considered to be suitable for the distance. The distance d between two observable access patterns $\mathbf{b}$ and $\mathbf{b}'$ is defined as the smallest number of character insertions and deletions needed for the transformation of $\mathbf{b}$ into $\mathbf{b}'$. Specifically, we utilize the hamming distance as an edit distance, because the length of any access pattern is the same. Given two observable access patterns $\mathbf{b} = (b_1,\ldots,b_i,\ldots,b_{2n})$ and $\mathbf{b}' = (b'_1,\ldots,b'_i,\ldots,b'_{2n})$ of length $2n$, the Hamming distance between $\mathbf{b}$ and $\mathbf{b}'$ can be written as follows:

$$d(\mathbf{b},\mathbf{b}') = \frac{\sum_i I(b_i,b'_i)}{2n}, \quad (11)$$

where I is an indicator function that takes value 1 when $b_i \neq b'_i$ and value 0 otherwise. We give the following two ways to estimate the distance distribution of the Hamming distance experimentally: parametric approach and non-parametric approach.

Parametric Approach. Assuming that the probability that b_i and b'_i are identical can be represented by θ for any index i, the probability distribution of the Hamming distance d can be written as the following binomial distribution $\mathsf{Bi}(\theta,\hat{q})$:

$$p_D(d) = \frac{\hat{q}!}{(\hat{q}d)!(\hat{q}!(1-d))!}\theta^{\hat{q}(1-d)}(1-\theta)^{\hat{q}d}, \quad (12)$$

where $\hat{q}$ is the number of meaningful accesses. If there is some correlation between accesses, $\hat{q}$ may be reduced from $2n$. θ and $\hat{q}$ can be calculated by estimating the values of expectation and variance of the Hamming distance d from its samples DB_D. The expectation and the variance of the binomial distribution $\mathsf{Bi}(\theta,\hat{q})$ can be represented by $1-\theta$ and $\frac{\theta(1-\theta)}{\hat{q}}$, respectively. Once the values of θ and $\hat{q}$ are given, the collision entropy can be easily calculated parametrically as $H_2(\mathbf{B}|\mathbf{A}) = -\log\theta^{\hat{q}}$. However, in some cases, the probability that b_i and b'_i are identical is different for some index i. In this case, the distance distribution p_D cannot be modeled as the Binomial distribution of Equation (12), and there does not exist other suitable parametric approach. Thus, if the probability that b_i and b'_i are identical, the probability cannot be represented by the same parameter θ, the following non-parametric approach should be applied to the estimation.

Non-parametric Approach. Since the probability distribution of the Hamming distance is discrete, it is required to utilize a non-parametric estimator with a discrete kernel function. We thus adopt an estimator with a discrete triangular kernel proposed by Kokonendji et al. (Kokonendji et al., 2007). Given a data set $\mathsf{DB}_D = \{\hat{d}_1,\ldots,\hat{d}_i,\ldots,\hat{d}_{\frac{k(k-1)}{2}}\}$ of samples of the Hamming distance of Equation (11), the probability distribution can be estimated as:

$$p_D(d) = \frac{1}{q}\sum_{i=1}^{k(k-1)/2}\mathcal{K}_{\alpha,\beta,d}(\hat{d}_i), \quad (13)$$

where $q = 2n$ is the length of an observable access pattern. $\mathcal{K}_{\alpha,\beta,d}$ is a discrete triangular kernel of the order α with the arm β, which is defined as follows:

$$\mathcal{K}_{\alpha,\beta,d}(\hat{d}_i) = \begin{cases}\frac{(\beta+1)^\alpha - |q(\hat{d}_i-d)|^\alpha}{P(\alpha,\beta)} & (d \in \mathcal{D}_{\beta,d_i}) \\ 0 & (\text{otherwise})\end{cases}, \quad (14)$$

where $P(\alpha,\beta) = (2\beta+1)(\beta+1)^\alpha - 2\sum_{j=0}^{\beta} j^\alpha$, and $\mathcal{D}_{\beta,d_i} = [\hat{d}_i - \frac{\beta}{q}, \hat{d}_i + \frac{\beta}{q}]$. Hence, $p_D(0)$ can be estimated as:

$$p_D(0) = \frac{1}{q}\sum_{i=0}^{\beta}\frac{w_i[(\beta+1)^\alpha - i^\alpha]}{P(\alpha,\beta)}, \quad (15)$$

where w_i is the number of samples of the distance whose value is $\frac{i}{q}$. See (Kokonendji et al., 2007) for optimizing the parameters α and β. However, non-parametric approach requires more samples as compared parametric one. Thus the parametric estimation based on the Binomial distribution should be utilized in preference to this non-parametric approach if applicable.

6 EXPERIMENTAL EVALUATION

We applied the fastest ORAM to a program of AES, and evaluated the average min-entropy and the amount of information leakage by using the experimental evaluation method provided in Section 5.2.

6.1 Setup

In the experiments, we used an unsecured memory with $N_{\mathcal{U}} = 1424$ data blocks. We also used seven kinds of secure buffers of different sizes $N_{\mathcal{S}} = 8, 16, 32, 64, 128, 256, 512$. The half of data blocks in each secure buffer were used for a history table, i.e. the sizes of the history tables were $N_{\mathcal{H}} = 4, 8, 16, 32, 64, 128, 256$. Under the above conditions, we applied the fastest ORAMs to a program of AES with a 128-bit key, which accesses the memory 3,344 times. Namely, the lengths of the private access pattern and the observable access patterns were $n = 3344$ and $q = 6688$, respectively. The following experiments were conducted for each secure buffer, which means we obtained seven kinds of results.

We, first of all, executed the AES program 1,500 times to have samples of observable access patterns. Then, we calculated the Hamming distance of Equation (11) for any pair of samples of observable access patterns, and obtained 1,124,250 samples of the Hamming distance. For estimation of the probability distribution of the Hamming distance, we adopted both of our parametric approach and non-parametric approach provided as Section 5.3. However, we adopted only the parametric approach for the calculation of the collision entropy (See Section 6.2 for the reason). Finally, we evaluated the average min-entropy and the amount of information leakage by using Equations (9) and (10), respectively.

6.2 Evaluation Results

Distance Distribution. Figure 1 shows an example of measured and estimated distributions of the Hamming distance between observable access patterns. These distributions are for the fastest ORAM with a secure buffer of the size $N_{\mathcal{S}} = 32$. Both of parametrically and non-parametrically estimated distributions well-fitted the measured values. However, we adopted only the parametric approach for evaluating the average min-entropy and the amount of information leakage, as mentioned in Section 6.1. This is because the sample taking value zero or $1/6688$ was not observed. Since the optimized value of the parameter arm for our non-parametric approach was $\beta = 1$, such samples were needed for calculating the collision entropy by using Equation (15). For the fastest ORAMs with secure buffers of different sizes, the same property was observed. Therefore, in addition to the collision entropy, all the results shown in Section 6.2 were calculated from parametrically estimated distributions.

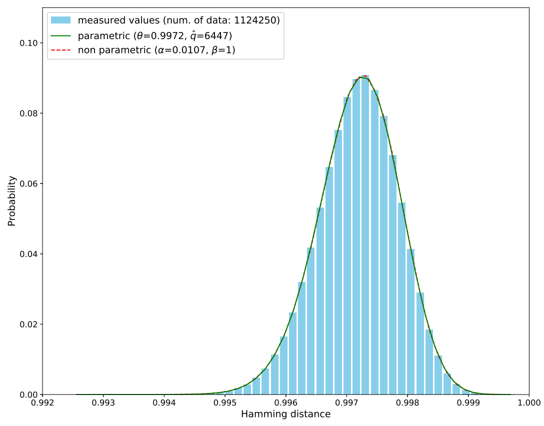

Figure 1: The distribution of the Hamming distance between observable access patterns, for the fastest ORAM with a secure buffer of the size $N_{\mathcal{S}} = 32$. The horizontal axis represents a value of the Hamming distance while the vertical axis indicates the probability that the value is taken. The histogram was made from samples of the Hamming distance. The solid line is the distribution estimated by our parametric approach. Parameters of the binomial distribution $\mathrm{Bi}(\theta, \hat{q})$ were given as $p = 0.9972$ and $\hat{q} = 6447$. The dashed line is the distribution estimated by our non-parametric approach. The optimized values of the order and the arm of the discrete triangular kernel were $\alpha = 0.0107$ and $\beta = 1$.

Average Min-entropy and Amount of Information Leakage. Table 1 shows evaluation results on the average min-entropy and the amount of information leakage for the fastest ORAMs with secured buffers of different sizes. We see from the results that there is the possibility the fastest ORAM involves large amount of information leakage, depending on the sizes of the secure buffer and the history table. On the other hand, the information leakage was improved by increasing these sizes. We thus say that by optimizing these sizes on the basis of a required security level, i.e. the leakage l to be allowed, the fastest ORAM can achieve l-leakage access pattern hiding.

In the case where the fastest ORAM is introduced to IoT devices, as mentioned in Section 1, since such devises do not usually have CPUs with large amount of storage, if a secure buffer of larger size is needed for ensuring security, the size is required to be determined at the design stage of the device. Thus, for such IoT devices that the long-term operation is expected, the CPU storage with an enough margin should be equipped in preparation of coming security risks, which allows us to re-configure the security level of the fastest ORAM.

Table 1: The evaluation results on the average min-entropy and the amount of information leakage for the fastest ORAMs with secure buffers of different sizes $N_S = 8, 16, 32, 64, 128, 256, 512$.

Size of secure buffer	8	16	32	64	128	256	512
Average min-entropy	24564	25032	26391	27319	27981	28224	28246
Amount of information leakage	10467	9998.5	8639.5	7712.1	7049.6	6806.6	6784.6

7 CONCLUSION

We proposed an evaluation framework for the *fastest Oblivious RAM (ORAM)* . While the computational overhead is dramatically improved by avoiding repeated shuffles of data blocks in a memory, the security of the fastest ORAM has not been analyzed sufficiently. We thus formulated a new security definition for ORAM constructions involving information leakage on the basis of the average min-entropy, namely, l-leakage access pattern hiding. We also provided a lower bound using the collision entropy for the average min-entropy. Then, for the fastest ORAM we introduced a practical way to evaluate the amount of information leakage from the probability distribution of distance between memory access patterns. Finally, we applied the fastest ORAM to a program of AES, and evaluated the actual amount of information leakage in the fastest ORAM. As a result, we confirmed that by optimizing the size of a secure buffer used for the fastest ORAM, it can achieve l-leakage access pattern hiding for a required security level l. In the future, we will evaluate the amount of the leakage when the fastest ORAM is applied to other types of programs for validating the usefulness of our proposed framework.

REFERENCES

Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A. (2008). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal of Computing*, 38(1):97–139.

Fehr, S. and Berens, S. (2014). On the conditional rényi entropy. *IEEE Transactions on Information Theory*, 60(11):6801–6810.

Goldreich, O. (1987). Towards a theory of software protection and simulation by oblivious rams. In *Proceedings of the 19th annual ACM symposium on Theory of computing (STOC 1987)*, pages 182–194.

Goldreich, O. and Ostrovsky, R. (2007). Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 19(6–8):241–254.

Goodrich, M. T. and Mitzenmacher, M. (2011). Privacy-preserving access of outsourced data via oblivious ram simulation. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP 2011)*, pages 576–587.

Goodrich, M. T., Mitzenmacher, M., Ohrimenko, O., and Tamassia, R. (2011). Oblivious ram simulation with efficient worstcase access overhead. In *Proceedings of the 3rd ACM Cloud Computing Security Workshop (CCSW 2011)*, pages 95–100.

Goodrich, M. T., Mitzenmacher, M., Ohrimenko, O., and Tamassia, R. (2012). Practical oblivious storage. In *Proceedings of the second ACM conference on Data and Application Security and Privacy (CODASPY 2012)*, pages 13–24.

Hidano, S., Ohki, T., Komatsu, N., and i, K. T. (2010). A metric of identification performance of biometrics based on information content. In *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision (ICARCV 2010)*, pages 1274–1279.

Hidano, S., Ohki, T., and Takahashi, K. (2012). Evaluation of security for biometric guessing attacks in biometric cryptosystem using fuzzy commitment scheme. In *Proceedings of 2012 International Conference of the Biometrics Special Interest Group (BIOSIG 2012)*, pages 1–6.

Kokonendji, C. C., Kiesse, T. S., and Zocchi, S. S. (2007). Discrete triangular distributions and non-parametric estimation for probability mass function. *Journal of Nonparametric Statistics*, 43(3):431–473.

Kushilevitz, E., Lu, S., and Ostrovsky, R. (2012). On the (in)security of hash-based oblivious ram and a new balancing scheme. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 143–156.

Nakano, Y., Cid, C., Kiyomoto, S., and Miyake, Y. (2012). Memory access pattern protection for resource-constrained devices. In *Proceedings of the 11th international conference on Smart Card Research and Advanced Applications (CARDIS 2012)*, pages 188–202.

Pinkas, B. and Reinman, T. (2010). Oblivious ram revisited. In *Proceedings of the 30th Annual Cryptology Conference (CRYPTO 2010)*, pages 512–519.

Renyi, A. (1960). On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561.

Shi, E., Chan, T. H., Stefanov, E., and Li, M. (2011). Oblivious ram with $o((\log n)^3)$ worst-case cost. In *Proceedings the 17th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2011)*, pages 197–214.

Stefanov, E., Dijk, M. V., Shi, E., Fletcher, C., Ren, L., Yu, X., and Devadas, S. (2013). Path oram: an extremely simple oblivious ram protocol. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS 2013)*, pages 299–310.

Stefanov, E. and Shi, E. (2013). Oblivistore: high performance oblivious cloud storage. In *Proceedings of 2013 IEEE Symposium on Security and Privacy (SP 2013)*, pages 253–267.

Stefanov, E., Shi, E., and Song, D. (2011). Towards practical oblivious ram. In *Proceedings of the 19th Network & Distributed System Security Symposium (NDSS 2012)*, pages 1–40.

Williams, P. and Sion, R. (2012). Single round access privacy on outsourced storage. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS 2012)*, pages 293–304.