Some Properties and Privacy Measurement of 0/1-Encoding

Ya-Ting Duan[1,2], Yan-Ping Li[1,2], Lai-Feng Lu[1], and Kai Zhang[1]
(Corresponding author: Yan-Ping Li)
School of Mathematics and Statistics, Shaanxi Normal University[1]
Xi'an 710119, China
Email: lyp@snnu.edu.cn
State key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications[2]
(Received Feb. 5, 2021; Revised and Accepted June 6, 2021; First Online Nov. 13, 2021)

Abstract

The 0/1-encoding is a new encoding method, mainly used to compare the numerical size of two positive integers without giving the specific integers. Its main idea is to reduce the problem of comparing two integers to the problem to find whether two sets have intersections. In this paper, we first analyze and discuss the inherent properties of the 0/1-encoding method by three theorems. Then, if the 0/1-encoding sets are used directly, it will make the adversary get 0/1-encoding results easily, and the adversary has a greater probability of recovering the positive integers being compared. Finally, we theoretically prove the above findings and depict the degree of privacy leakage compared to positive integers when the adversary obtains different 0/1-encoding results.

Keywords: 0/1-Encoding Method; Greater Than (GT) Problem; Millionaire Problem; Privacy Leakage; Set Intersection Problem

1 Introduction

The Millionaire problem is a well-known problem in cryptography [18], which is similar to the greater than (GT) problem [10], namely, to determine which of two integers is larger without leaking any information about integer values being compared. Since Yao introduced this problem and gave a solution, many scheme [1,6,8,10,15] have been put forward to solve this problem using different methods. In particular, Lin *et al.* [10] proposed a special encoding method, namely 0/1-encoding, which transformed the greater than problem into the set intersection problem. Because of the simplicity of the 0/1-encoding method, more and more researchers pay attention to it and it is widely applied in many fields.

References [2,3,5,7,11,12,14,16,19,20] introduced the 0/1-encoding into their schemes to solve problems in different fields. The schemes [7,11,14] introduced the 0/1-encoding into the time-limited signature. Firstly, the key expiration time T_1 was embedded into the user's private key, and then T_1 and the signature time T_2 were encoded according to the 1-encoding and 0-encoding respectively. If there was a common element in these two encoding sets, then the signature was valid and can be verified. The specific problems in [2,3,5,12,16,19,20] were also solved by the 0/1-encoding method. For example, Shishido *et al.* [16] proposed a test scheme to judge whether an integer d belongs to a range of $[a,b]$. This scheme first encoded the left and right endpoints a and b with 1-encoding [4] and 0-encoding, respectively, and then detected whether the prefix string set [16] of integer d has common elements with the encoding sets of two endpoints. If they had a common element, then d can be judged to be out of the range $[a,b]$.

In this paper, we find that if the 0/1-encoding sets are directly used without any processing, it may lead to the privacy leakage of the integer values being compared, which obviously contradicts the original intention of the 0/1-encoding method obviously. For example, some attribute-based encryption schemes [13,17] used the 0/1-encoding method to compare whether the numerical attributes (height, weight, or age) of the data owners and the data users are matched or not. These numerical attributes often are the personal privacy and the user does not want to expose their privacy, so they make indirect comparison using the 0/1-encoding method. If all 0/1-encoding results are stored in the cloud server, and any entity who obtains the encoding results can recover the exact values of these numerical attributes according to the original encoding sets, which is contrary to the original designed intention of the 0/1-encoding method. Therefore, this paper analyzes and discusses the properties and defects of the 0/1-encoding method, and points out the precautions for its use. Here, we summarize the contributions of this paper as follows:

- We analyze and discuss the inherent properties of the

0/1-encoding method by three theorems, and further extend the 0/1-encoding method to indirectly judge the relations (" $>$ ", " $\leq$ " and " $=$ ") of two positive integers.

- We find that if the 0/1-encoding is used directly, it will make the adversary get 0/1-encoding results easily and the adversary has a greater probability to recover the positive integers being compared, and prove the findings by a theorem.
- We depict the degree of privacy leakage of the compared positive integers when the adversary obtains different 0/1-encoding results by two theorems and some examples.

The remainder of this paper is organized as follows. Section 2 reviews some basic knowledge which might be used. Section 3 gives three theorems to introduce properties of the 0/1-encoding method. Section 4 describes the correlation between the exposure of the 0/1 encoding sets and the privacy leakage of the compared integers via three theorems and specific examples. Section 5 gives some suggestions for 0/1-encoding to protect the privacy of integers being compared.

2 Preliminaries

Definition 1. *(see [10]) Let $x = x_n x_{n-1}...x_1 \in GF_2^n$ be a n-bit binary string, then the 0/1-encoding set of x are defined as sets S_x^0 and S_x^1, which are shown below:*

$$\begin{aligned} S_x^0 &= \{x_n x_{n-1}...x_{i+1}1 | x_i = 0, 1 \leq i \leq n\}, \\ S_x^1 &= \{x_n x_{n-1}...x_i | x_i = 1, 1 \leq i \leq n\}. \end{aligned}$$

Both S_x^0 and S_x^1 have at most n elements.

To compare which of two integers x and y is greater by the 0/1-encoding method, we first encode them as binary strings of the same length (and if the length is not equal, the top should be added with 0 to the same length), then determine whether there is a non-empty intersection between the 1-encoding set S_x^1 of x and the 0-encoding set S_y^0 of y (or between the 0-encoding set S_x^0 of x and the 1-encoding set S_y^1 of y). If $S_x^1 \bigcap S_y^0 \neq \varnothing$, we can determine that $x > y$. Otherwise, $x \leq y$. We illustrate it with the following example.

Example 1. *Let $x = 45, y = 40$ and their binary strings be 101101 and 101000 respectively. Then the 0/1-encoding sets of x and y are*

$$\begin{aligned} S_x^0 &= \{10111, 11\}, S_x^1 = \{101101, 1011, 101, 1\}. \\ S_y^0 &= \{101001, 10101, 1011, 11\}, S_x^1 = \{101, 1\}. \end{aligned}$$

Since $S_x^1 \bigcap S_y^0 = \{1011\} \neq \varnothing$, we can infer that $x > y$. Of course, we also can get $y \leq x$ by $S_x^0 \bigcap S_y^1 = \varnothing$. By the 0/1-encoding, we can get a conclusion that x is greater than y.

3 Main Properties

In this section, three theorems are given to illustrate how to determine the relations (" $>$ ", " $\leq$ " and " $=$ ") of two positive integers by the 0/1-encoding method, which indicates that the 0/1-encoding can indirectly judge the above relations (" $>$ ", " $\leq$ " and " $=$ ") of two integers. First, we give some notations and their meanings in Table 1 that might be used later.

Theorem 1. Let x and y are any two positive integers. Then $x > y$ if and only if S_x^1 and S_y^0 have only a common element.

Proof. Let $x = x_n x_{n-1}...x_1 \in GF_2^n, y = y_n y_{n-1}...y_1 \in GF_2^n$.

Prove sufficiency. Suppose there is an element $t = t_n t_{n-1}...t_i \in S_x^1 \bigcap S_y^0$ with $t_i = 1, i \in [1, n]$. Since $t \in S_x^1$, there must exsit $x_n x_{n-1}...x_i = t_n t_{n-1}...t_i$. And due to $t \in S_y^0$, we can get that $y_n y_{n-1}...y_i = t_n t_{n-1}...\bar{t_i}$, namely

$$\begin{cases} x_j = y_j, & j \in [i+1, n] \\ x_i = 0, y_i = 0, & i = j \end{cases}.$$

Therefore, we can infer that $x > y$.

Prove necessity. We first prove that S_x^1 and S_y^0 have a common element, that is, prove the existence of the common element. If $x > y$, then there must be an integer $i \in [1, n]$, such that

$$\begin{cases} x_j = y_j, & j \in [i+1, n] \\ x_i = 0, y_i = 0, & i = j \end{cases} \quad (1)$$

where i must be the first value of i that satisfies the above condition (1). From the above conditions (1), we can know $x_n x_{n-1}...x_{i+1}x_i = y_n y_{n-1}...y_{i+1}1$. According to the 1-encoding set S_x^1 of x and the 0-encoding set S_y^0 of y, it is easy to get $x_n x_{n-1}...x_{i+1}x_i \in S_x^1$ and $y_n y_{n-1}...y_{i+1}1 \in S_y^0$. Hence, $x_n x_{n-1}...x_{i+1}x_i = y_n y_{n-1}...y_{i+1}1 \in S_x^1 \bigcap S_y^0$, namely, S_x^1 and S_y^0 have a common element.

Next, we will prove that S_x^1 and S_y^0 have only a common element, i.e., the uniqueness of the common element. Assume there is another common element $t \in S_x^1 \bigcap S_y^0$, then t must be represented as follows:

$$t = x_n...x_{i+1}x_i x_{i-1}...x_{j+1}x_j \text{ (or } y_n...y_{i+1}1y_{i-1}...y_{j+1}1).$$

Obviously, the corresponding values of the above two binary strings are equal, namely, $x_k = y_k, k \in [j, n]$, where $y_i = y_j = 1$. Since $t \in S_y^0$, there be $y_n y_{n-1}...y_j = y_n y_{n-1}...y_{i+1}1y_{i-1}...y_{j+1}0$, and then $y_n y_{n-1}...y_{i+1}y_i = y_n y_{n-1}...y_{i+1}1$. And due to $y_n y_{n-1}...y_{i+1}1 \in S_y^0$, we can infer that $y_n y_{n-1}...y_{i+1}y_i = y_n y_{n-1}...y_{i+1}0$. This is a contradiction obviously. Therefore, S_x^1 and S_y^0 have only a common element.

Table 1: Notations

notations	meanings	notations	meanings
$[1,n]$	all integers between 1 and n	GF_2	Galois field
$\bar{t}$	complement of t in GF_2	$\|S\|$	the cardinality of the set S
C_n^m	the combinations number $\frac{n!}{m!(n-m)!}$	$\|\|$	cascading symbol of binary strings

□

Remark 1. From the uniqueness proof process of Theorem 1, we know that their encoding sets S_x^1 and S_y^0 (or S_x^0 and S_y^1) do not have two or more common elements for any two positive integers x and y. Therefore, the following $S_x^1 \bigcap S_y^0 \neq \varnothing$ (or $S_x^0 \bigcap S_y^1 \neq \varnothing$) indicates that S_x^1 and S_y^0 (or S_x^0 and S_y^1) have only a common element.

Theorem 2. Let x and y be any two positive integers. Then $x \leq y \Leftrightarrow S_x^1 \bigcap S_y^0 = \varnothing$.

Proof. We use apagoge to prove it. Given $S_x^1 \bigcap S_y^0 = \varnothing$, if $x > y$, then S_x^1 and S_y^0 have only a common element. This is contradictory with $S_x^1 \bigcap S_y^0 = \varnothing$, so $x \leq y$. On the contrary, let $x \leq y$, if $S_x^1 \bigcap S_y^0 \neq \varnothing$, namely, S_x^1 and S_y^0 have only a common element, then $x > y$ according to Theorem 1. It contradicts with $x \leq y$, so $S_x^1 \bigcap S_y^0 = \varnothing$. □

Theorem 3. Let x and y be any two positive integers. The following three conditions are equivalent*:*

1) $x = y$*;*

2) $S_x^1 \bigcap S_y^0 = \varnothing$ and $S_x^0 \bigcap S_y^1 = \varnothing$*;*

3) $S_x^1 \bigcap S_y^0 = \varnothing$ and $S_x^1 \bigcap S_{y-1}^0 \neq \varnothing$.

Proof.

(1) ⇒ (2) $x = y \Leftrightarrow x \leq y$ **and** $y \leq x$**.** According to Theorem 2, we can get that

$$x \leq y \Leftrightarrow S_x^1 \bigcap S_y^0 = \varnothing, y \leq x \Leftrightarrow S_y^1 \bigcap S_x^0 = \varnothing.$$

Therefore, $S_x^1 \bigcap S_y^0 = \varnothing$ and $S_x^0 \bigcap S_y^1 = \varnothing$.

(2) ⇒ (3)**.** We just need to prove $S_x^1 \bigcap S_{y-1}^0 \neq \varnothing$. From Theorem 2, we know that

$$S_x^0 \bigcap S_y^1 = \varnothing \Rightarrow y \leq x.$$

Since $y - 1 < y$, we can get $y - 1 < x$. It is known by Theorem 1 that

$$x > y - 1 \Leftrightarrow S_x^1 \text{ and } S_{y-1}^0 \text{ have only a common element.}$$

Therefore, $S_x^1 \bigcap S_{y-1}^0 \neq \varnothing$.

(3) ⇒ (1)**.** Suppose that $S_x^1 \bigcap S_y^0 = \varnothing$ and $S_x^1 \bigcap S_{y-1}^0 \neq \varnothing$. According to Theorem 1 and Theorem 2

$$S_x^1 \bigcap S_y^0 = \varnothing \Rightarrow x \leq y,\ S_x^1 \bigcap S_{y-1}^0 \neq \varnothing \Rightarrow x > y - 1.$$

That is, $y-1 < x \leq y$. And due to x be a positive integer, then $x = y$. □

4 Correlation between the 0/1-Encoding Sets and the Privacy of Compared Integer

If the 0/1-encoding sets are not processed and directly submitted to the third party or both parties exchange the 0/1-encoding sets to find whether there is a non-empty intersection, which may lead to the privacy leakage of the integer values being compared. This contradicts with the original designed intention of this encoding method. The following will illustrate this fact via three theorems and concrete examples.

Theorem 4. If the length(n-bit) of binary string of a positive integer x is known, and

$$|S_x^1| = n_1 \ (\text{or } |S_x^0| = n_2),$$

then we have the probability of $\frac{1}{C_n^{n_1}}$ (or $\frac{1}{C_n^{n-n_2}}$) to recover x, where $n = n_1 + n_2$.

Proof. Let $x = t_n t_{n-1} ... t_1 \in GF_2^n$. If $|S_x^1| = n_1$, then there are n_1 values of t_i ($i \in [1,n]$) which are equal to 1. And there are totally $C_n^{n_1}$ positions that locate $t_i = 1$, thus there is the probability of $\frac{1}{C_n^{n_1}}$ to get x. Similarly, if we know $|S_x^0| = n_2$, we can also prove that there is the probability of $\frac{1}{C_n^{n-n_2}}$ to recover x value. □

Example 2. *Suppose that an adversary knows the number of elements in a 0/1-encoding set, namely,* $|S_x^1| = 3$ *(or* $|S_x^0| = 3$*), then the adversary has the probability of* $\frac{1}{C_6^3} = \frac{1}{20}$ *to get* x*. See Table 2.*

Obviously, there are 20 possible x values. However, since x is usually associated with a specific attribute (height, weight, or age), the adversary can further determine x based on other information. For example, x represents the age of the breast cancer patient in the example, then the range of x can be determined to [40,56], and the possible values of x are 41, 42, 44, 49, 50, 52, 56. Therefore, the adversary has the probability of $\frac{1}{7}$ *to get x.*

Next, we first discuss the rule of binary strings of the 0/1-encoding sets for a certain integer x by Theorem 5, and then summarize how to recover the integer x from the 0/1-encoding sets and prove the fact with Theorem 6.

Theorem 5. Assume that $x = x_k^{(i)} = t_k t_{k-1} ... t_1 \in GF_2^k$, then the 0/1-encoding set of $x_k^{(i)}$ can be represented by

Table 2: All the possible values of x

x	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
binary	111000	110100	110010	110001	101100	101010	101001	100110	100101	100011
decimal	56	52	50	49	44	42	41	38	37	35
x	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
binary	011100	011010	011001	010110	010101	010011	001110	001101	001011	000111
decimal	28	26	25	22	21	19	14	13	11	7

the 0/1 encoding set of $x_{k-1}^{(j)}$ as follows.

$$S^0_{x_k^{(i)}} = \begin{cases} \left\{1,0||c|c \in S^0_{x_{k-1}^{(j)}}\right\}, & t_k = 0 \\ \left\{1||c|c \in S^0_{x_{k-1}^{(j)}}\right\}, & t_k = 1 \end{cases},$$

$$S^1_{x_k^{(i)}} = \begin{cases} \left\{0||c|c \in S^1_{x_{k-1}^{(j)}}\right\}, & t_k = 0 \\ \left\{1,1||c|c \in S^1_{x_{k-1}^{(j)}}\right\}, & t_k = 1 \end{cases},$$

where $j = \begin{cases} i, & 1 \le i \le 2^{k-1} \\ i - 2^{k-1}, & 2^{k-1} < i \le 2^k \end{cases}$.

Proof. We will use the mathematical induction to prove it.

Let $n = 1, x = x_1^{(i)} = t_1 \in GF_2$, then there are two cases:

$$x_1^{(1)} = 0, S^0_{x_1^{(1)}} = \{1\}, S^1_{x_1^{(1)}} = \varnothing;$$
$$x_1^{(2)} = 1, S^0_{x_1^{(2)}} = \varnothing, S^1_{x_1^{(2)}} = \{1\}.$$

Let $n = 2, x = x_2^{(i)}) = t_2t_1 \in GF_2^2$, then there are four cases:

$$x_2^{(1)} = 00, S^0_{x_2^{(1)}} = \{1,01\}, S^1_{x_2^{(1)}} = \varnothing;$$
$$x_2^{(2)} = 01, S^0_{x_2^{(2)}} = \{1\}, S^1_{x_2^{(2)}} = \{01\};$$
$$x_2^{(3)} = 10, S^0_{x_2^{(3)}} = \{11\}, S^1_{x_2^{(3)}} = \{1\};$$
$$x_2^{(4)} = 11, S^0_{x_2^{(4)}} = \varnothing, S^1_{x_2^{(4)}} = \{1,11\}.$$

Let $n = 3, x = x_3^{(i)}) = t_3t_2t_1 \in GF_2^3$, then there are eight cases:

$$x_3^{(1)} = 000, S^0_{x_3^{(1)}} = \{1,01,001\}, S^1_{x_3^{(1)}} = \varnothing;$$
$$x_3^{(2)} = 001, S^0_{x_3^{(2)}} = \{1,01\}, S^1_{x_3^{(2)}} = \{001\};$$
$$x_3^{(3)} = 010, S^0_{x_3^{(3)}} = \{011,1\}, S^1_{x_3^{(3)}} = \{01\};$$
$$x_3^{(4)} = 011, S^0_{x_3^{(4)}} = \{1\}, S^1_{x_3^{(4)}} = \{01,011\};$$
$$x_3^{(5)} = 100, S^0_{x_3^{(5)}} = \{101,11\}, S^1_{x_3^{(5)}} = \{1\};$$
$$x_3^{(6)} = 101, S^0_{x_3^{(6)}} = \{11\}, S^1_{x_3^{(6)}} = \{1,101\};$$
$$x_3^{(7)} = 110, S^0_{x_3^{(7)}} = \{111\}, S^1_{x_3^{(7)}} = \{1,11\};$$
$$x_3^{(8)} = 111, S^0_{x_3^{(8)}} = \varnothing, S^1_{x_3^{(8)}} = \{1,11,111\}.$$

From above, we can see that

$$S^0_{x_2^{(i)}} = \begin{cases} \left\{1,0||c|c \in S^0_{x_1^{(j)}}\right\}, & t_2 = 0 \\ \left\{1||c|c \in S^0_{x_1^{(j)}}\right\}, & t_2 = 1 \end{cases},$$

$$S^1_{x_2^{(i)}} = \begin{cases} \left\{0||c|c \in S^1_{x_1^{(j)}}\right\}, & t_2 = 0 \\ \left\{1,1||c|c \in S^1_{x_1^{(j)}}\right\}, & t_2 = 1 \end{cases},$$

where $j = \begin{cases} i, & 1 \le i \le 2 \\ i - 2, & 2 < i \le 2^2 \end{cases}$.

$$S^0_{x_3^{(i)}} = \begin{cases} \left\{1,0||c|c \in S^0_{x_2^{(j)}}\right\}, & t_3 = 0 \\ \left\{1||c|c \in S^0_{x_2^{(j)}}\right\}, & t_3 = 1 \end{cases},$$

$$S^1_{x_3^{(i)}} = \begin{cases} \left\{0||c|c \in S^1_{x_2^{(j)}}\right\}, & t_3 = 0 \\ \left\{1,1||c|c \in S^1_{x_2^{(j)}}\right\}, & t_3 = 1 \end{cases},$$

where $j = \begin{cases} i, & 1 \le i \le 2^2 \\ i - 2^2, & 2^2 < i \le 2^3 \end{cases}$. So when $n = 2$ and $n = 3$, the conclusion holds.

Now suppose $n = k - 1$, the conclusion also holds, namely

$$S^0_{x_{k-1}^{(i)}} = \begin{cases} \left\{1,0||c|c \in S^0_{x_{k-2}^{(j)}}\right\}, & t_{k-1} = 0 \\ \left\{1||c|c \in S^0_{x_{k-2}^{(j)}}\right\}, & t_{k-1} = 1 \end{cases},$$

$$S^1_{x_{k-1}^{(i)}} = \begin{cases} \left\{0||c|c \in S^1_{x_{k-2}^{(j)}}\right\}, & t_{k-1} = 0 \\ \left\{1,1||c|c \in S^1_{x_{k-2}^{(j)}}\right\}, & t_{k-1} = 1 \end{cases},$$

where $j = \begin{cases} i, & 1 \le i \le 2^{k-2} \\ i - 2^{k-2}, & 2^{k-2} < i \le 2^{k-1} \end{cases}$. Then when $n = k$, namely, $x = x_k^{(i)} = t_kt_{k-1}...t_1 \in GF_2^k$, there are two cases:

1) $t_k = 0$.

a. Let $t_{k-1}=0$, that is, $x_k^{(i)}=00t_{k-2}...t_1$, we can get $S^0_{x_k^{(i)}}=\left\{1,01,0||0||c|c\in S^0_{x_{k-2}^{(j)}}\right\}$ and $S^1_{x_k^{(i)}}=\left\{0||0||c|c\in S^1_{x_{k-2}^{(j)}}\right\}$, where $1\le i=j\le 2^{k-2}$.

b. Let $t_{k-1}=1$, namely, $x_k^{(i)}=01t_{k-2}...t_1$, we can get $S^0_{x_k^{(i)}}=\left\{1,0||0||c|c\in S^0_{x_{k-2}^{(j)}}\right\}$ and $S^1_{x_k^{(i)}}=\left\{01,0||0||c|c\in S^1_{x_{k-2}^{(j)}}\right\}$, where $j=i-2^{k-2}, 2^{k-2}\le i\le 2^{k-1}$.

So when $t_k=0$, we can get $S^0_{x_k^{(i)}}=\left\{1,0||c|c\in S^0_{x_{k-1}^{(j)}}\right\}, S^1_{x_k^{(i)}}=\left\{0||c|c\in S^1_{x_{k-1}^{(j)}}\right\}$, where $1\le i=j\le 2^{k-1}$.

2) $t_k=1$.

a. Let $t_{k-1}=0$, that is to say, $x_k^{(i)}=10t_{k-2}...t_1$, we can get $S^0_{x_k^{(i)}}=\left\{11,1||0||c|c\in S^0_{x_{k-2}^{(j)}}\right\}$ and $S^1_{x_k^{(i)}}=\left\{1,1||0||c|c\in S^1_{x_{k-2}^{(j)}}\right\}$, where $j=i-2^{k-1}, 2^{k-1}<i\le 3\times 2^{k-2}$.

b. Let $t_{k-1}=1$, that means $x_k^{(i)}=11t_{k-2}...t_1$, we can get $S^0_{x_k^{(i)}}=\left\{1||1||c|c\in S^0_{x_{k-2}^{(j)}}\right\}$ and $S^1_{x_k^{(i)}}=\left\{1,11,1||1||c|c\in S^1_{x_{k-2}^{(j)}}\right\}$, where $j=i-3\times 2^{k-2}, 3\times 2^{k-2}<i\le 2^k$.

So when $t_k=1$, $S^0_{x_k^{(i)}}=\left\{1||c|c\in S^0_{x_{k-1}^{(j)}}\right\}$ and $S^1_{x_k^{(i)}}=\left\{1,1||c|c\in S^1_{x_{k-1}^{(j)}}\right\}$ hold, where $j=i-2^{k-1}, 2^{k-1}<i\le 2^k$.

Therefore, when $n=k$, we can get

$$S^0_{x_k^{(i)}}=\begin{cases}\left\{1,0||c|c\in S^0_{x_{k-1}^{(j)}}\right\}, & t_k=0\\ \left\{1||c|c\in S^0_{x_{k-1}^{(j)}}\right\}, & t_k=1\end{cases},$$

$$S^1_{x_k^{(i)}}=\begin{cases}\left\{0||c|c\in S^1_{x_{k-1}^{(j)}}\right\}, & t_k=0\\ \left\{1,1||c|c\in S^1_{x_{k-1}^{(j)}}\right\}, & t_k=1\end{cases},$$

where $j=\begin{cases}i, & 1\le i\le 2^{k-1}\\ i-2^{k-1}, & 2^{k-1}<i\le 2^k\end{cases}$.

To sum up, the conclusion of Theorem 5 is true. □

Remark 2. When $S^1_{x_k^{(i)}}=\varnothing$ (or $S^0_{x_k^{(i)}}=\varnothing$), $\left\{1||c|c\in S^1_{x_k^{(i)}} \text{ or } S^0_{x_k^{(i)}}\right\}=\varnothing$.

Theorem 6. Let x be an any positive integer. Given the 0/1-encoding sets S_x^0 and S_x^1, the value of x must be recoverable.

Proof. Given a positive integer x, it must be equal to some $x_n^{(i)}$. Thus we can recover x according to Theorem 5 theoretically. The proof process of Theorem 5 shows that we can recover the value of $x_n^{(i)}$ when given the corresponding 0/1-encoding sets $S^0_{x_n^{(i)}}$ and $S^1_{x_n^{(i)}}$, where $n=1, 2, 3$. When $n=4$, given the 0/1-encoding sets $S^0_{x_4^{(i)}}$ and $S^1_{x_4^{(i)}}$ of $x_4^{(i)}$, they must be represented by the 0/1-encoding sets $S^0_{x_3^{(j)}}$ and $S^1_{x_3^{(j)}}$ of $x_3^{(j)}$, where i and j satisfy Theorem 5. Specifically, we just need to figure out the corresponding $x_3^{(j)}$. Assume $x_3^{(j)}=t_3t_2t_1$, if the 0/1-encoding sets of $x_4^{(i)}$ and $x_3^{(j)}$ satisfy the first case of Theorem 5, then $x_4^{(i)}=0t_3t_2t_1$. Otherwise, $x_4^{(i)}=1t_3t_2t_1$. In this way, we can certainly recover the value of $x_n^{(i)}$ when $n\le k-1$.

When $n=k$, given the 0/1-encoding sets $S^0_{x_k^{(i)}}$ and $S^1_{x_k^{(i)}}$, we can definitely find the corresponding $x_{k-1}^{(j)}$, where the 0-encoding sets $S^0_{x_{k-1}^{(j)}}$ and $S^0_{x_k^{(i)}}$ (and the 1-encoding sets $S^1_{x_{k-1}^{(j)}}$ and $S^1_{x_k^{(i)}}$) satisfy Theorem 5. Suppose $x_{k-1}^{(j)}=t_{k-1}...t_2t_1$, we can get $x_k^{(i)}=0t_{k-1}...t_2t_1$ or $x_k^{(i)}=1t_{k-1}...t_2t_1$ according to Theorem 5.

To sum up, if S_x^0 and S_x^1 of x are given, we can definitely recover the value of x theoretically. □

Remark 3. In the above proof, we can recover the value $x=x_n^{(i)}$ by finding the corresponding $x_{n-1}^{(j)}$, and recover the value of $x_{n-1}^{(j)}$ by the corresponding $x_{n-2}^{(k)}$, and so on. We can finally get the value of x using the recursion method. However, it is a little bit tedious, Theorem 5 just claims that x can be recovered theoretically.

In the following, we will give a specific method to recover the value of x.

Proof. Assume $x=t_nt_{n-1}...t_1\in GF_2^n$ and t be the longest binary string of the 0/1-encoding sets of x, then there must be two cases:

1) If $t_1=1$, then $t=t_nt_{n-1}...t_1\in S_x^1$;

2) If $t_1=0$, then $t=t_nt_{n-1}...\bar{t_1}\in S_x^0$;

Given the 0/1-encoding sets S_x^0 and S_x^1 of x, it is easy to find the longest binary string $t=y_ny_{n-1}...y_1$ of two sets, and then we can determine the relationship x and t according to the set which t belongs to. Finally, we can recover x, i.e.,

$$\begin{cases}x=t=y_ny_{n-1}...y_1, & t\in S_x^1\\ x=y_ny_{n-1}...\bar{y_1}, & t\in S_x^0\end{cases}.$$

From above, we can see that the value of x depends on the set which the longest binary string t belongs to, and

it mainly depends on the value of t_1 ($t_1 \in GF_2$). Given an any positive integer x, it must be

$$t_1 = \begin{cases} 0, & x = 2n \\ 1, & x = 2n-1 \end{cases}, n \in N^*.$$

Since $Pr(x = 2n) = Pr(x = 2n-1) = \frac{1}{2}$, so $Pr(t_1 = 1) = Pr(t_1 = 0) = \frac{1}{2}$, namely $Pr(t \in S_x^1) = Pr(t \in S_x^0) = \frac{1}{2}$. Therefore, if an adversary only has a set S_x^0 or S_x^1 of x, it also has a probability of $\frac{1}{2}$ to recover the value of x. □

Example 3. *The adversary captures two encoding sets of x (and it maybe do nor know which set of S_1 and S_2 is the 0-encoding set or 1-encoding set),*

$$\begin{aligned} S_1 &= \{10110101, 1011011, 10111, 11\}, \\ S_2 &= \{101101, 1011, 101, 1\}. \end{aligned}$$

The adversary can first find the longest binary string $t = 10110101$ and $t \in S_1$. By observing the relationship of these elements in S_1, it can be found that S_1 is 0-encoding set (because short codes must be prefixed to long codes in the 1-encoding set). Hence, $t \in S_1 = S_x^0$ and $x = 10110100 = 180$. To sum up, even if the adversary captures part of the 0/1-encoding sets, there still exists a certain probability that the adversary will recover the integer values being compared.

5 Conclusions

In this paper, three theorems are given to illustrate how to determine the relations (" $>$ ", " $\leq$ " and " $=$ ") of two positive integers by the 0/1-encoding method. Then, another three theorems and related examples show that if the 0/1-encoding results are not blindly preprocessed, it is easy to leak the integer values being compared, which obviously contradicts the original designed intention of the encoding method. Since the privacy of integer values in many fields, when using the 0/1-encoding method for numerical comparison, the 0/1-encoding sets should be properly encrypted or blinded, or the intersection of two 0/1-encoding sets should be calculated confidentially [9] to avoid privacy leakage.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61802243), the Key R&D program in industry field of Shaanxi Province (2019GY-013) and the basic science research program of Shaanxi Province (2019JQ273,2020JM288), and Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2020-1-03). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] I. F. Blake and V. Kolesnikov, "Strong conditional oblivious transfer and computing on intervals," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 515–529, 2004.

[2] H. Dai, T. Wei, Y. Huang, J. Xu, and G. Yang, "Random secure comparator selection based privacy-preserving max/min query processing in two-tiered sensor networks," *Journal of Sensors*, vol. 2016, pp. 1–13, 2015.

[3] J. N. Doctor, J. Vaidya, X. Jiang, W. Shuang, and D. Meeker, "Efficient determination of equivalence for encrypted data," *Computers and Security*, vol. 97, 2018. (`https://doi.org/10.1016/j.cose.2020.101939`)

[4] Y. Dou, H. C. B. Chan, and M. H. Au, "Order-hiding range query over encrypted data without search pattern leakage," *The Computer Journal*, vol. 61, no. 12, pp. 1806–1824, 2018.

[5] A. Dupin, J. M. Robert, and C. Bidan, "Location-proof system based on secure multi-party computations," *IACR Cryptology ePrint Archive*, vol. 2018, pp. 525, 2018.

[6] I. Ioannidis and A. Grama, "An efficient protocol for Yao's millionaires' problem," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003. DOI: 10.1109/HICSS.2003.1174464.

[7] I. Karnil, O. Olakanmi, and S. O. Ogundoyin, "A secure and privacy-preserving lightweight authentication protocol for wireless communications," *Information Systems Security*, vol. 26, no. 4-6, pp. 287–304, 2017.

[8] S. D. Li, Y. Q. Dai, and Q. Y. You, "An efficient solution to Yao's millionaires' problem (in chinese)," *Acta Electronica Sinica*, vol. 33, no. 5, pp. 769–773, 2005.

[9] S. D. Li, S. F. Zho, Y. M. Guo, J. W. Dou, and D. S. Wang, "Secure set computing in cloud environment (in chinese)," *Journal of Software*, 2016. DOI:10.13328/j.cnki.jos.004996.

[10] H. Y. Lin and W. G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," *IACR Cryptology ePrint Archive*, vol. 2005, pp. 43–43, 2005.

[11] J. K. Liu, C. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou, "Time-bound anonymous authentication for roaming networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 178–189, 2014.

[12] M. S. I. Mamun and A. Miyaji, "Secure VANET applications with a refined group signature," in *Twelfth Annual International Conference on Privacy, Security and Trust*, 2014. DOI: 10.1109/PST.2014.6890940.

[13] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword

search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 1–1, 2017.
[14] O. S. Oyinlola and A. S. Oladele, "Edas: Efficient data aggregation scheme for internet of things," *Journal of Applied Security Research*, vol. 13, no. 3, pp. 347–375, 2018.
[15] B. Schoenmakers and P. Tuyls, "Practical two-party computation based on the conditional gate," in *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 3329, pp. 119–136, 2004.
[16] K. Shishido and A. Miyaji, "Secure online-efficient interval test based on empty-set check," in *The 14th Asia Joint Conference on Information Security (AsiaJCIS'19)*, 2019. DOI: 10.1109/AsiaJCIS.2019.000-5.
[17] K. Xue, J. Hong, Y. Xue, D. S. L. Wei, N. Yu, and P. Hong, "Cabe: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding," *IEEE Transactions on Computers*, vol. 66, no. 9, pp. 1491–1503, 2017.
[18] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982. DOI: 10.1109/SFCS.1982.38.
[19] L. Zhang, J. Song, and J. Pan, "A privacy-preserving and secure framework for opportunistic routing in dtns," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7684–7697, 2016.
[20] B. Zhao, S. Tang, X. Liu, X. Zhang, and W. N. Chen, "Ironm: Privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.

Biography

Ya-Ting Duan received the B.S. degree from the North University of China in 2019. She is currently pursuing the M.S. degree in applied mathematics with the School of Mathematics and Statistics, Shaanxi Normal University, Xi'an, China. Her research interests include security protocols and searchable encryption in cloud storage.

Yan-Ping Li received her M.S. degree from Shaanxi Normal University in 2004 and Ph.D. degree from Xidian University in 2009, Xi'an, China. She now is an Associate Professor with the School of Mathematics and Statistics, Shaanxi Normal University. Her research interests include public key cryptography and its applications.

Lai-Feng Lu received M.S. and Ph.D. degrees in Computer System Architecture from Xidian University, Shaanxi, China, in 2005 and 2012, respectively. Now she is an Associate Professor in Shaanxi Normal University. Her research interests include privacy protection and ad hoc network security.

Kai Zhang received the M.S. degree in Applied Mathematics from Shaanxi Normal University in 2013 and Ph.D. degree in from Xidian University in 2017, Shaanxi, China. Now he is a Lecture in Shaanxi Normal University. His research interests include information security and privacy, and applied cryptography.