Supplementary Material for "Secure Wavelet Matrix: Alphabet-Friendly Privacy-Preserving String Search for Bioinformatics"

Hiroki Sudo[1,4] Masanobu Jimbo[1,4] Koji Nuida[2,3]
Kana Shimizu[1,4]
[1] Department of Computer Science and Engineering
School of Fundamental Science and Engineering
Faculty of Science and Engineering,
Waseda University 3-4-1 Okubo Shinjuku-ku Tokyo, 169-8555, Japan
[2] Information Technology Research Institute,
National Institute of Advanced Industrial Science and Technology,
2-4-7 Aomi Koto-ku, Tokyo 135-0064, Japan,
[3] Japan Science and Technology Agency (JST) PRESTO Researcher,
Tokyo, Japan
[4] AIST-Waseda University Computational Bio Big-Data Open Innovation Laboratory,
Tokyo, Japan

S1 Additional search option

We referred to different search options of the secure FM-Index in Section 3.4 . Here, we describe in detail one of those search options that enables searching for the longest substring match whose occurrence is at least ϵ. A server can avoid leaking information about rare substrings in its database to a user by using this search option. Recall that in FM-Index, k-th reported intervals $[f_k, g_k)$ implies k-prefix of a query matches to the database with $g_k - f_k$ positions. Therefore, checking whether occurrence of substring is less than ϵ is equivalent to $g - f - \epsilon < 0$. The key idea of implementation is the design of flags: each flag indicates whether $g - f - i$ $(i = 0, 1, \ldots, \epsilon - 1)$ is 0 or not. i-th flag is 0 iff. k-prefix of a query matches to the database with i positions. Therefore, only one flag will be 0 iff. $g - f - \epsilon < 0$. In practice, all flags are randomized and encrypted.

The outline is as follows:

Server side procedure
The server prepares encrypted flags $\boldsymbol{x}$ as follows:

$$\boldsymbol{x} = \{(\mathsf{Enc}(g_k) \bigoplus \mathsf{Enc}(-f_k) \bigoplus \mathsf{Enc}(-i)) \otimes \mathsf{Enc}(r_i)\},$$

where $i = 0, 1, \ldots, \epsilon - 1$, each r_i is a random value different from the other r_j. The server shuffles and sends $\boldsymbol{x}$ to the user.

User side procedure
The user then decrypts $\boldsymbol{x}$ and checks whether one of the flags is 0 or not (only one flag will be 0 at most). If one of the flags is equal to $\mathsf{Enc}(0)$, the user knows the occurrence of k-prefix substring match is less than ϵ. Note that the user cannot know the exact occurrence of a substring match.

To implement this search option, we replace isLongest with another function isELongest corresponding to the server side procedure above. Also, we need to slightly modify the user side procedure for checking the end condition. A detailed algorithm of modified secure FM-Index is presented in Algorithm S1. isELongest function and Step 3b are mainly modified parts.

Algorithm S1 Detailed description of secure FM-Index.

function isELongest($\mathsf{Enc}(f)$, $\mathsf{Enc}(g), \epsilon$)
  for $i = 0$ **to** $\epsilon - 1$ **step** 1 **do**
    Generate random value r_i.
    $x_i \leftarrow (\mathsf{Enc}(g) \oplus \mathsf{Enc}(-f) \oplus \mathsf{Enc}(-i)) \otimes r_i$
  end for
  $\boldsymbol{x} \leftarrow \{x_0, \ldots, x_{\epsilon-1}\}$
  Shuffle order of elements in $\boldsymbol{x}$
  return $\boldsymbol{x}$ ▷ $x_i = \mathsf{Enc}(0)$ iff. occurrence of match is ϵ.
end function

- Public input: Problem size N; alphabet Σ
- Private input of user: A query sequence $\boldsymbol{q}$ of length ℓ
- Private input of server: A database text T

0. (*Key setup of cryptosystem*) User generates key pair $(\mathsf{pk}, \mathsf{sk})$ by key generation algorithm KeyGen for additive-homomorphic cryptosystem and sends public key pk to server.
1. (*Server initialization*)
   - Server creates BWT of T and stored it as $\hat{T}$.
   - Server creates a set of sub-lookup tables for $\hat{T}$:
     $V = \{\boldsymbol{v}^0, \boldsymbol{v}^1, \ldots, \boldsymbol{v}^{b-1}\}$, by the same process described in Step 1 of Algorithm 1
2. (*User initialization*) Set initial interval $[\hat{f}_0 = 0, \hat{g}_0 = N)$.
3. (*Recursive search*) Initialize an index: $i \leftarrow 0$
   Initialize random factors: $r_f \leftarrow 0, \quad r_g \leftarrow 0$
   while $(i < \ell)$ **do**
   (a) (*Update interval*)
   - The user and the server execute:
     $\hat{f}_{i+1}, \mathsf{Enc}(f_{i+1}), r_f \leftarrow \mathsf{sWM}(\hat{f}_i, \mathsf{pk}, \mathsf{sk}, q[i], V, r_f)$
     $\hat{g}_{i+1}, \mathsf{Enc}(g_{i+1}), r_g \leftarrow \mathsf{sWM}(\hat{g}_i, \mathsf{pk}, \mathsf{sk}, q[i], V, r_g)$
     to obtain:
     $\hat{f}_{i+1}$, $\hat{g}_{i+1}$ for the user,
     $\mathsf{Enc}(f_{i+1})$, $\mathsf{Enc}(g_{i+1})$, r_f, r_g for the server.

   (b) (*Operate*) The server performs the following steps:
   - Compute an encrypted flag showing if the match is longest.
     $\boldsymbol{x} \leftarrow \mathsf{isELongest}(\mathsf{Enc}(f_{i+1}), \mathsf{Enc}(g_{i+1}), \epsilon)$
   - Send $\boldsymbol{x}$ to the user

   (c) (*Decryption of the encrypted flag*) The user performs the following steps:
   Set flag $t \leftarrow 0$ ▷ $t = 1$ if any element of $\boldsymbol{x}$ is equal to $\mathsf{Enc}(0)$, i.e, the occurrence of the match is less than ϵ
   for $j = 0$ **to** $\epsilon - 1$ **step** 1 **do**
     $d \leftarrow \mathsf{Dec}(x_j)$
     if $d = 0$
       $t \leftarrow 1$
   end for
   if $t = 1$
     if $i = 0$ Report that no prefix matches to T at least ϵ positions
     else Reports that $q[0, \ldots, i-1]$ is the longest match
     Sends decoy queries to server until $i = \ell - 1$
   $i \leftarrow i + 1$
   end while

The user reports that $q[0, \ldots, \ell - 1]$ is the longest match, if $t \neq 1$ for $i = 0, \ldots, \ell - 1$.

S2 Security property for our protocol and a proof of the security

In this study, we are supposing that an additively homomorphic encryption scheme used in our proposed protocol is semantically secure; that is, no information in the original message can be learned from a ciphertext. A more precise formulation of the condition requires the notion of *security parameter*, which is a part of public parameters in a protocol but is made implicit in the description throughout the main text for the sake of notational simplicity. A non-negative value $\varepsilon = \varepsilon(\kappa)$ depending on the security parameter κ is said to be *negligible*, if for any integer $k > 0$, we have $\varepsilon < \kappa^{-k}$ for any κ that is larger than a threshold depending on the k. In the following text, the term "polynomially bounded" for some quantity means that the quantity is bounded from above by a certain positive polynomial in the security parameter. Here we say that two random variables x and y, parameterized (implicitly) by the security parameter and also (possibly) by some auxiliary object aux, are *indistinguishable*, if for any probabilistic circuit D of polynomially bounded size with binary output (called a *distinguisher* for x and y), the difference between the probabilities that D outputs 1 for input x and for input y are smaller than a negligible value (in the sense above) that is independent of the choice of the auxiliary object aux. Based on these notions, the semantic security for encryption schemes can be formalized as follows:

Definition 1. *We say that a public-key encryption scheme is* semantically secure*, if for any probabilistic circuit C of polynomially bounded size, there exists a probabilistic circuit S of polynomially bounded size (called a* simulator *for C) with the following property; the two output distributions $C(\mathsf{pk}, \mathsf{Enc}(m))$ and $S(\mathsf{pk})$ are indistinguishable, where m is any plaintext and pk is a public key generated according to the key generation algorithm KeyGen.*

The definition above intuitively means that, any information that can be (efficiently) inferred from a given ciphertext $\mathsf{Enc}(m)$ for original message m can in fact be (efficiently) inferred as well from publicly available information only, without the ciphertext (hence independently of m); this implies that the ciphertext gives essentially no information in the original message. The security property for our proposed protocol adopted in this paper (which is known as the security in the semi-honest model) is defined in a similar manner. Informally, the security property here means that any information that is learned by the user (respectively, the server) during an interactive execution of the protocol can also be inferred solely from the input and the output for the user (respectively, the server) in the protocol; this implies that the user (respectively, the server) learns essentially no information in the server's (respectively, the user's) secret input during the protocol, except for the user's (respectively, the server's) own output in the protocol which is legally available owing to the intended functionality of the protocol. A formal description of this security property can be expressed as follows:

Definition 2. *We say that an interactive protocol between the user and the server is* secure (in the semi-honest model)*, if there exist probabilistic circuits S_{user} and S_{server} of polynomially bounded sizes (called* simulators *for the user and for the server) with the following property; when writing the user's (respectively, the server's) input in the protocol as x_{user} (respectively, x_{server}) and writing the user's (respectively, the server's) corresponding output as $\mathsf{out}_{\mathsf{user}}(x_{\mathsf{user}}, x_{\mathsf{server}})$ (respectively, $\mathsf{out}_{\mathsf{server}}(x_{\mathsf{user}}, x_{\mathsf{server}})$), the output distribution of S_{user} with inputs x_{user} and $\mathsf{out}_{\mathsf{user}}(x_{\mathsf{user}}, x_{\mathsf{server}})$ (respectively, of S_{server} with inputs x_{server} and $\mathsf{out}_{\mathsf{server}}(x_{\mathsf{user}}, x_{\mathsf{server}})$) is indistinguishable from the distribution of the list of all objects obtained by the user (respectively, the server) during an execution of the protocol, which consists of the random number for the user (respectively, the server) and all the messages sent from the server to the user (respectively, from the user to the server).*

Now we have the following result:

Theorem 1. *Suppose that the additively homomorphic encryption scheme used in our protocol is semantically secure. Suppose moreover that, the plaintext space of the encryption scheme is of the form $\mathbb{Z}/\widetilde{N}\mathbb{Z}$ with integer $\widetilde{N} > 0$ that is coprime to any integer from 1 to N, where N denotes the public problem size[1]. Then our proposed secure FM-Index protocol, described in Algorithm 2 of the main text, is secure in the sense above. (We note that the length ℓ of the user's query sequence $\boldsymbol{q}$ is regarded here as public information.)*

Proof. First we note that, the number of loops in Step 3 of Algorithm 2 depends solely on the query length ℓ and is independent of the contents of the query $\boldsymbol{q}$ and the database T. Similarly, the numbers of loops in the subroutine sWM executed in Step 3(a) of Algorithm 2 depend solely on the known size of the alphabet Σ and are independent of the contents of the query $\boldsymbol{q}$ and the database T.

For the security of the user's secret input against the server, first note that the messages received by the server during Algorithm 2 (except for a public key pk) are those received during the subroutine sWM only, which are ciphertexts received at Steps 3(b) and 4 in Algorithm 1. Then a simulator S_{server} for the server to

[1]This technical assumption is indeed satisfied in practical situations, where the underlying encryption scheme is either the "lifted" version of the ElGamal cryptosystem (where $\widetilde{N}$ is a large prime number) or the Paillier cryptosystem (where $\widetilde{N}$ is the product of two prime numbers both having over 500-bit lengths).

prove the security against the server can be constructed by combining the semantic security of the underlying encryption scheme with so-called "hybrid argument" which is a famous proof technique in cryptography. An outline of the argument is explained in the following two paragraphs.

First, we set the circuit C in the definition of the semantic security of the encryption scheme to be the circuit outputting the given ciphertext itself, and take the corresponding simulator S. Then the distribution of the list of ciphertexts received by the server during the protocol remains indistinguishable when a random output of S (that is computable by using public information only) is used instead of the first ciphertext received by the server during the protocol. Indeed, assume for the contrary that these two distributions are distinguishable, hence there is a distinguisher D for those distributions that yields a non-negligible difference of the probabilities to output 1. Then we can construct a distinguisher D' for the output distributions of C and S in a way that D' executes the protocol by emulating the roles of both the user and the server, where the given output of either C or S is used as the first ciphertext received by the server (emulated by D'), and then D' executes D and outputs the output of D. Now the construction of D' implies that the probabilities that D' outputs 1 with input given by C and by S are equal to the corresponding output probabilities of D with the original and the modified list of ciphertexts, hence these two probabilities have non-negligible difference due to the aforementioned choice of D. The existence of such a distinguisher D' contradicts the semantic security of the underlying encryption scheme.

We have shown in the previous paragraph that the distribution of the list of ciphertexts received by the server during the protocol remains indistinguishable when the first ciphertext among them is replaced by a certain object that is computable by using public information only. A similar argument (combined with the fact that the sum of two negligible values is also negligible) also shows that the distribution of the list of ciphertexts received by the server remains indistinguishable as well when the second ciphertext among them is also replaced by a certain object that is computable by using public information only. By repeating this process to replace all of the ciphertexts one by one, it follows that the distribution of the list of ciphertexts received by the server during the protocol is indistinguishable from the list of certain objects that are all computable by using public information only. Now we can construct a simulator S_{server} for the server satisfying the condition in the definition of the security in the semi-honest model, by using the latter list of publicly computable objects as the list of messages in the output of S_{server}. This implies the security of the user's secret input against the server.

From now, we consider the security of the server's secret input against the user. First note that, any message sent from the server to the user during the protocol is a ciphertext for a plaintext m of one of the following forms:

- $m = m^{\dagger} + r \bmod N + 1$, where $m^{\dagger}$ is a certain value (possibly correlated with the server's secret input) and r is an independent and uniformly random value chosen by the server and known to the server only.
- $m = (g_{i+1} - f_{i+1}) \cdot r \bmod \widetilde{N}$, where $\widetilde{N}$ is the size of the plaintext space as assumed in the statement, f_{i+1} and g_{i+1} are as in the function $\mathsf{isLongest}$ executed in Step 3(b), and r is an independent and uniformly random value chosen by the server and known to the server only.

For the former case of plaintext m, the plaintext m is independent and uniformly random from the user's viewpoint regardless of the value of $m^{\dagger}$, since the additive mask r is independent and uniformly random and is known to the server only.

On the other hand, for the latter case of plaintext m, let $0 \leq \ell_{\mathsf{match}} \leq \ell$ denote the length of the longest match obtained as the output by the user, i.e., $q[0, \ldots, \ell_{\mathsf{match}} - 1]$ is the longest match. Then by the construction of the protocol, we have $f_{i+1}, g_{i+1} \in \{0, \ldots, N\}$, and we have $f_{i+1} = g_{i+1}$ if and only if $i = \ell_{\mathsf{match}}$. Therefore, we have $m = (g_{i+1} - f_{i+1}) \cdot r \bmod \widetilde{N} = 0$ (regardless of r) if $i = \ell_{\mathsf{match}}$. On the other hand, when $i \neq \ell_{\mathsf{match}}$, we have $1 \leq |g_{i+1} - f_{i+1}| \leq N$; now the assumption on the size $\widetilde{N}$ of the plaintext space in the statement implies that $g_{i+1} - f_{i+1}$ is coprime to $\widetilde{N}$ and hence $m = (g_{i+1} - f_{i+1}) \cdot r \bmod \widetilde{N}$ is independent and uniformly random from the user's viewpoint.

According to the arguments above, we can construct a simulator S_{user} for the user in a way that S_{user} outputs a random ciphertext for plaintext 0 as the message received at Step 3(b) of Algorithm 2 with $i = \ell_{\mathsf{match}}$ and S_{user} outputs a random ciphertext for random plaintext as any other message received during the protocol. Then the arguments above show that the list of simulated messages generated by S_{user} is indistinguishable from the original messages received by the user during the protocol, which implies the security of the server's secret input against the user. This completes the proof. □

We note that, the proof above shows that the cryptographic assumption on the semantic security of the underlying additive homomorphic encryption scheme is relevant only to the security for the user's secret input against the server in our protocol; the security for the server's secret input against the user is proved without any cryptographic assumption (provided the server's random number is assumed to be ideally random).

S3 More detailed description of the recursive oblivious transfer

The recursive oblivious transfer problem (ROT) is defined as follows:

Model[n1] 1. *A user has a private value $0 \leq x_1 < N$ and a server has a private vector $\boldsymbol{v}$ of length N. Let us denote $x_{k+1} = v[x_k]$ and the user is allowed to access the server $\ell - 1$ times. After the calculation, the user learns only x_ℓ and the server learns nothing about $x_1, \ldots, x_\ell$.*

Here we describe the linear-communication-size algorithm for the ROT. In the initial step, the user creates an encrypted bit vector query as follows:

$$\vec{\mathsf{Enc}}(\boldsymbol{q}) = (\mathsf{Enc}(q_0 = 0), \ldots, \mathsf{Enc}(q_{x_1} = 1), \ldots, \mathsf{Enc}(q_{N-1} = 0)) \,, \tag{1}$$

where x_1 is the index of the element that the user wants to search with in the next round.

After the user sends $\vec{\mathsf{Enc}}(\boldsymbol{q})$ to the server, the server generates a random value $r \in \{0, 1, \ldots, N-1\}$ and returns the ciphertext:

$$\hat{c} = \bigoplus_{i=0}^{N-1} \big(((\boldsymbol{v}[i] + r)_{\mathsf{mod}N}) \otimes \mathsf{Enc}(q_i)\big) = \mathsf{Enc}((x_2 + r)_{\mathsf{mod}N}),$$

to the user, where $x_2 = \boldsymbol{v}[x_1]$ and $(a+b)_{\mathsf{mod}N}$ denotes addition in a number field modulo N. Note that the user decrypts it and knows a *randomized result* $(x_2 + r)_{\mathsf{mod}N}$, instead of knowing a correct result.

The user creates the next query in the same manner when he/she created the initial one, as follows:

$$\vec{\mathsf{Enc}}(\hat{\boldsymbol{q}}) = (\mathsf{Enc}(\hat{q}_0 = 0), \ldots, \mathsf{Enc}(\hat{q}_{((x_2+r)_{\mathsf{mod}N})} = 1), \ldots, \mathsf{Enc}(\hat{q}_{N-1} = 0)).$$

The key observation is that the query $\vec{\mathsf{Enc}}(\hat{\boldsymbol{q}})$ is the r-rotated permutation of the *true* query, in which x_2-th element is an encryption of 1 and rest of all elements are an encryption of 0. Therefore, the server can create the true query by permutating it such that i-th element moves to $((i - r)_{\mathsf{mod}N}))$-th position. By repeating above steps, the user and the server can successfully compute $x_{ell} = \boldsymbol{v}[\boldsymbol{v}[\ldots \boldsymbol{v}[x_1] \ldots]]$.

As is apparent from the equation (1), the communication complexity of the above algorithm is $O(N)$. To improve the complexity from $O(N)$ to $O(\sqrt{N})$, following algorithm was proposed in the previous study [5]. The key idea of the algorithm is to transform each element or position x in the vector $\boldsymbol{v}$ into two values $t_0 = x \,/\lceil\sqrt{N}\,\rceil$ and $t_1 = (x)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil}$, and search them respectively. To this end, the user transforms the initial query x_1 into two values $t_0 = x_1 \,/\lceil\sqrt{N}\,\rceil$ and $t_1 = (x_1)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil}$ and creates an encrypted bit vector:

$$\vec{\mathsf{Enc}}(\boldsymbol{q}) = (\mathsf{Enc}(q_0 = 0) \ldots, \mathsf{Enc}(q_{t_1} = 1), \ldots, \mathsf{Enc}(q_{\lceil\sqrt{N}\,\rceil - 1} = 0)),$$

and an encrypted value $\mathsf{Enc}(t_0)$.

The server also transforms $\boldsymbol{v}$ into two matrices $V_0, V_1 \in \mathbb{N}^{\lceil\sqrt{N}\,\rceil \times \lceil\sqrt{N}\,\rceil}$:

$$V_0[i,\ j] = \boldsymbol{v}[i \times \lceil\sqrt{N}\,\rceil + j]/\lceil\sqrt{N}\,\rceil,$$

$$V_1[i,\ j] = (\boldsymbol{v}[i \times \lceil\sqrt{N}\,\rceil + j])_{\mathsf{mod}\lceil\sqrt{N}\,\rceil},$$

for $i = 0, \ldots, \lceil\sqrt{N}\,\rceil - 1$ and $j = 0, \ldots, \lceil\sqrt{N}\,\rceil - 1$.

After the server receives the query $\mathsf{Enc}(t_0)$ and $\vec{\mathsf{Enc}}(\boldsymbol{q})$ from the user, he/she computes:

$$\hat{c}_{0,k} = \bigoplus_{i=0}^{\lceil\sqrt{N}\,\rceil - 1} \big((V_0[k,\ i] + r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil} \otimes \mathsf{Enc}(q_i)\big) \oplus \big(r_{0,k} \otimes \mathsf{Enc}(t_0 - k)\big),$$

$$\hat{c}_{1,k} = \bigoplus_{i=0}^{\lceil\sqrt{N}\,\rceil - 1} \big((V_1[k,\ i] + r_1)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil} \otimes \mathsf{Enc}(q_i)\big) \oplus \big(r_{1,k} \otimes \mathsf{Enc}(t_0 - k)\big),$$

for $k = 0, \ldots, \lceil\sqrt{N}\,\rceil - 1$, where r_0, r_1, $r_{0,k}$ and $r_{1,k}$ are random values, and returned encrypted vectors $\hat{\boldsymbol{c}}_0$ and $\hat{\boldsymbol{c}}_1$ to the user.

The important observation here is that only two elements $\hat{c}_{0,t_0}$ and $\hat{c}_{1,t_0}$ of $\hat{\boldsymbol{c}}_0$ and $\hat{\boldsymbol{c}}_1$ are encryption of $\hat{t}_0 = (V_0[t_0,\ t_1] + r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil}$ and $\hat{t}_1 = (V_1[t_0,\ t_1] + r_1)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil}$, and rest of all elements are random values. Note that user wants to search x_2-th element of $\boldsymbol{v}$ which corresponds to $V_0[t_0,\ t_1]$ and $V_1[t_0,\ t_1]$, and if the user creates the next query in the same manner when he/she creates the initial one, $\vec{\mathsf{Enc}}(\boldsymbol{q})$ is the r_1-rotated permutation of the *true* query and either $\mathsf{Enc}(\hat{t}_0 - (r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil})$ or $\mathsf{Enc}(\hat{t}_0 - (r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil} + \lceil\sqrt{N}\ \rceil)$ is the *true* query. Therefore, as similar to the linear communication complexity algorithm, the server can create the true query by permutating $\vec{\mathsf{Enc}}(\boldsymbol{q})$ such that i-th element moves to $((i - r_1)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil})$-th position, and uses $\mathsf{Enc}(\hat{t}_0 - (r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil})$ and $\mathsf{Enc}(\hat{t}_0 - (r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil} + \lceil\sqrt{N}\ \rceil)$ to compute next $\hat{\boldsymbol{c}}_0$ and $\hat{\boldsymbol{c}}_1$. (The server does not know either $\mathsf{Enc}(\hat{t}_0 - (r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil})$ or $\mathsf{Enc}(\hat{t}_0 - (r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil} + \lceil\sqrt{N}\ \rceil)$ is correct, therefore, he/she uses both values to compute two different results. Note that one of them is a correct result and the other is random result, and no additional information is leaked to the user.) Since the user should not learn r_0, the server needs to permutate the results $\hat{\boldsymbol{c}}_0$ and $\hat{\boldsymbol{c}}_1$ such that i-th element moves to $((i + r_0)_{\mathsf{mod}\lceil\sqrt{N}\,\rceil})$-th position.

We would like to note that the private information retrieval (PIR) [1] aims a similar problem to the 1-out-of-N oblivious transfer (OT), and many studies have been published for the problem. (See the survey papers such as [2], [4] and [6].) Among the several sub-problems of the PIR, the single database symmetric computational private information retrieval problem is equivalent to OT, and we notice that the previous study achieves polylogarithmic communication complexity for OT [3], and it is an interesting topic to try to design ROT algorithm based on such an efficient algorithm to improve the complexity furthermore.

S4 Characters used in the experiments

Table 1 shows a set of characters and corresponding code points of Unicode that were used in the experiments. We used a CJK unified ideographs table which is included in Unicode version 8.0, because it contains most of the Chinese ideographs that are commonly used in Japan.

Table 1: Unicode code points of characters included in Clinical DB1 and DB2

	Unicode code point	DB name
Arabic numerals	0x0030 - 0x0039	Clinical DB1, 2
Roman alphabet (lower case)	0x0041 - 0x005A	Clinical DB1, 2
Greek alphabet (upper case)	0x0391 - 0x03A9	Clinical DB1, 2
Greek alphabet (lower case)	0x03B1 - 0x03C9	Clinical DB1, 2
Hiragana	0x3041 - 0x3093	Clinical DB1, 2
Symbols for long vowel sound	0x30FC	Clinical DB2
Katakana	0x30A1 - 0x30F6	Clinical DB2
Chinese ideograph (CJK unified ideographs table)	0x4E00 - 0x9FD5	Clinical DB2

References

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, pp. 41–50, Washington, DC, USA, 1995. IEEE Computer Society.

[2] William Gasarch. A survey on private information retrieval. *Bulletin of the EATCS*, 82:72–107, 2004.

[3] Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Communication. In *Information Security, 8th International Conference, ISC 2005, Singapore, September 20-23, 2005, Proceedings*, pp. 314–328, 2005.

[4] Rafail Ostrovsky and William E. Skeith, III. A survey of single-database private information retrieval: Techniques and applications. In *Proceedings of the 10th International Conference on Practice and Theory in Public-key Cryptography*, PKC'07, pp. 393–411, Berlin, Heidelberg, 2007. Beijing, China, Springer-Verlag.

[5] Kana Shimizu, Koji Nuida, and Gunnar Rätsch. Efficient privacy-preserving string search and an application in genomics. *Bioinformatics*, 32(11):1652–1661, 2016.

[6] Sergey Yekhanin. Private information retrieval. *Commun. ACM*, 53(4):68–73, April 2010.