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 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