doi: 10.12068/j.issn.1005-3026.2019.04.002

基于后缀树的基因数据可搜索加密方法

秦诗悦[1], 周福才[2], 柳 璐[2]
(1. 东北大学 计算机科学与工程学院, 辽宁 沈阳 110169; 2. 东北大学 软件学院, 辽宁 沈阳 110169)

摘 要: 为保障用户免遭侵犯隐私的风险,提出了一种特别支持基因数据的可搜索加密方法. 针对目前密文搜索方案大多数仅支持通过关键字进行搜索,而无法用于不含关键字的基因数据的问题,利用后缀树和伪随机函数等密码学原语构建安全索引,实现对密文基因数据的任意子字符串搜索. 安全性证明该方法满足动态自适应安全,利用理论分析和真实数据对效率进行测评. 该方法可以对基因数据进行高效安全的任意子字符串搜索,保护数据完整性和隐私性,在个性化医疗大众化的环境下具备广阔的应用前景.

关 键 词: 基因数据;后缀树;可搜索加密;子字符串搜索;现代医疗
中图分类号: TP 309 **文献标志码**: A **文章编号**: 1005-3026(2019)04-0461-06

Searchable Encryption Scheme of Genomic Data Based on Suffix Tree

QIN Shi-yue[1], *ZHOU Fu-cai*[2], *LIU Lu*[2]
(1. School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China; 2. School of Software, Northeastern University, Shenyang 110169, China. Corresponding author: ZHOU Fu-cai, E-mail: fczhou@mail.neu.edu.cn)

Abstract: A searchable encryption scheme that specifically supports genetic data was proposed to protect users from privacy risks. The existing searchable encryption schemes only support keywords search, they cannot be applied to genetic data without keywords. So, a security index using cryptographic primitives such as suffix trees and pseudo-random functions was constructed to implement arbitrary substring searches for ciphertext genomic data. The safety proof indicated that the method satisfies the dynamic adaptive security, and the efficiency is evaluated by both theoretical analysis and real data. This scheme can perform efficient and safe arbitrary substring search on genomic data, protect data integrity and privacy and has broad application prospects in the environment of personalized medical popularization.

Key words: genomic data; suffix tree; searchable encryption; substring search; modern medicine

随着人类基因组测序成本的下降,个性化医疗将逐渐成为患者的首选. 通过对患者进行基因测试获得患者的基因数据,选择更具针对性的治疗方案. 通过搜索患者的基因序列中是否存在突变基因,计算基于基因数据的子字符串匹配问题,决定是否需要采取早期干预措施来预防遗传疾病发生. 基因数据的特性之一是其序列长度十分庞大. 为了对这样的海量数据进行存储和计算,许多基因医疗机构选择外包给云服务器或者第三方代理服务器. 然而,基因数据作为个人可标识信息,患者将其提供给不可信第三方就可能会把自己置身于隐私侵犯和数据泄露的危险之中[1].

为了保证基因数据的隐私性,以及防止数据泄露带来的隐私威胁问题,在数据外包之前进行加密是有效技术手段. 同时要求一个不可信实体(利用不可信服务器)能够完成对密文基因数据的子字符串搜索工作. 传统的可搜索加密[2-3]通过预设关键字构造索引实现密文搜索,但在基因数据中无法给出明确的关键字,同时待搜索目标基因数据可能重复性较高,因此无法直接应用.

收稿日期: 2018-02-28
基金项目: 国家自然科学基金资助项目(61772127,61872069); 中央高校基本科研业务费专项资金资助项目(N151704002).
作者简介: 秦诗悦(1994-),女,辽宁沈阳人,东北大学博士研究生; 周福才(1964-),男,吉林长春人,东北大学教授,博士生导师.

Lu 等[4]提出一个全基因组关联研究的安全外包方案,该方案根据已知的基因突变和疾病间关联计算加密的统计数值作健康风险评估. Chase 等[5]提出的方案会泄露搜索模式,仅实现部分隐私保护. Ayday 等[6]通过保序加密构建保护隐私的 DNA 检索方案. 混淆电路[7]和差分隐私技术[8-9]同样可以用来保护基因数据,但是噪音问题会干扰临床医生给出正确治疗方案,同时影响方案效率. Wang 等[10]利用属性加密解决序列匹配问题,但无法获得具体目标基因位置信息.

本文针对组成元素简单、重复部分多且数据量大的基因数据,提出一个基于后缀树的基因数据可搜索加密方法. 通过构建具有后缀树结构的、支持子字符串搜索的安全索引,利用伪随机函数等保证数据机密性与隐私性,实现一个能在任何基因序列中搜索任意基因片段,并返回其所有位置信息的可搜索加密方案.

1 预备知识

1.1 后缀树

后缀树(suffix tree)是一种可快速实现多种重要字符串操作的数据结构[11]. 一个长度为 n 的字符串 $s = s_1 s_2 \cdots s_n$ 的后缀树是一棵包含根节点的树,由节点和边组成. 具备以下属性:

1) 每条边代表一个非空的 s 子字符串.

2) 若非叶子节点只有 1 个孩子节点,去掉此孩子节点进行压缩. 后缀树最多为 $2n$ 个节点.

3) 从根到叶子节点 i 的路径与 s 从 i 位置开始的后缀对应,即每条路径唯一代表 s 的一个后缀.

4) 同一节点下每条边以不同的字符开头[12].

后缀树的子字符串搜索速度较快,对于长度为 n 字符串,最多花费 $O(n)$ 时间进行匹配.

1.2 伪随机函数

对长字符串计算伪随机函数,需要先对长字符串做 Hash 处理,使之变成定长字符,然后对此 Hash 值计算伪随机函数. 如果 Hash 具有通用抗碰撞性,则该伪随机函数的伪随机性就依然成立.

设有限域集 D, R, U, B, 其中 D, U 是定义域,R, B 是结果集. λ, μ 为密钥长度,存在伪随机函数 $F: \{0,1\}^{\lambda} \times D \to R$ 和通用抗碰撞 Hash 函数 $H: \{0,1\}^{\mu} \times U \to B$, 输入长字符串 x, 则函数 $F(H(x)): \{0,1\}^{\lambda+\mu} \times U \to R$ 是一个合法伪随机函数.

2 模型与定义

本文考虑用户 User 与不可信服务器 Ser 进行交互,对密文基因数据进行子字符串搜索,确定某段特定标记序列是否出现及出现的位置. 给定基因数据字符串 s 和一个待搜索字符串 p,找出 p 作为 s 子字符串出现的所有情况.

2.1 形式化定义

基于后缀树的基因数据可搜索加密方法主要由 5 个多项式时间算法/协议组成,形式化定义为

SDNAsearch = (GenKey, STree, EncInd, GenTok, Search).

各算法具体描述如下:

1) $K \leftarrow$ GenKey(1^{λ}):密钥生成算法,输入安全参数 λ,输出所需密钥组 K.

2) $T \leftarrow$ STree(s):后缀树构建算法,输入明文字符串 s,输出 s 的后缀树结构 T.

3) SI $\leftarrow$ EncInd(K, T, s, ε):安全索引构建算法,输入密钥 K、明文字符串 s、其后缀树结构 T 及加密方案 ε;输出安全索引 SI = (D, C, L). 其中,D 为加密字典结构,C 为加密文件信息,L 为加密数组信息,SI 上传至服务器保存.

4) Tok $\leftarrow$ GenTok(K, p, ε):令牌生成算法,输入待搜索子字符串 p,令牌所需密钥 K,构建安全索引所用加密方案 ε,输出为令牌 Tok.

5) User:Rst $\leftarrow$ Search(User:Tok;Ser:SI):交互搜索协议,用户向服务器发送令牌 Tok,服务器根据 SI 进行搜索,交互结束用户输出结果 Rst.

2.2 安全性定义

假设服务器 Ser 为一个半诚信多项式时间敌手 A,具有自适应性,即 A 可以任意选择信息加密并执行搜索协议进行质询,并可以根据已得质询结果发起新质询. 定义多项式时间模拟器 S 生成随机字符串. 定义 2 个泄漏函数 L_1, L_2:L_1(SI)为加密过程的泄露函数,是安全索引 SI 的泄漏信息;L_2(SI, $p_1, \cdots, p_j$)为搜索过程中的泄露函数,是前 j 次搜索的泄漏信息,p_i 为第 i 次搜索.

构建现实实验 $\text{Real}_{A,U}(\lambda)$:敌手 A 和用户 U 利用真实加密方法进行交互. 对 A 选择的字符串 s,U 生成安全索引 SI 并发送给 A. A 根据 SI 进行多次自适应质询,即待搜索字符串 $p_1, \cdots, p_t$(t 为多项式级). 每次质询 p_i 后,U 利用 Tok 算法计算令牌 Tok_i 并返回给 A.

构建理想实验 $\text{Ideal}_{A,S}(\lambda)$:敌手 A 和有状态的模拟器 S 进行理想交互. 对于 A 选择的字符串

s,S 以泄露函数 L_1,L_2 作为输入生成随机结构 SI′发送给 A,A 根据 SI′自适应进行多次质询 p_1,…,p_t(t 为多项式级). 每次质询 p_i 后,S 利用生成随机字符串 v_i,返回给 A.

若在多项式时间内,敌手 A 以可忽略的概率区分实验输出结果,则称本文构建的方法在泄露函数 L_1,L_2 下是满足动态自适应安全的.

3　详细描述

3.1　符号说明

选择安全参数 λ,满足密钥不可区分性及 IND－CPA 安全的对称加密算法 $\varepsilon=(\mathrm{Gen},\mathrm{Enc},\mathrm{Dec})$,伪随机函数 $F:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{\lambda}$,伪随机置换 $P:\{0,1\}^{\lambda}\times[n]\rightarrow[n]$,$P':\{0,1\}^{\lambda}\times[d]\rightarrow[d]$,$\pi:\{0,1\}^{\lambda}\times[m]\rightarrow[m]$ 和 $\pi':\{0,1\}^{\lambda}\times[\mathrm{num}]\rightarrow[\mathrm{num}]$. 定义基因数据字符集 Σ(例如:{A, T, G, C}),集合所含字符个数 $|\Sigma|=d$,各符号定义如表 1 所示.

表 1　符号说明

Table 1　Symbols description

符号	表示
s	长度为 n 基因数据字符串
p	长度为 m 待搜索字符串
u	后缀树节点
$\mathrm{par}(u)$	u 的父亲节点
$\mathrm{child}(u,i)$	u 的第 i 个孩子节点
$\mathrm{de}(u)$	分支度(所有孩子节点的数量)
$p(u)$	从根节点到 u 路径上的字符串和
$\hat{p}(u)$	从根节点到 $\mathrm{par}(u)$路径上的字符串连接从 $\mathrm{par}(u)$到 u 路径上的第一个字符
leaf_i	第 i 个叶子节点(从左到右,$1\leqslant i\leqslant n$)
len_u	$\hat{p}(u)$的长度
ind_u	p 在 s 中第一次出现的位置的下标
$\mathrm{leftleaf}_u$	u 的最左子后裔的位置
num_u	$p(u)$作为子字符串重复出现的次数

3.2　算法描述

基于后缀树的基因数据可搜索加密方法的详细算法及协议描述如下.

1) $K\leftarrow\mathrm{GenKey}(1^{\lambda})$:密钥生成算法由 User 执行. 随机生成 $K_D,K_C,K_L,K_1,K_2,K_3,K_4\xleftarrow{R}\{0,1\}^{\lambda}$7 个不同比特串组成密钥组 K. K_D,K_C,K_L 分别是字典结构 D、密文数组 C 及叶子数组 L 的密钥;K_1,K_2 用于处理字典内数据以及令牌 Tok;K_3,K_4 用于伪随机置换. 输出用户密钥 K.

2) $T\leftarrow\mathrm{STree}(s)$:后缀树构建算法由 User 执行. 首先,在字符串 $s=s_1\cdots s_n\in\Sigma^n$ 末尾添加特殊字符′$′,避免后缀隐藏问题. 构建一棵空树,每插入一个字符 s_i,利用后缀指针 SuffixLink 进行树结构变换,生成后缀树 T.

算法伪代码如下:

Algorithm $T\leftarrow\mathrm{STree}(s)$

Input: s

Output: T

```
For the s[i…n] ∈ s
    Add character ′$′ and set num = 0
Create root node u0 = (null,0,0)
Set SuffixLink = null
For i = 1:n + 1 do
    If p̂(u)[0] = = si, set num + +
    Else create new node u′ and set SuffixLink
    If u. SuffixLink! = null,
        set SuffixLink to CreateNode(ui)
    Insert(si, leafi)
Return T
```

3) $\mathrm{SI}\leftarrow\mathrm{EncInd}(K,T,s,\varepsilon)$:安全索引构建算法由 User 执行. 根据 $s=s_1\cdots s_n\in\Sigma^n$ 的后缀树 T 构建安全索引 $\mathrm{SI}=(D,C,L)$,其中 D 是字典,C 是密文数组,L 是叶子数组. 将安全索引 SI 上传至 Ser.

①字典 D:以字符串 $s=$"agaggtc"为例构建初始字典 D 如表 2 所示. 然而,其存在中断搜索的错误情况. 因此,利用 $\hat{p}(u)=p(\mathrm{par}(u))\,\|\,\omega$ 代替 $p(u)$构建字典 D,ω 是 $\mathrm{par}(u)$与 u 连接边上的第一个字符. 改进字典 D 如表 3 所示.

在字典 $D=(\mathrm{key},\mathrm{value})$中,key 存储后缀树 T 的每个节点入口,即 $f_1(u)=F(K_1,\hat{p}(u))$. value 值由搜索结构和密文结构组成. 对每个节点 u 的每个孩子节点 $\mathrm{child}(u,i)$ 计算 $f_{2,i}(u)=F(K_2,\hat{p}(\mathrm{child}(u,i)))$. 若 $\mathrm{de}(u)<d$,添加随机项$\{0,1\}^{\lambda}$ 进行补充,并利用伪随机置换 P'进行顺序随机化,得到搜索结构 $f_{2,P'(1)}(u),\cdots,f_{2,P'(d)}(u)$. 利用 K_D 加密得到密文结构,$X_u=\varepsilon$. $\mathrm{Enc}(K_D,(\mathrm{ind}_u,\mathrm{leftleaf}_u,\mathrm{num}_u,\mathrm{len}_u,f_1(u),f_{2,P'(1)}(u),\cdots,f_{2,P'(d)}(u)))$,则 $\mathrm{value}=(f_{2,P'(1)}(u),\cdots,f_{2,P'(d)}(u),X_u)$. 同时为保障字典 D 的安全性,除了 T 现存的 N 个节点入口,另外构建 $2n-N$ 个虚拟入口,每个虚拟入口利用随机项 $f_1,f_{2,1},\cdots,f_{2,d}\xleftarrow{\mathrm{Rand}}\{0,1\}^{\lambda}$填充,即:$\mathrm{key}=f_1$,$\mathrm{value}=(f_{2,1},\cdots,$

$f_{2,d}, \varepsilon.\mathrm{Enc}(K_D, 0))$.

②密文数组 C:利用密钥 K_C 和伪随机置换 $P_{K_3}:[n]\to[n]$对字符串 s 进行确定性加密. 保证密文唯一,生成 $C[P_{K_3}(i)] = \varepsilon.\mathrm{Enc}(K_C, s_i \| i)$.

③叶子数组 L:为返回 p 在 s 中的全部位置指针,构建叶子数组 L. 利用密钥 K_L 和伪随机置换 $P_{K_4}:[n]\to[n]$,生成 $L[P_{K_4}(i)] = \varepsilon.\mathrm{Enc}(K_L, \mathrm{ind}_{\mathrm{leaf}_i} \| i)$.

4) Tok←GenTok(K, p, ε):令牌生成算法由 User 执行. 由 $p = p_1, \cdots, p_m$ 计算 Tok = $T_1, \cdots, T_m$, $f_0(\in), f_0(\in)$是后缀树根节点的密文,即字典 D 入口,$T_i = \varepsilon.\mathrm{Enc}_{F(K_1, p[1\cdots i])}(F(K_2, p[1\cdots i]))$.

5) User:Rst←Search(User:Tok;Ser:SI):交互搜索协议由 User 和 Ser 进行. User 发送 Tok 给 Ser. Ser 根据 $f_1(\in)$找到字典 D 中相等的入口 key 值,并利用对应 value 值中的搜索结构对 $T_i(i = 1, \cdots, m)$进行解析,即计算 $Y = \varepsilon.\mathrm{Dec}(f_{2,j}, T_i)(j = 1, \cdots, d)$. 若 Y 有效且能与字典 D 的某个 key 值匹配,则记录其对应 value 值的密文结构 X,并利用该 value 值中的搜索结构解析下一个令牌 T_{i+1};若 Y 无效或者没有 key 值与有效 Y 相等,则继续利用同一个搜索结构解析 T_{i+1}. 在所有令牌解析完成后,对字典的搜索结束. Ser 发送最终记录的密文结构 X 至 User.

表 2 初始字典示例

Table 2 Example of initial dictionary

节点	key	value
u_1	$F(K_1, \in)$	$\mathrm{Enc}(K_D, 0, \cdots)$
u_2	$F(K_1, "ag")$	$\mathrm{Enc}(K_D, 0, \cdots)$
u_3	$F(K_1, "c")$	$\mathrm{Enc}(K_D, 6, \cdots)$
u_4	$F(K_1, "g")$	$\mathrm{Enc}(K_D, 1, \cdots)$
u_5	$F(K_1, "tc\,\$")$	$\mathrm{Enc}(K_D, 5, \cdots)$
u_6	$F(K_1, "\,\$")$	$\mathrm{Enc}(K_D, 7, \cdots)$
u_7	$F(K_1, "agaggtc\,\$")$	$\mathrm{Enc}(K_D, 0, \cdots)$
u_8	$F(K_1, "aggtc\,\$")$	$\mathrm{Enc}(K_D, 2, \cdots)$
u_9	$F(K_1, "gaggtc\,\$")$	$\mathrm{Enc}(K_D, 1, \cdots)$
u_{10}	$F(K_1, "ggtc\,\$")$	$\mathrm{Enc}(K_D, 3, \cdots)$
u_{11}	$F(K_1, "gtc\,\$")$	$\mathrm{Enc}(K_D, 4, \cdots)$

如果 User 解密 $W = \varepsilon.\mathrm{Dec}(K_D, X)$得到有效明文,首先检验 $f_1 = F(K_1, p[1\cdots i])$是否成立,并判断在 $a = i+1, \cdots, m$ 且 $b = 1, \cdots, d$ 时,$\varepsilon.\mathrm{Dec}(f_{2,b}, T_a) = \perp$ 是否成立. 若皆成立,则 $p[1\cdots i]$是最长匹配前缀,计算 $x_{\pi(i)} = P(K_3, \mathrm{ind} + i - 1)$并发送给 Ser,否则,搜索失败. Ser 根据$(x_1, \cdots, x_m)$搜索 C,返回$(C_1, \cdots, C_m)$. 如果 User 解密 $Y = \varepsilon.\mathrm{Dec}(K_C, C_{\pi_1}(i))$得到有效值 $Y = (p'_i \| j)$,判断 $j = \mathrm{ind} + i - 1$ 且 $p'_1 \cdots p'_m = p$ 是否成立;若成立,计算 $y_{\pi'(i)} = P_{K_4}(\mathrm{leftleaf} + i - 1)$发送给 Ser,否则,搜索失败. Ser 根据$(y_1, \cdots, y_{\mathrm{num}})$搜索 L,返回 $L[y_1], \cdots, L[y_{\mathrm{num}}]$. User 若解密 Rst = $\varepsilon.\mathrm{Dec}(K_L, L[y_i])$得到 $\perp$,或者 Rst = (a_i, j)中存在 $j \neq \mathrm{leftleaf} + i - 1$,则代表搜索失败;否则,结果 Rst = $\{a_1, \cdots, a_{\mathrm{num}}\}$.

表 3 改进字典示例

Table 3 Example of improved dictionary

节点	key	value
u_1	$F(K_1, \in)$	$\mathrm{Enc}(K_D, 0, \cdots)$
u_2	$F(K_1, "a")$	$\mathrm{Enc}(K_D, 0, \cdots)$
u_3	$F(K_1, "c")$	$\mathrm{Enc}(K_D, 6, \cdots)$
u_4	$F(K_1, "g")$	$\mathrm{Enc}(K_D, 1, \cdots)$
u_5	$F(K_1, "t")$	$\mathrm{Enc}(K_D, 5, \cdots)$
u_6	$F(K_1, "\,\$")$	$\mathrm{Enc}(K_D, 7, \cdots)$
u_7	$F(K_1, "aga")$	$\mathrm{Enc}(K_D, 0, \cdots)$
u_8	$F(K_1, "agg")$	$\mathrm{Enc}(K_D, 2, \cdots)$
u_9	$F(K_1, "ga")$	$\mathrm{Enc}(K_D, 1, \cdots)$
u_{10}	$F(K_1, "gg")$	$\mathrm{Enc}(K_D, 3, \cdots)$
u_{11}	$F(K_1, "gt")$	$\mathrm{Enc}(K_D, 4, \cdots)$

4 安全性证明与效率分析

4.1 安全性证明

构建理想实验具体过程如下:

$\mathrm{Ideal}_{A,S}(1^\lambda)$:

$T \leftarrow L_1(A(s))$

$\mathrm{SI}' \leftarrow S^{L_1(f,T)}(\lambda)$

for $1 \leqslant i \leqslant j$

$L_2(s, p_1, \cdots, p_i) \xleftarrow[\text{each time}]{\text{one query}} A(\mathrm{SI}', p_i)$

$v_i \leftarrow L_2(s, p_1, \cdots, p_i)$

$L_2(s, p_1, \cdots, p_i) \xleftarrow[\text{each time}]{\text{one query}} A(\mathrm{SI}', p_i, v_i)$

$v_i \leftarrow L_2(s, p_1, \cdots, p_i)$

output $b' \leftarrow A(\mathrm{SI}', V_{\mathrm{Ideal}}(v_1, \cdots, v_j))$

根据安全性定义,预证本方法在泄露函数 L_1, L_2 下满足动态自适应安全,应对任意多项式敌手 A,存在一个多项式模拟器 S,满足 $|\Pr[\mathrm{Real}_{A,U}(1^\lambda) = 1] - \Pr[\mathrm{Ideal}_{A,S}(1^\lambda) = 1]| \leqslant \lambda$. 其中,$\lambda$ 为可忽略函数. 称基于后缀树的基因数据可搜索加密方法在泄露函数 L_1, L_2 下是满足动态自适应安全的.

证明：

1）在加密过程中，S 可以根据 L_1 构建模拟结构 D'，生成 $2n$ 条定长随机数据模拟 D. 对任意多项式时间敌手 A 满足

$$|\Pr[1\leftarrow \mathrm{A}(D,n)]-\Pr[1\leftarrow \mathrm{A}(D',n)]|\leqslant\lambda.$$

假设 λ 不可忽略，则敌手 A 可以区分 D 和 D'，即能以不可忽略概率区分随机数和伪随机函数，违背伪随机函数的伪随机性. 因此，概率 λ 可忽略.

2）在加密过程中，满足 IND－CPA 安全的真实加密方法 ε 生成密文数组 $C[i]=\varepsilon.\mathrm{Enc}(K_C, s_i\|i)$. S 构建游戏 $H_0,\cdots,H_n$：若 $i'>i$，$C_{i'}=\varepsilon.\mathrm{Enc}(K_C,(\sigma,0))$（$\sigma$ 是随机数），否则 $C_{i'}=\varepsilon.\mathrm{Enc}(K_C,(s_{i'},i'))$. 要求敌手 A 区分 C_i 是由 $(\sigma,0)$ 还是 $(s_{i'},i')$ 计算得到，则对于所有多项式时间敌手 A，存在不等式：

$$\left|\Pr\left[\begin{array}{l}[m_0=(s_i,i),m_1=(\sigma,0)];\\ b\xleftarrow{R}\{0,1\};\\ C=\varepsilon.\mathrm{Enc}(K_C,m_b);b'\leftarrow A(\varepsilon,\\ \mathrm{Enc}(K_C,\cdot)):b'=b\end{array}\right]-1/2\right|\leqslant\lambda.$$

假设 λ 不可忽略，表示敌手 A 可以以不可忽略概率区分 $\varepsilon.\mathrm{Enc}(K_C,(\sigma,0))$ 和 $\varepsilon.\mathrm{Enc}(K_C,s_i\|i)$，则打破 IND－CPA 安全，产生悖论. 因此，概率 λ 可忽略.

3）在搜索过程中，假设两次查询 $p_j,p_{j'}$ 和令牌 $T_{i,j},T_{i,j'}$，存在相同不匹配前缀 $p_j[1\cdots i]=p_{j'}[1\cdots i]$. S 构建状态表 Ψ_1 模拟字典 D. $\Psi_1=(i,j,k)$ 代表字典 D 的入口 k 是第 i 次搜索访问第 j 个节点. 定义 $p_{\max}\times(s_{\max+1})$ 个游戏 $H_{i,j}$，其中 $p_{\max}$ 为最大查询次数，$s_{\max}$ 为最长待搜索字符串长度，$i\in\{1,\cdots,s_{\max}\}$ 且 $j\in\{1,\cdots,p_{\max}\}$. 当 $H_{i,j}$ 满足 $p_j[1\cdots i]\neq p_{j'}[1\cdots i]$，随机选择 r，令 $T_{j',i}\leftarrow\varepsilon.\mathrm{Enc}(r,\Psi_1(p_{j'}[1\cdots i]))$. 若 p_j 为匹配前缀，则重选 r' 替换 $T_{j',i}$ 为 $T_{j,i}$；否则，根据 L_2 返回 $T_{j',i}=QP(s,p_1,\cdots p_j)[j']$，如果 $D(\Psi_1(p_j[1\cdots i]))$ 成立，令 $T_{j',i}=T_{j,i}$. 如果真实 ε 具有密钥不可区分性，则对任意多项式时间敌手 A 满足不等式：

$$\left|\begin{array}{l}\Pr\left[\begin{array}{l}K\leftarrow\varepsilon.\mathrm{Gen}(1^\lambda);K'\leftarrow\varepsilon.\mathrm{Gen}(1^\lambda);\\ A(\varepsilon.\mathrm{Enc}(K,\cdot),\varepsilon.\mathrm{Enc}(K',\cdot))=1\end{array}\right]\\ -\Pr[K\leftarrow\varepsilon.\mathrm{Gen}(1^\lambda);A(\varepsilon.\\ \mathrm{Enc}(K,\cdot),\varepsilon.\mathrm{Enc}(K,\cdot))=1]\end{array}\right|<\lambda.$$

假设 λ 是不可忽略的，敌手 A 能够以不可忽略概率区分 $T_{i,j},T_{i,j'}$ 是由匹配前缀还是由不匹配前缀产生的，即敌手 A 可以打破密钥不可区分性，产生悖论. 因此概率 λ 是可忽略的.

综合实验所述，可得 $|\Pr[\mathrm{Real}_{A,U|}(\lambda)=1]-\Pr[\mathrm{Ideal}_{A,S}(\lambda)=1]|\leqslant\lambda$，即真实实验 $\mathrm{Real}_{A,U}(\lambda)$ 与理想实验 $\mathrm{Ideal}_{A,S}(\lambda)$ 的输出是不可区分的. 因此，如果 ε 是满足密钥不可区分性以及 IND－CPA安全的对称加密算法，F 和 P 分别是伪随机函数和伪随机置换，则本文方法在泄漏函数 L_1,L_2 下是满足动态自适应安全的.

4.2　效率分析

本方法使用 UKK 算法进行后缀树快速构建，时间复杂度 $O(n)$. 调用 Openssl 库中的随机数生成算法构建密钥组 K，时间复杂度 $O(1)$. 利用 AES_CBC 加密算法以及 SHA－128 伪随机函数构建安全索引. 实验数据集来源为 NCBI 数据库[13].

1）加密阶段：安全索引构建耗时如图 1 所示，构建安全索引的耗时会随着基因长度 d 增加而增加. 在 d 较大时，其安全索引构建所需要添加的虚拟孩子节点和虚拟字典条目也会增多，总共需要 $O(d^2)$ 次伪随机函数计算、伪随机序列计算和加密计算. 由于基因数据存在稳定性，因此，对其进行一次索引构建即可支持后续的多次搜索，所以本次测试结果可以考虑应用于实际.

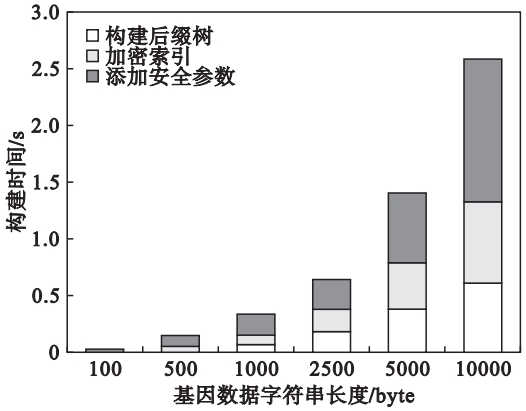

图 1　基因数据长度对安全索引构建耗时的影响

Fig. 1　Effect of genomic data length on secure index constructing time

2）搜索阶段：待搜索字符串长度对搜索时间的影响如图 2 所示. 待搜索字符串长度 m 影响构建搜索令牌数量，而令牌数量决定搜索字典时执行解密的次数，搜索 D 和 C 的复杂度分别为 $O(md)$ 和 $O(m)$ 个解密操作，L 的搜索次数保持为 $O(1)$. 执行一次搜索需要用户和服务器之间进行 3 次交互.

综上，根据基因数据的“一次构建，多次搜索”的特殊需求，本文提出的基于后缀树的基因数据可搜索加密方法适合现代医疗场景.

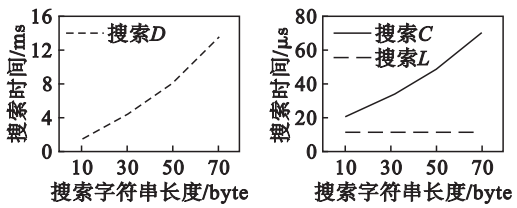

图 2 子字符串长度对搜索耗时的影响

Fig. 2 Effect of substring length on searching time

5 结 论

1) 针对多数密文搜索方案仅支持关键字搜索的问题,本文提出一种基于后缀树的基因数据可搜索加密方法. 在不可信环境下,对不具有关键字信息且重复性较大的基因数据的可搜索加密,实现基于密文的子字符串搜索并返回其具体位置信息.

2) 利用后缀树的结构特点,结合伪随机函数、哈希函数等密码学原语进行高效安全的方案构建,并给出相关形式化描述和安全性定义.

3) 安全性实验和性能实验结果表明,本文提出的基于后缀树的基因数据可搜索加密方法在个性化医疗逐渐大众化的环境下具备广阔的应用前景.

参考文献:

[1] Erlich Y, Narayanan A. Routes for breaching and protecting genetic privacy[J]. *Nature Reviews Genetics*, 2014, 15(6): 409 – 421.

[2] Cao N, Wang C, Li M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data[C]//IEEE Conference on Computer Communications. Piscataway: IEEE, 2011: 829 – 837.

[3] Wang B, Yu S, Lou W, et al. Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud[C]//IEEE Conference on Computer Communications. Piscataway: IEEE, 2014: 2112 – 2120.

[4] Lu W, Yamada Y, Sakuma J. Efficient secure outsourcing of genome-wide association studies[C]//2015 IEEE Security and Privacy Workshops. Los Alamitos: IEEE, 2015: 3 – 6.

[5] Chase M, Shen E. Substring searchable symmetric encryption [J]. *Proceedings on Privacy Enhancing Technologies*, 2015, 2015(2): 263 – 281.

[6] Ayday E, Raisaro J L, Hengartner U, et al. Privacy preserving processing of raw genomic data[C]//International Workshop on Data Privacy Management and Autonomous Spontaneous Security. New York: Springer-Verlag, 2013: 133 – 147.

[7] Riazi M S, Dantu N K R, Gattu L N V, et al. GenMatch: secure DNA compatibility testing[C]//IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). Piscataway: IEEE, 2016: 248 – 253.

[8] Simmons S, Berger B. Realizing privacy preserving genome-wide association studies[J]. *Bioinformatics*, 2016, 32(9): 1293 – 1300.

[9] Wang M, Ji Z, Wang S, et al. Mechanisms to protect the privacy of families when using the transmission disequilibrium test in genome-wide association studies[J]. *Bioinformatics*, 2017, 33(23): 3716 – 3725.

[10] Wang B, Song W, Lou W, et al. Privacy-preserving pattern matching over encrypted genetic data in cloud computing [C]//IEEE Conference on Computer Communications. Piscataway: IEEE, 2017: 1 – 9.

[11] Weiner P. Linear pattern matching algorithms[C]//14th Annual Symposium on Switching Automata Theory. New York: IEEE, 1973: 1 – 11.

[12] McCreight E M. A space-economical suffix tree construction algorithm[J]. *Journal of the ACM*, 1976, 23(2): 262 – 272.

[13] U. S. National Library of Medicine. National center for biotechnology information(NCBI): GenBank flat file release 229. 0[DB/OL]. [2018 – 11 – 15]. https://www.ncbi.nlm.nih.gov/genbank/.

(上接第 460 页)

[2] Govindana K, Fattahib M, Keyvanshokooh E. Supply chain network design under uncertainty: a comprehensive review and future research directions[J]. *European Journal of Operational Research*, 2017, 263(1): 108 – 141.

[3] Li T, Ma J H. Complexity analysis of the dual-channel supply chain model with delay decision[J]. *Nonlinear Dynamics*, 2014, 78(4): 2617 – 2626.

[4] Sivadasan S, Eftathiou J, Calinescu A, et al. Advances on measuring the operational complexity of supplier-customer systems[J]. *European Journal of Operational Research*, 2006, 171: 208 – 226.

[5] 邱菀华. 管理决策与应用熵学[M]. 北京: 机械工业出版社, 2002.
(Qiu Wan-hua. The decision-making of management and the application of entropy [M]. Beijing: China Machine Press, 2002.)

[6] 邱若臻,苑红涛,冯俏. 有限需求信息下基于最大熵原理的风险厌恶库存模型[J]. 东北大学学报(自然科学版), 2016, 37(10): 1512 – 1516.
(Qiu Ruo-zhen, Yuan Hong-tao, Feng Qiao. Risk aversion inventory model based on maximum entropy approach under limited demand information [J]. *Journal of Northeastern University(Natural Science)*, 2016, 37(10): 1512 – 1516.)

[7] Perea-López E, Grossmann I E, Ydstie B E, et al. Dynamic modeling and decentralized control of supply chains[J]. *Industrial Engineering Chemistry Research*, 2001, 40(15): 3369 – 3383.

[8] Raj T S, Lakshminarayanan S. Entropy-based optimization of decentralized supply-chain networks [J]. *Industrial Engineering Chemistry Research*, 2010, 49(7): 3250 – 3261.

[9] 李国家,汪定伟. 供应链中控制多级存储的射频辨识 Push/Pull 混合策略设计与仿真[J]. 控制理论与应用, 2014, 31(10): 1302 – 1309.
(Li Guo-jia, Wang Ding-wei. Design and simulation of radio frequency identification-enabled hybrid Push/Pull strategy for multi-echelon inventory of supply chain [J]. *Control Theory & Applications*, 2014, 31(10): 1302 – 1309.)

[10] Fu D F, Ionescu C M, Aghezzaf E H, et al. Decentralized and centralized model predictive control to reduce the Bullwhip effect in supply chain management[J]. *Computers & Industrial Engineering*, 2014, 73: 21 – 31.

[11] Xu H, Rong G, Feng Y P, et al. Control variance amplification in linear time invariant decentralized supply chains: a minimum variance control perspective[J]. *Industrial Engineering Chemistry Research*, 2010, 49(18): 8644 – 8656.

[12] Jung J Y, Blaua G, Pekny J F, et al. A simulation based optimization approach to supply chain management under demand uncertainty [J]. *Computers and Chemical Engineering*, 2004, 28(10): 2087 – 2106.

[13] Mele F D, Guillén G, Espuña A, et al. A simulation-based optimization framework for parameter optimization of supply-chain networks [J]. *Industrial Engineering Chemistry Research*, 2006, 45: 3133 – 3148.