4.2 Construction

在不损失一般性的情况下，我们在一个固定的字母表$\Sigma \in Z_q^*$上观察最大长度为 n 的向量，其他字母表可以映射到这样一个子集上。系统不要求关键字具有相同的长度，但是我们将用 0 填充不同长度的关键字直到均匀的长度n，这种填充将在计算中被安全地忽略【填充后面有举例说明】。完整的系统构造由如下四个算法$(Setup, Enc, KeyGen, Dec)$构成。

(1) $\boldsymbol{Setup(1^{\lambda}, n) \rightarrow (mpk, msk)}$: Setup 算法将安全参数$\lambda$和指定所需关键字长度的正整数$n$作为输入，并输出主公钥$mpk$和主私钥$msk$。首先运行$G(1^{\lambda})$，得到由$G_1$,$G_2$,$G_T$组成的非对称双线性群，群的阶数均为素数阶$q$，生成元$g_1 \in G_1$，$g_2 \in G_2$【使用 Type-A 曲线初始化】。

然后随机选择一个$v \overset{R}{\leftarrow} Z_q$并计算$h = g_1^v$，对于每个$i \in [2n]$均匀随机选择$s_i, t_i \overset{R}{\leftarrow} Z_q$并计算$h_i = g_1^{s_i} h^{t_i}$。

主公钥输出为$mpk = (G_1, G_2, G_T, q, e, g_1, g_2, h, (h_1, \ldots, h_{2n}))$

主私钥为$msk = (v, (s_1, \ldots, s_{2n}), (t_1, \ldots, t_{2n}))$

(2) $\boldsymbol{Enc(mpk, W) \rightarrow ct}$:加密算法以主公钥$mpk$和关键字$W$作为输入，输出密文$ct$。让$W = (w_1, \ldots, w_n) \in \Sigma^n$，其中$n$为关键字的长度，$w_i$为$W$的第$i$位。首先，数据提供者需要将表示关键字的向量的维数从一维扩展到二维。随机选取$r, r_1, \cdots, r_n \overset{R}{\leftarrow} Z_N$，构造向量$X = (x_1, \ldots, x_{2n})$，如下所示:

$$x_{2i-1} \coloneqq r \cdot r_i \cdot w_i \quad x_{2i} \coloneqq -r \cdot r_i$$

其中所有乘法都是模q运算。然后计算$C_1 = g_1^r, C_2 = h^r$，对于每个$i \in [2n]$，计算 $E_i = g_1^{x_i} \cdot h_i^r$。它将此关键字$W$的密文输出为$ct = (C_1, C_2, (E_1, \ldots, E_{2n}))$.

(3) $\boldsymbol{KeyGen(msk, \overline{W}) \rightarrow sk_{\overline{W}}}$:密钥生成算法以主密钥$msk$和包含要查询的通配符的关键字$\overline{W}$作为输入，输出一个功能密钥$sk_{\overline{W}}$。让$\overline{W} = (\overline{w}_1, \cdots, \overline{w}_n) \in \Sigma_*^n$，其中包含通配符或“不在乎”符号。同样的数据用户需要将关键字$\overline{W}$扩展到向量$Y = (y_1, \cdots, y_{2n})$，从一维扩展到二维，如下所示:

if $\overline{w}_i = \star$, $y_{2i-1} = y_{2i} = 0$;

if $\overline{w}_i \neq \star$, $y_{2i-1} = 1$, $y_{2i} = \overline{w}_i$.

选择一个随机数$m \overset{R}{\leftarrow} Z_q$并计算

$$\{K_i = g_2^{m \cdot y_i}\}_{i=1}^{2n},$$

$$T_1 = g_2^{m \cdot \sum_{i=1}^{2n} s_i \cdot y_i}, \qquad T_2 = g_2^{m \cdot \sum_{i=1}^{2n} t_i \cdot y_i}$$

然后输出搜索陷门的功能密钥，如$sk_{\overline{W}} = (T_1, T_2, (K_1, \ldots, K_{2n}))$.

(4) $\boldsymbol{Dec(ct, sk_{\overline{W}}) \rightarrow z}$: 云服务器运行的解密算法以主公钥$mpk$、密文$ct$和功能秘钥$sk_{\overline{W}}$作为输入，输出一个组元素$z \in G_T$，以下等式成立则$ct$和$sk_{\overline{W}}$匹配

$$\frac{\prod_{i=1}^{2n} e(E_i, K_i)}{e(C_1, T_1) \cdot e(C_2, T_2)} \overset{?}{\Leftrightarrow} 1$$

云服务器需要严格按照上述公式计算并输出。如果两个关键字向量 W 和$\overline{W}$在指数上的内积为 0，则最终结果为 1，此包含通配符的匹配成功。如果匹配成功，云服务器将向数据用户返回检索关键字对应的数据文件。否则，云服务器将返回一个失败标志。

备注：

1.等式匹配过程【原理，便于理解，这个步骤不用模拟】

$\prod_{i=1}^{2n} e(E_i, K_i) = \prod_{i=1}^{2n} e(g_1^{x_i} \cdot h_i^r, g_2^{m \cdot y_i}) = \prod_{i=1}^{2n} e(g_1^{x_i} \cdot (g_1^{s_i} \cdot g_1^{v \cdot t_i})^r, g_2^{m \cdot y_i})$

$= \prod_{i=1}^{2n} e(g_1^{x_i + r \cdot s_i + v \cdot r \cdot t_i}, g_2^{m \cdot y_i}) = \prod_{i=1}^{2n} e(g_1, g_2)^{m \cdot x_i \cdot y_i + r \cdot m \cdot s_i \cdot y_i + v \cdot r \cdot m \cdot t_i \cdot y_i}$

$= e(g_1, g_2)^{m \cdot <X,Y> + r \cdot m \cdot <S,Y> + v \cdot r \cdot m \cdot <T,Y>}$

$e(C_1, T_1) \cdot e(C_2, T_2) = e(g_1^r, g_2^{m \cdot \sum_{i=1}^{2n} s_i \cdot y_i}) \cdot e(h^r, g_2^{m \cdot \sum_{i=1}^{2n} t_i \cdot y_i})$

$= e(g_1^r, g_2^{m \cdot \sum_{i=1}^{2n} s_i \cdot y_i}) \cdot e(g_1^{vr}, g_2^{m \cdot \sum_{i=1}^{2n} t_i \cdot y_i})$

$= e(g_1, g_2)^{r \cdot m \cdot \sum_{i=1}^{2n} s_i \cdot y_i} \cdot e(g_1, g_2)^{v \cdot r \cdot m \cdot \sum_{i=1}^{2n} t_i \cdot y_i}$

$= e(g_1, g_2)^{r \cdot m \cdot <S,Y>} \cdot e(g_1, g_2)^{v \cdot r \cdot m \cdot <T,Y>}$

$\frac{\prod_{i=1}^{2n} e(E_i, K_i)}{e(C_1, T_1) \cdot e(C_2, T_2)} = \frac{e(g_1, g_2)^{m \cdot <X,Y> + r \cdot m \cdot <S,Y> + v \cdot r \cdot m \cdot <T,Y>}}{e(g_1, g_2)^{r \cdot m \cdot <S,Y>} \cdot e(g_1, g_2)^{v \cdot r \cdot m \cdot <T,Y>}} = e(g_1, g_2)^{m \cdot <X,Y>}$

2.关键词转换过程

假定一个关键字是 10byte，一般的单词很少超过 10 个字母吧

(word)=(01110111,01101111,01110010,01100100,剩下全是 0)

w 的 ASCII 码是 01110111

这里 W=(01110111,01101111,01110010,01100100,剩下全是 0), 按照上一页$\boldsymbol{Enc, KeyGen}$的加密方式加密

假定一个关键字是 20byte

(word)=(01110111,01101111,01110010,01100100,剩下全是 0)

假定不同关键字长度的原因是我们后期需要变换不同的关键字长度。