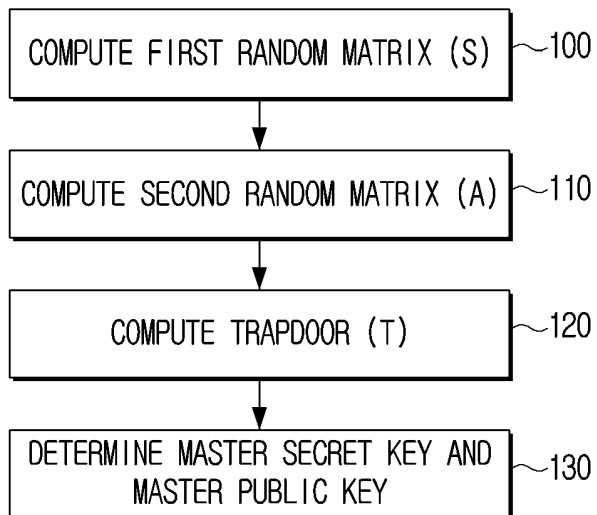
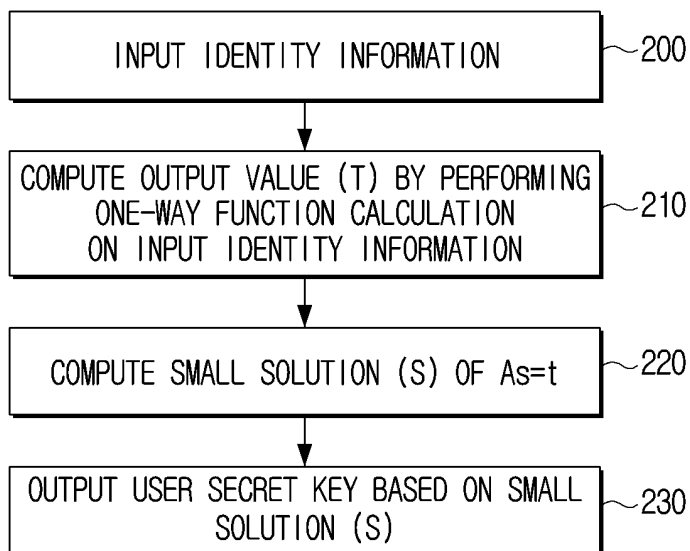


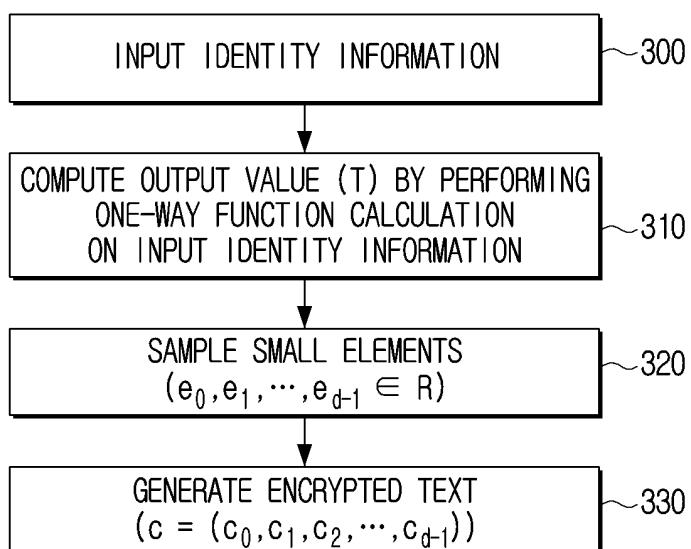
[Fig. 1]



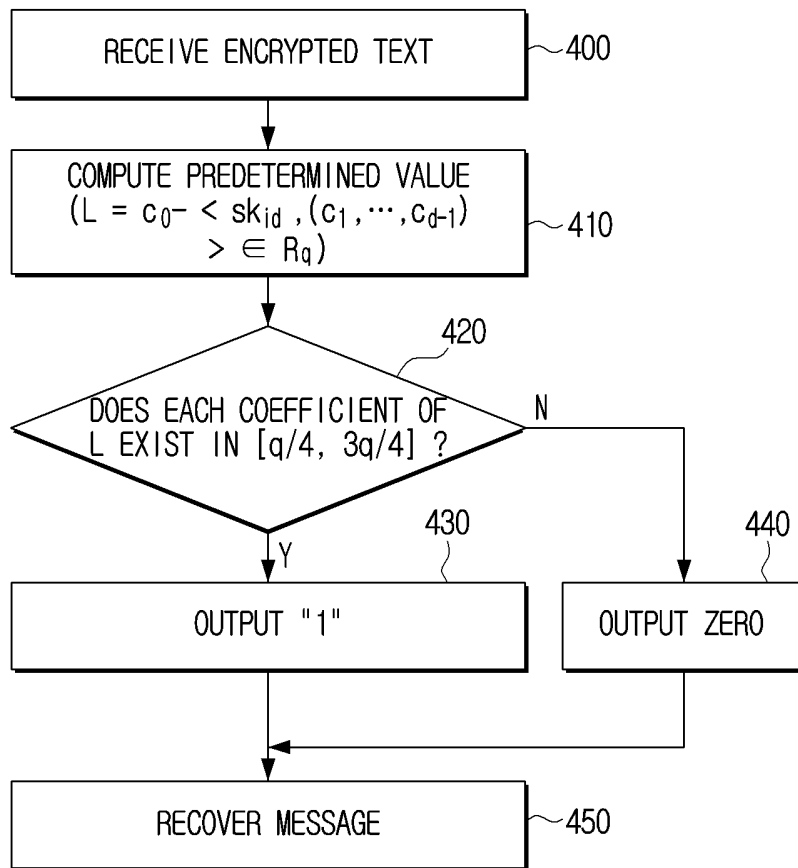
[Fig. 2]



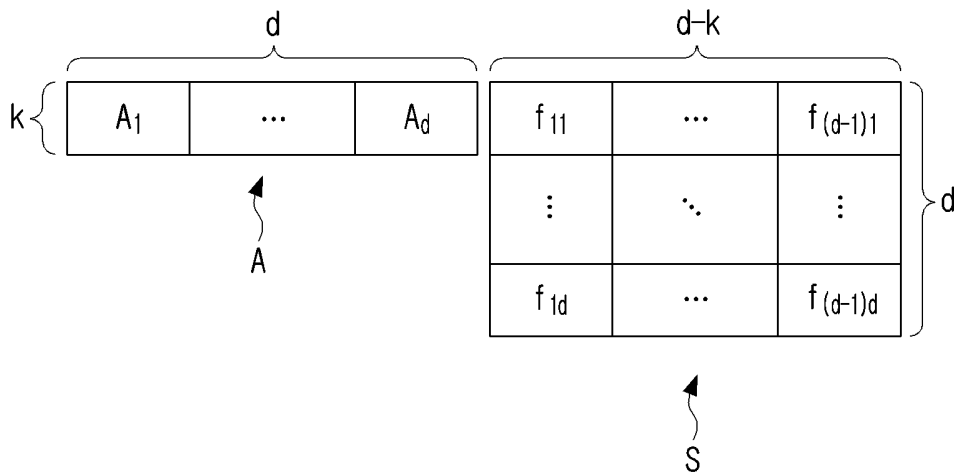
[Fig. 3]



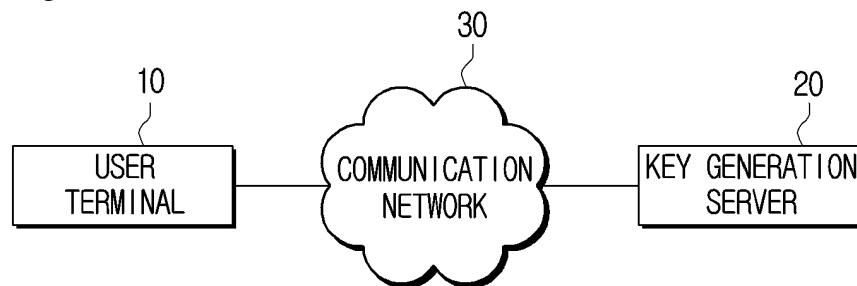
[Fig. 4]



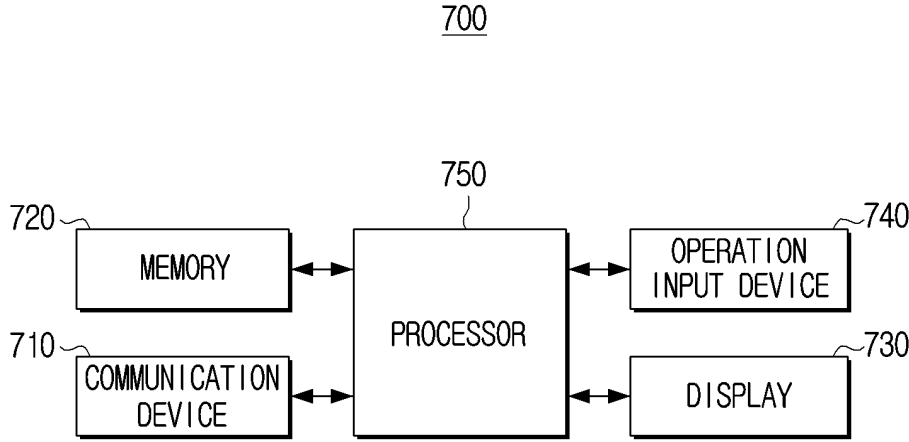
[Fig. 5]



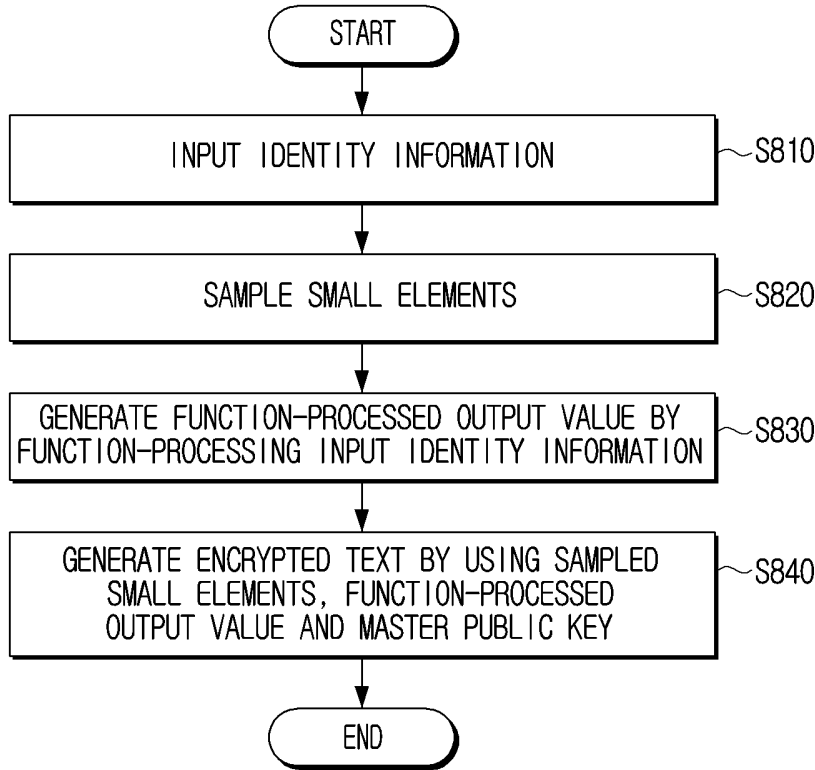
[Fig. 6]



[Fig. 7]



[Fig. 8]



[Fig. 9]

Algorithm 1 KeyGen

Input: n, q, d
Output: $\text{MPK} = \mathbf{h} \in R_q^{d-1}$ and $\text{MSK} = \mathbf{T}_{\text{MNTRU}} \in \mathbb{Z}^{dn \times dn}$

- 1: **for** $i = 1$ to $d - 1$ **do**
 - 2: $\sigma_i \leftarrow \sqrt{\frac{d}{d-i+1}} c_d \cdot q^{1/d} / \sqrt{dN}$ ▷ According to Section 3.2
 - 3: $\mathbf{f}_i \leftarrow (f_{1,i}, \dots, f_{d,i})$ where each coefficient of $f_{j,i} \in \mathcal{R}$ is sampled from $\mathcal{D}_{\mathbb{Z}, \sigma_i}$
 - 4: **end for**
 - 5: $\mathbf{S} \leftarrow [\mathbf{f}_1, \dots, \mathbf{f}_{d-1}]$
 - 6: $\det \leftarrow (\det_1, \dots, \det_d)$ where $\det_i = (-1)^{i-1} \cdot \det(\mathbf{S}_i)$
 - 7: $\mathbf{h} \leftarrow \det^{-1} \cdot (\det_2, \dots, \det_d) \in \mathcal{R}_q^{d-1}$
 - 8: Find a solution $\mathbf{F} = (F_1, \dots, F_d) \in \mathcal{R}^d$ of the MNTRU equation $\sum_{i=1}^d \det_i \cdot F_i = q$
 - 9: $\mathbf{T} \leftarrow [\mathcal{A}(\mathbf{S}) \parallel \mathcal{A}(\mathbf{F})]$
 - 10: **return** $\text{MPK} = \mathbf{h}$ and $\text{MSK} = \mathbf{T}$
-

[Fig. 10]

Algorithm 2 Extract

Input: An identity id , the master secret key $\mathbf{T}$, the master public key $\mathbf{h}$ and a hash function $H : \{0, 1\}^* \rightarrow \mathcal{R}_q$

Output: A user secret key $\mathbf{sk}_{id} \in \mathcal{R}^{d-1}$

- 1: **if** id is previously queried **then**
- 2: **return** $\mathbf{sk}_{id}$ in local storage
- 3: **else**
- 4: $\mathbf{t} \leftarrow (H(id), 0, \dots, 0) \in R_q^d$
- 5: $\sigma \leftarrow \frac{c_d}{\pi} \cdot \sqrt{\frac{\ln 2}{2} \left(\frac{\lambda}{2} + \log_2(4\sqrt{2} \cdot dn) \right)} \cdot q^{1/d}$
- 6: $\mathbf{c} \leftarrow \text{GaussianSampler}(\mathbf{T}, \sigma, \mathbf{t})$
- 7: $\mathbf{s} = (s_0, s_1, \dots, s_{d-1}) \leftarrow \mathbf{t} - \mathbf{c}$ $\triangleright \langle \mathbf{s}, (1, \mathbf{h}) \rangle = t$
- 8: Add $\mathbf{sk}_{id} = (s_1, \dots, s_{d-1})$ in local storage and **return** $\mathbf{sk}_{id}$
- 9: **end if**

[Fig. 11]

Algorithm 3 Encrypt

Input: An identity id , a message $\mu \in \{0, 1\}^m$, the master public key $\mathbf{h} \in R_q^{d-1}$, hash functions $H : \{0, 1\}^* \rightarrow R_q$ and $H' : \{0, 1\}^n \rightarrow \{0, 1\}^m$

Output: A ciphertext $C = (\mathbf{c}, c')$ where $\mathbf{c} \in R_q^d$ and $c' \in \{0, 1\}^m$.

- 1: $r, e_i \leftarrow \{-1, 0, 1\}^n$ for $0 \leq i \leq d-1$
- 2: $k \leftarrow \{0, 1\}^n$ $\triangleright k$ is an ephemeral key
- 3: $t \leftarrow H(id)$
- 4: $c_0 \leftarrow rt + e_0 + \lfloor \frac{q}{2} \rfloor \cdot k$
- 5: $c_0 \leftarrow 2^{\lceil \log_2 q \rceil - 3} \cdot \left\lfloor \frac{c_0}{2^{\lceil \log_2 q \rceil - 3}} \right\rfloor$ $\triangleright$ Store only 3 most significant bits of c_0
- 6: $\mathbf{c} \leftarrow (c_0, c_1, \dots, c_{d-1})$ where $c_i = rh_i + e_i$ for $1 \leq i \leq d-1$
- 7: $c' \leftarrow \mu \oplus H'(k)$
- 8: **return** $C = (\mathbf{c}, c')$

[Fig. 12]

Algorithm 4 Decrypt

Input: A ciphertext $C = (\mathbf{c}, c')$, a user secret key $\mathbf{sk}_{id} \in \mathcal{R}^{d-1}$, and hash functions $H : \{0, 1\}^* \rightarrow \mathcal{R}_q$ and $H' : \{0, 1\}^n \rightarrow \{0, 1\}^m$

Output: A message $\mu \in \{0, 1\}^m$

- 1: $\mathbf{s}' = (1, -\mathbf{sk}_{id})$
- 2: $w \leftarrow \langle \mathbf{c}, \mathbf{s}' \rangle$
- 3: $k \leftarrow \left\lfloor \frac{2}{q} \cdot w \right\rfloor$
- 4: **return** $m \leftarrow c \oplus H'(k)$