\begin{appendix}

\section{Hierarchical Inner-Product Encryption}

We use the definition proposed by Okamoto and Takashima [OT09.....]. We call a tuple of positive integers $\overrightarrow{\mu}:=(n,d;\mu\_{1},...,\mu\_{d})$ s.t $\mu\_{0}=0<\mu\_{1}<\mu\_{2}<...<\mu\_{d}=n$ a format of hierarchy of depth $d$ attribute spaces. Let $\sum\_{l}(l=1,..,d)$ be the sets of attributes, where each $\sum\_{l}:= \F^{\mu\_{l}-\mu\_{l-1}}\_{q} \backslash \{\overrightarrow{0}\}$. Let the hierarchical attributes $\sum:=\bigcup^{d}\_{l=1}(\sum\_{1} \times ...\times \sum\_{l})$, where the union is a disjoint union. Then, for $\overrightarrow{v}\_{i} \in \F^{\mu\_{l}-\mu\_{l-1}}\_{q} \backslash \{\overrightarrow{0}\}$, the hierarchical predicate $f\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}(\overrightarrow{x}\_{1},...,\overrightarrow{x}\_{h})=1$ iff $l\leq h$ and $\overrightarrow{x}\_{i}.\overrightarrow{v}\_{i}=0$ for all $i$ s.t. $a\leq i \leq l$.\

Let the space of hierarchical predicates $\mathcal{F}=:\{f\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}|\overrightarrow{v}\_{i} \in \F^{\mu\_{l}-\mu\_{l-1}}\_{q}\backslash \{\overrightarrow{0}\}\}$. We call $h$ (resp.$l$) the level of $(\overrightarrow{x}\_{1},...,\overrightarrow{x}\_{h})$ (resp.$(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{h})$.

\begin{definition}

Let $\overrightarrow{\mu}:=(n,d;\mu\_{1},...,\mu\_{d})$ s.t $\mu\_{0}=0<\mu\_{1}<\mu\_{2}<...<\mu\_{d}=n$ be a format of hierarchy of depth $d$ attribute spaces. A hierarchical predicate encryption (HPE) scheme for the class of hierarchical inner-product predicates $\mathcal{F}$ over the set of hierarchical attributes $\sum$ consists of probabilistic polynomial time algorithms $\setup, \keygen, \enc, \dec$ and $\delegate\_{l}$ for $l=1,...,d-1$. They are given as follows:

\begin{itemize}

\item $\setup$ takes as input security parameter $1^{\lambda}$ and format of hierarchy $\overrightarrow{\mu}$, and outputs master public key $\pp$ and master secret key $\msk$.

\item $\keygen$ takes as input the $\pp, \msk$, and predicate vectors $(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})$. It outputs a corresponding secret key $\sk\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}$.

\item $\enc$ takes as input the $\pp$, attribute vectors $(\overrightarrow{x}\_{1},...,\overrightarrow{x}\_{h})$, where $1\leq h \leq d$, and plaintext $m$ in some associated plaintext space, \textbf{msg}. It returns ciphertext $c$.

\item $\dec$ takes as input the master public key $\pp$, secret key $\sk\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}$, where $1\leq l \leq d$, and ciphertext $c$. It outputs either plaintext $m$ or the distinguished symbol $\perp$.

\item $\delegate\_{l}$ takes as input the master public key $\pp$, $l$-th level secret key $\sk\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}$, and $(l+1)$-th level predicate vector $\overrightarrow{v}\_{l+1}$. It returns $(l+1)$-th level secret key $\sk\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l+1})}$.

\end{itemize}

\end{definition}

\paragraph{Correctness.} For all correctly generated $\pp$ and $\sk\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}$, generate $c \xleftarrow{R} \enc(\pp, m, (\overrightarrow{x}\_{1},...,\overrightarrow{x}\_{h}))$ and $m^{'}= \dec(\pp, \sk\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}, c)$. If $f\_{(\overrightarrow{v}\_{1},...,\overrightarrow{v}\_{l})}(\overrightarrow{x}\_{1},...,\overrightarrow{x}\_{h})=1$, then $m^{'}=m$. Otherwise, $m^{'}\neq m$ except for negligible probability. For $f$ and $f^{'}$ in $\mathcal{F}$, we denote $f^{'}\leq f$ if the predicate vector for $f$ is the prefix of that for $f^{'}$.

\paragraph{Security.} A hierarchical inner-product predicate encryption scheme for hierarchical predicates $\mathcal{F}$ over hierarchical attributes $\sum$ is \textbf{selectively attribute-hiding against plaintext attacks} if for all probabilistic polynomial-time adversaries $\A$, the advantage of $\A$ in the following experiment is negligible in the security parameter.

\begin{enumerate}

\item $\A$ outputs challenge attribute vectors $X^{(0)}=(\overrightarrow{x}^{(0)}\_{1},....,\overrightarrow{x}^{(0)}\_{h}), X^{(1)}=(\overrightarrow{x}^{(1)}\_{1},....,\overrightarrow{x}^{(1)}\_{h})$.

\item $\setup$ is run to generate keys $\pp$ and $\msk$, and $\pp$ is given to $\A$.

\item $\A$ may adaptively makes a polynomial number of queries of the following type:

\begin{itemize}

\item $\A$ asks the challenger to create a secret key for a predicate $f \in \mathcal{F}$. The challenger creates a key for $f$ without giving it to $\A$.

\item $\A$ specifies a key for predicate $f$ that has already been created, and asks the challenger to perform a delegation operation to create a child key for $f^{'} \leq f$. The challenger computes the child key without giving it to the adversary.

\item $\A$ asks the challenger to reveal an already-created key for predicate $f$ s.t. $f(X^{(0)})=f(X^{(1)})=0$.

\end{itemize}

Note that when key creation requests are made, $\A$ does not automatically see the created key. $\A$ see a key only when it makes a reveal key query.

\item $\A$ outputs challenge plaintexts $m\_{0},m\_{1}$.

\item A random bit $b$ is chosen. $\A$ is given $c\xleftarrow{R} \enc(\pp, m\_{b}, X^{(b)})$.

\item The adversary may continue to request keys for additional predicate vectors subject to the restrictions given in step 3.

\item $\A$ outputs a bit $b^{'}$, and succeeds if $b^{'}=b$.

\end{enumerate}

We define the advantage of $\A$ as the quantity $\advantage^{HIPE}\_{\A}(\lambda)=|Pr[b^{'}=b]-1/2|$.

\subsection{Our HIPE Scheme}

We can apply the technical in our basic IPE scheme to optimize the hierarchical inner-product encryption scheme proposed by [Xagawa......]. For $i \in [d]$, we denote $s\_{i}$ to be the Gaussian parameter used in the $\keygen$ and $\delegate$ algorithm. Roughly speaking, we stack matrices $\textbf{B}\_{\vec{v}\_{i}}$ to make IPE hierarchical.\\[0.4cm]

$\setup(1^{\lambda}, n, q, m, \vec{\mu})$. On input a security parameter $\lambda$, and an hierarchical format of depth $d$ $\vec{\mu}=(l,d;\mu\_{1},...,\mu\_{d})$:

\begin{enumerate}

\item $(\textbf{A}, \mat{T}\_\mat{A})\leftarrow \trapgen(1^{\lambda}, q, n, m)$.

\item Chose a random matrix $\textbf{B}\xleftarrow{\$} \Z^{n \times m}\_{q}$ and a random vector $\vec{u}\xleftarrow{\$} \Z^{n}\_{q}$.

\end{enumerate}

~~~~~Output $\pp=(\textbf{A}, \textbf{B}, \vec{u})$ and $\msk= \mat{T}\_\mat{A}$.\\[0.4cm]

$\keygen(\pp, \msk, \overrightarrow{\textbf{V}})$: On input $\pp, \msk$ and predicate vectors $\overrightarrow{\textbf{V}}=(\vec{v}\_{1},...,\vec{v}\_{j})$ where $\vec{v}\_{i}=(v\_{i,1},...,v\_{i,l})$:

\begin{enumerate}

\item For all $i \in [j]$, define the matrix

\begin{equation}

\textbf{V}^{'}\_{i}= \begin{bmatrix}

v\_{i,1} \textbf{I}\_{n}\\

v\_{i,2} \textbf{I}\_{n}\\

\vdots\\

v\_{i,l} \textbf{I}\_{n}

\end{bmatrix} \in \Z^{ln \times n}\_{q},~~~ \textbf{V}\_{i}=\textbf{G}^{-1}\_{nl,l^{'},m}(\textbf{V}^{'}\_{i}.\textbf{G}\_{n,2,m}) \in [l^{'}]^{m \times m}.

\end{equation}

\item Set $\textbf{A}\_{\overrightarrow{\textbf{V}}}=[\textbf{A} | \textbf{B}\textbf{V}\_{1}| ...... | \textbf{B}\textbf{V}\_{j}] \in \Z^{n \times (j+1)m}\_{q}$.

\item Using the master secret key $\mat{T}\_\mat{A}$ to construct short basis $\textbf{E}\_{\overrightarrow{\textbf{V}}}$ for $\Lambda^{\perp}\_{q}(\textbf{A}\_{\overrightarrow{\textbf{V}}})$ by invoking the $\sampleleft$ algorithm.

\end{enumerate}

~~~~~Output $\sk\_{\overrightarrow{\textbf{V}}}=\textbf{E}\_{\overrightarrow{\textbf{V}}}$.\\[0.4cm]

$\enc(\pp, \overrightarrow{\textbf{W}}, m)$: On input $\pp$, a message $m \in \{0,1\}$, and attribute vectors $\overrightarrow{\textbf{W}}=(\vec{w}\_{1},...,\vec{w}\_{h})$ where $\vec{w}\_{i}=(w\_{i,1},...,w\_{i,l})$:

\begin{enumerate}

\item Choose a random vector $\vec{s} \xleftarrow{\$} \Z^{n}\_{q}$.

\item Set $\vec{c}\_{0}\leftarrow \vec{s}^{T}\textbf{A}+ \vec{e}^{T}\_{0}$, where $\vec{e}\_{0}\leftarrow \chi^{m}$.

\item Set $c\leftarrow \vec{s}^{T}\vec{u}+e+\lfloor q/2 \rfloor m$, where $e\leftarrow \chi$.

\item For all $i \in [h]$, define the matrix

\begin{equation}

\textbf{W}^{'}\_{i}= \begin{bmatrix}

w\_{i,1} \textbf{I}\_{n}&w\_{i,2} \textbf{I}\_{n}&\ldots&w\_{i,l} \textbf{I}\_{n}

\end{bmatrix} \in \Z^{n \times ln}\_{q},~~~ \textbf{W}\_{i}=\textbf{W}^{'}\_{i}.\textbf{G}\_{nl,l^{'},m} \in \Z^{n \times m}\_{q}.

\end{equation}

choose a random matrix $\textbf{R}\_{i} \xleftarrow{\$} \{-1,1\}^{m \times m}$ and set $\vec{c}\_{i} \leftarrow \vec{s}^{T}(\textbf{B}+ \textbf{W}^{'}\_{i})+ \vec{e}^{T}\_{0}\textbf{R}\_{i}$.

\end{enumerate}

~~~~~Output $\ct=(\vec{c}\_{0}, \{\vec{c}\_{i}\}, c)$.\\[0.4cm]

$\dec(\pp, \sk\_{\overrightarrow{\textbf{V}}}, \ct)$: On input $\pp$, a decryption key $\sk\_{\overrightarrow{\textbf{V}}}$ where $\overrightarrow{\textbf{V}}=(\vec{v}\_{1},...,\vec{v}\_{j})$, and a ciphertext $\ct=(\vec{c}\_{0}, \{\vec{c}\_{i}\}, c)$:

\begin{enumerate}

\item For $i \in [j]$, set $\vec{c}\_{\vec{v}\_{i}}=\vec{c}\_{i}.\textbf{V}\_{i}$.

\item Let $\vec{c} \leftarrow [\vec{c}\_{0},\vec{c}\_{1}...,\vec{c}\_{j}] \in \Z^{(j+1)m}\_{q}$.

\item Set $\tau\_{t}=\sigma\_{t}.\sqrt{(t+1)m}.\omega(\sqrt{(t+1)m})$, Then $\tau\_{t}\geq \| \widetilde{\textbf{E}\_{\overrightarrow{\textbf{V}}}} \|.\omega(\sqrt{(t+1)m})$.

\item Compute $\vec{x}\_{\overrightarrow{\textbf{V}}} \leftarrow \samplepre(\textbf{A}\_{\overrightarrow{\textbf{V}}}, \textbf{E}\_{\overrightarrow{\textbf{V}}}, \vec{u}, \tau\_{t})$.

\item Compute $z=c-\vec{c}.\vec{x}\_{\overrightarrow{\textbf{V}}} (mod q)$, and output $\lfloor (2/q)z \rceil \in \{0,1\}$.

\end{enumerate}

$\delegate(\pp, \sk\_{\overrightarrow{\textbf{V}}}, \textbf{V}^{'})$: On input $\pp$, a decryption key $\sk\_{\overrightarrow{\textbf{V}}}=\textbf{E}\_{\overrightarrow{\textbf{V}}}$, where $\overrightarrow{\textbf{V}}=(\vec{v}\_{1},...,\vec{v}\_{j})$, and $\overrightarrow{\textbf{V}}^{'}=(\vec{v}\_{1},...,\vec{v}\_{j},\vec{v}\_{j+1},...,\vec{v}\_{t})$, $\vec{v}\_{i}=(v\_{i,1},...,v\_{i,l})$. do:

\begin{enumerate}

\item For all $i \in [t]$, define the matrices

\begin{equation}

\textbf{V}^{'}\_{i}= \begin{bmatrix}

v\_{i,1} \textbf{I}\_{n}\\

v\_{i,2} \textbf{I}\_{n}\\

\vdots\\

v\_{i,l} \textbf{I}\_{n}

\end{bmatrix} \in \Z^{ln \times n}\_{q},~~~ \textbf{V}\_{i}=\textbf{G}^{-1}\_{nl,l^{'},m}(\textbf{V}^{'}\_{i}.\textbf{G}\_{n,2,m}) \in [l^{'}]^{m \times m}.

\end{equation}

\item Set $\textbf{A}\_{\overrightarrow{\textbf{V}}^{'}}=[\textbf{A} | \textbf{B}\textbf{V}\_{1}| ...... | \textbf{B}\textbf{V}\_{t}] \in \Z^{n \times (t+1)m}\_{q}$

\item Recall that the secret key $\textbf{E}\_{\overrightarrow{\textbf{V}}}$ is a short basis for $\Lambda^{\perp}\_{q}(\mat{A}\_{\overrightarrow{\textbf{V}}})$. Using it to construct a short basis for $\Lambda^{\perp}\_{q}(\textbf{A}\_{\overrightarrow{\textbf{V}}^{'}})$ by invoking

\begin{equation}

\textbf{E}\_{\overrightarrow{\textbf{V}}^{'}} \leftarrow \sampleleft ( \textbf{A}\_{\overrightarrow{\textbf{V}}}, [\mat{B}\textbf{V}\_{j+1}|...|\mat{B}\textbf{V}\_{t}], \sk\_{\textbf{V}}, \sigma\_{t}).

\end{equation}

\end{enumerate}

~~~~~Output $\sk\_{\overrightarrow{\textbf{V}}^{'}}=\textbf{E}\_{\overrightarrow{\textbf{V}}^{'}}$\\[0.4cm]

We omit the correctness and security proof of the scheme, since they are very similar to our basis IPE scheme and ones in [ADCM12......]. We also omit the detail of parameters.

\section{Fuzzy Identity-based Encryption}

In this section, we construct a FIBE scheme from our IPE scheme, we first introduce the definition and the security model of Fuzzy IBE.\\[0.4cm]

A Fuzzy Identity Based encryption scheme consists of the following four algorithms:

\begin{description}

\item $\fuzzy.\setup(\lambda, l)\rightarrow (\pp, \msk)$: The algorithm takes as input the security parameter $\lambda$ and the maximum length of identities $l$. It outputs the public parameters $\pp$, and the master secret key $\msk$.

\item $\fuzzy.\extract(\msk, \pp, id, k)$: This algorithm takes as input the master key $\msk$, the public parameters $\pp$, an identity $id$ and the threshold $k\leq l$. It outputs a decryption key $\sk\_{id}$.

\item $\fuzzy.\enc(\pp, m, id^{'}) \rightarrow \ct\_{id^{'}}$: The algorithm takes as input: a message bit $m$, an identity $id^{'}$, and the public parameters $\pp$. It outputs the ciphertext $\ct\_{id^{'}}$.

\item $\fuzzy.\dec(\pp, \ct\_{id^{'}}, \sk\_{id})\rightarrow m$: This algorithm takes as input the ciphertext $\ct\_{id^{'}}$, the decryption key $\sk\_{id}$ and the public parameters $\pp$. It outputs the message $m$ if $|id \bigcap id^{'}| \geq k$.

\end{description}

\paragraph{Security.} We follow the Selective-ID model of security of Fuzzy Identity Based Encryption as given by Sahai and Waters[SW......].

\begin{itemize}

\item \textbf{Target}: The adversary declares the challenge identity, $id^{\*}$, and he wishes to be challenged upon.

\item \textbf{Setup}: The challenger runs the Setup algorithm of Fuzzy-IBE and gives the public parameters to the adversary.

\item \textbf{Phase 1}: The adversary is allowed to issue queries for private keys for identities $id\_{j}$ of its choice, as long as $|id\_{j} \bigcap id^{\*}|< k; \forall j$

\item \textbf{Challenge}: The adversary submits a message to encrypt. the challenger encrypts the message with the challenge $id^{\*}$ and then flips a random coin $r$. If $r=1$, the ciphertext is given to the adversary, otherwise a random element of the ciphertext space is returned.

\item \textbf{Phase 2}: Phase 1 is repeated.

\item \textbf{Guess}: the adversary outputs a guess $r^{'}$ of $r$. the advantage of an adversary A in this game is defined as $|Pr[r^{'}=r]-1/2|$.

\end{itemize}

A Fuzzy Identity Based Encryption scheme is secure in the Selective-Set model of security if all polynomial time adversaries have at most a negligible advantage in the Selective-Set game.\\[0.4cm]

We introduce the embedding of exact threshold by Katz, Sahai, and Waters[KSW......].

\paragraph{Exact threshold}: For binary vector $\vec{x} \in \{0,1\}^{N}$, $H\_{w}(\vec{x}$ denotes the Hamming weight of $\vec{x}$. For binary vectors $\vec{a}, \vec{x} \in \{0,1\}^{n}$, the exact threshold predicate is denoted by $\mathcal{P}^{th}\_{=t}(\vec{a}, \vec{x})$ and output 1 if and only if $H\_{w}(\vec{a} \& \vec{x})=t$, where $\&$ denotes the logical conjunction. Suppose that $t<q$. Set $\mu=N+1$, $\vec{v}=(\vec{a}, 1) \in \Z^{\mu}\_{q}$, and $\vec{w}=(\vec{x}, -t) \in \Z^{\mu}\_{q}$. We have that $\langle \vec{v}, \vec{w} \rangle =0 $ if and only if $H\_{w}(\vec{a} \& \vec{x})=t$.

\subsection{Our FIBE scheme}

Now, we use our basic IPE scheme to construct a FIBE scheme. Let $\{0,1\}^{N}$ be a space of identities. The threshold predicate over $\{0,1\}^{N}$ is defined by $\mathcal{P}^{th}\_{\geq t}(\vec{a}, \vec{x})$ and output 1 if and only if $H\_{w}(\vec{a} \& \vec{x}) \geq t$.\

It's easy to see that the above predicate can be written as $\bigcup^{N}\_{i=t}\mathcal{P}^{th}\_{i=t}(\vec{a}, \vec{x})$. Hence, we can implement a FIBE scheme in a lazy way by repeating ciphertexts of an IPE scheme that supports the relations $\mathcal{P}^{th}\_{\geq t}$ for $i=t,...,N$.\\[0.4cm]

$\fuzzy.\setup(1^{\lambda}, 1^{N})$: On input a security parameter $\lambda$, and identity size $N$, do:

\begin{enumerate}

\item $(\textbf{A}, \mat{T}\_\mat{A})\leftarrow \trapgen(1^{\lambda}, q, n, m)$.

\item Chose a random matrix $\textbf{B}\xleftarrow{\$} \Z^{n \times m}\_{q}$ and a random vector $\vec{u}\xleftarrow{\$} \Z^{n}\_{q}$.

\end{enumerate}

~~~~~Output $\pp=(\textbf{A}, \textbf{B}, \vec{u})$ and $\msk= \mat{T}\_\mat{A}$.\\[0.4cm]

$\fuzzy.\extract(\pp, \msk, id, t)$: On input public parameters $\pp$, a master key $\msk$, an identity $id= (a\_{1},...,a\_{N}) \in \{0,1\}^{N}$ and threshold $t \leq N$, do:

\begin{enumerate}

\item Set vector $\vec{v}=(a\_{1},...,a\_{N},1) \in \Z^{\mu}\_{q}$.

\item Define the matrix

\begin{equation}

\textbf{V}^{'}= \begin{bmatrix}

a\_{1} \textbf{I}\_{n}\\

a\_{2} \textbf{I}\_{n}\\

\vdots\\

a\_{N} \textbf{I}\_{n}\\

\textbf{I}\_{n}

\end{bmatrix} \in \Z^{\mu n \times n}\_{q},~~~ \textbf{V}=\textbf{G}^{-1}\_{\mu n,l^{'},m}(\textbf{V}^{'}\_{i}.\textbf{G}\_{n,2,m}) \in [l^{'}]^{m \times m}.

\end{equation}

\item Define the matrix $\textbf{U}=\textbf{BV} \in \Z^{n \times m}\_{q}$, $\textbf{A}\_{id}=[\textbf{A}|\textbf{U}]$.

\item Using the master secret key $\msk=(\mat{T}\_\mat{A},\sigma)$, compute $\vec{r}\leftarrow$ $\sampleleft$ ($\textbf{A}, \mat{T}\_\mat{A},\textbf{U}, \textbf{u},\sigma$). Then $\vec{r}$ is a vector in $\Z^{2m}$ satisfying $\textbf{A}\_{id}.\vec{r}=\vec{u}$ (mod $q$).

\end{enumerate}

~~Output the secret key $\textbf{sk}\_{id}=\vec{r}$.\\[0.4cm]

$\fuzzy.\enc(\pp, id^{'}, m)$: On input public parameters $\pp$, an identity $id^{'}=(x\_{1},...,x\_{N}) \in \{0,1\}^{N}$, and a message $m \in \{0,1\}$, do:

\begin{enumerate}

\item Define a sequence of vectors $\vec{w}\_{i}=(x\_{1},...,x\_{N},-t-i+1) \in \Z^{\mu n \times n}\_{q}, i=1,...,N-t+1$.

\item Choose a uniformly random $\vec{s}\xleftarrow{\$} \Z^{n}\_{q}$.

\item Choose a noise vector $\vec{e}\_{0}\leftarrow D\_{\Z^{m}\_{q},\alpha}$ and a noise term $e\leftarrow D\_{\Z\_{q},\alpha}$.

\item Compute $\vec{c}\_{0}=\vec{s}^{T}\textbf{A}+\vec{e}^{T}\_{0}$.

\item For all $i \in [N-t+1]$ Define the matrix

\begin{equation}

\textbf{W}^{'}\_{i}= \begin{bmatrix}

x\_{1} \textbf{I}\_{n}&x\_{2} \textbf{I}\_{n}&\ldots&x\_{N} \textbf{I}\_{n}& -(t+i-1)\textbf{I}\_{n}

\end{bmatrix} \in \Z^{n \times \mu n}\_{q},~~~ \textbf{W}\_{i}=\textbf{W}^{'}\_{i}.\textbf{G}\_{\mu n,l^{'},m} \in \Z^{n \times m}\_{q}.

\end{equation}

Pick a sequence of random matrices $\textbf{R}\_{i}\xleftarrow{\$} \{-1,1\}^{m \times m}, i=1,...,N-t+1$, define error vectors $\vec{e}^{T}\_{i}=\vec{e}^{T}\_{0}\textbf{R}\_{i}$. Set

\begin{equation}

\vec{c}\_{i}=\vec{s}^{T}(\textbf{B}+\textbf{W}\_{i})+\vec{e}^{T}\_{i},~~~c=\vec{s}^{T}\vec{u}+e+\lfloor \frac{q}{2} \rfloor m.

\end{equation}

\end{enumerate}

~~Output ciphertext $\ct\_{id^{'}}=(\vec{c}\_{0},\{\vec{c}\_{i}\},c)$.\\[0.4cm]

$\fuzzy.\dec(\pp, \sk\_{id}, \ct\_{id^{'}})$: On input public parameters $\pp$, a decryption key $\sk\_{id}$, and a ciphertext $\ct\_{id^{'}}$, do:

\begin{enumerate}

\item Let $H\_{w}(id \& id^{'})$ denotes Hamming weight of the logical conjunction of $id$ and $id^{'}$. If $H\_{w}(id \& id^{'})=k<t$, output $\perp$. Otherwise, we parse the $\ct\_{id^{'}}=(\vec{c}\_{0},\{\vec{c}\_{i}\},c)$, and compute the matrix $\textbf{V}$ for $id$ as above.

\item Compute $\tilde{\vec{c}}\_{k-t+1}=\vec{c}\_{k-t+1}.\textbf{V}$, let $\vec{c}=[\vec{c}\_{0}|\tilde{\vec{c}}\_{k-t+1}]$.

\item Compute $z=c-\vec{c}.\vec{r}$ (mod $q$).

\end{enumerate}

~~Output $0$ if $|z|< q/4$ and 1 otherwise.

\end{appendix}