A Secure Data Aggregation Protocol in VANETs Based on Multi-key FHE

Bo Mi[1], Hongyang Pan[1], Darong Huang[1(✉)], Tiancheng Wei[1], and Xingfeng Wang[2]

[1] Institute of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China
drhuang@cqjtu.edu.cn
[2] College of Cybersecurity, Sichuan University, Chengdu, China

Abstract. For sake of data aggregation in VANETs, a protocol is devised based on multi-key fully homomorphic encryption (MFHE). In order to introduce practical properties, such as scalability, into the proposed protocol, a dynamic topology is utilized to structure the very-basic framework. To address the problem of dynamic changes with respect to floating nodes, linear secret sharing scheme is applied to multi-key fully homomorphic encryption with threshold decryption, and then the partial sharing decryption is proposed. Performance analysis illustrated that the proposed scheme is feasible and the complexity expands. Under the universal composability frame, the proposed protocol is also proved to be semantically secure.

Keywords: VANETs · Data aggregation · MFHE · GSW · Linear secret sharing · Threshold decryption

1 Introduction

With the expansion of the data and the increase in the accounting overhead, it is natural to store the clients' data and perform the expensive computation on the remote powerful "cloud" servers. Although the "cloud" can provide considerably many advantages in costs and functionality, how to protect the data privacy has become one of the most serious problems in the process.

Fully homomorphic encryption (FHE), which was first proposed by Gentry in 2009 [1], can perform arbitrary circuits on encrypted data as the plaintext. FHE was initially designed to only involve one user and one cloud. However, there are many scenarios including multiparty communication, such as multiuser to one core, which could carry out the FHE operation under different keys. Multi-key FHE (MFHE) [7, 8] is an interesting result derived from it.

Further, we can consider a more complex situation in VANETs. Considering a basic VANETs data aggregation protocol with clusters abstractly, VANETs can be divided into many clusters which consist of vehicle members. In the process of data aggregation, the vehicle members will broadcast their traffic data to complete aggregations. In traffic transportation, we need to achieve the safety aggregation of vehicle

X. Sun et al. (Eds.): ICAIS 2019, LNCS 11635, pp. 178–190, 2019.
https://doi.org/10.1007/978-3-030-24268-8_17

data [9, 13, 14] within a certain range to complete various functions, such as early warning, congestion control and so on (Fig. 1).

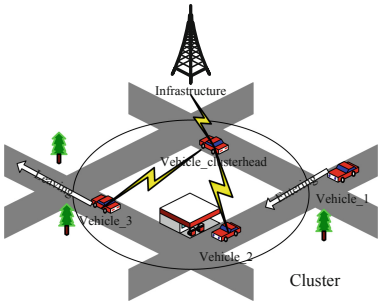

Fig. 1. VANETs

In order to achieve data aggregation privately, FHE could be used in the communication in VANETs with a set of natural and stringent requirements. First, we should protect the privacy information. Second, with the dynamic change of VANETs topology, we need to ensure the correctness of data aggregation. And some MFHE schemes make these requirements true partially. But the dynamic change also put up some new requirements for MFHE. For example, with the increase or decrease of the node number, an MFHE scheme should achieve a multi-hop homomorphic encryption.

Based on MFHE and threshold decryption, we present a secure 3-round protocol of data aggregation in VANETs. We applied an MFHE scheme based on GSW [8] and a Two-Round MPC protocol [16] to complete the data aggregation. And then in the dynamic situation, the linear secret sharing scheme has been used to cut apart the secret key and store the sharing separately on the other nodes. With the reduction of vehicle nodes, the sharing of the node left from the cluster will be reconstructed from the other nodes which are still in it to finish the decryption of the final ciphertext.

1.1 Our Results and Techniques

In order to achieve the data aggregation in VANETs, we make some changes:

Based on the threshold encryption, the linear secret sharing has been applied to realize the variant partial decryption. By the linear secret sharing, a secret key will be split into the other nodes. Then we can reconstruct the variant partial decryption of the secret key using one-round communication. And then we can complete the threshold decryption.

To construct the 3-round data aggregation protocol in VANETs, we applied the Two-Round MPC protocol based on MFHE to complete the basis communication, and the variant partial decryption to ensure the reliability in the dynamic situation. In the dynamic situation of VANETs, each secret key will be split into some sharing for the other nodes. If one vehicle leaves the cluster, the variant partial decryption will be executed for this node to complete the final decryption.

1.2 Other Related Work

The basic idea of performing the evaluation between the ciphertexts encrypted by different keys using the homomorphic encryption schemes was first proposed by López-Alt, Tromer and Vaikuntanathan [7]. Their protocol, however, was built on the NTRU scheme which relied on a non-standard assumption, referred to as the Decisional Small Polynomial Ratio assumption. Clear and McGoldrick [8], on the basis of GSW IBFHE schemes [6] and GPV IBE schemes [10], constructed a new approach to achieve the multi-identity IBFHE. Coincidentally, based on the standard LWE assumption, the approach implements the multi-key FHE of [11]. Based on the Clear and McGoldrick's multi-key FHE scheme, Mukherjee and Wichs [16] proposed a two-round MPC protocol.

1.3 Organization

In Sect. 2, we introduce the notation used in this paper and the related definition of the MFHE with threshold decryption. In Sect. 3, we show the threshold decryption and the variant partial decryption. In Sect. 4, we show how to construct the data aggregation and analyze the security and performance.

2 Preliminary

Notations. Throughout, we let λ denote the security parameter and $\mathrm{negl}(\lambda)$ denote a negligible function. We represent elements in $\mathbb{Z}_q$ as integers in the range $(-q/2, q/2]$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$ be a vector. We use the notation $\mathbf{x}[i]$ to denote the i-th component scalar. Similarly, for a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times m}$, we use $\mathbf{M}[i, j]$ to denote the scalar element located in the i-th row and the j-th column. And, for an integer $x \in \mathbb{Z}_q$, we use $x[i]$ to denote the i-th bit. The infinity norm of a vector $\mathbf{x}$ is defined as $\|\mathbf{x}\|_\infty = \max_i(|\mathbf{x}[i]|)$. The norm of matrices is defined similarly.

2.1 Multi-key FHE with Threshold Decryption

We start with the definition of Threshold multi-key FHE which has been proposed in [16].

Definition 2.1. Threshold multi-key FHE scheme (TMFHE) *is a multi-key FHE scheme with two additional algorithms* PartDec, FinDec *described as follow:*

- $p_i \leftarrow \mathrm{PartDec}(\hat{c}, (pk_1, \ldots, pk_N), i, sk_i)$*: On input an expanded ciphertext under a sequence of N keys and the i-th secret key output a partial decryption* p_i.
- $\mu \leftarrow \mathrm{FinDec}(p_1, \ldots, p_N)$*: On input N partial decryption output the plaintext* μ.

Now we propose our definition for the variant partial decryption. 3 algorithms have been inserted into the new definition as follow:

Definition 2.3. Threshold multi-key FHE scheme* (TMFHE*) *is a threshold multi-key FHE scheme with three additional algorithms* SecSplit, SharPartDec, SharFinDec *described as follow:*

- $\{s_j\}_{j\in[N]\setminus\{i\}} \leftarrow \text{SecSplit}(N, i, sk_i)$*: On input a secret key* sk_i *and the number of parties N output* $N-1$ *sharing.*
- $sp_j \leftarrow \text{SharPartDec}(\hat{c}, (pk_1, \ldots, pk_N), j, s_j)$*: On input an expanded ciphertext under a sequence of N keys and the j-th sharing* s_j *output a partial sharing decryption* sp_j*.*
- $p_i' \leftarrow \text{SharFinDec}(sp_1, \ldots, sp_N)$*: On input N partial sharing decryptions output the partial decryption* p_i'*.*

This definition requires correctness and security as follow:

Simulator Security. *There exists a PPT simulator* $\mathcal{S}^{vthr}$ *which, on input the index* $j \in [N]$ *and all but the i-th sharing* $\{s_j\}_{j\in[N]\setminus\{i\}}$*, the evaluated ciphertext* $\hat{c}$ *and the k-th secret key* sk_k *produces a simulated partial sharing decryption* $sp_i' \leftarrow \mathcal{S}^{vthr}(sk_k, \hat{c}, i, \{s_j\}_{j\in[N]\setminus\{i,k\}})$ *such that:*

$$sp_i' \overset{comp}{\approx} sp_i$$

where $sp_i \leftarrow \text{SharPartDec}(\hat{c}, (pk_1, \ldots, pk_N), i, s_i)$.

Correctness. *The following holds with probability 1:*

$$\text{FinDec}(p_1, \ldots, p_i', \ldots, p_N) = \mu$$

where $p_i' \leftarrow \text{SharFinDec}(sp_1, \ldots, sp_N)$.

2.2 Other Related Definitions

Now we give some related definitions which would be used in the rest of this paper.

Definition 2.4 (B-Bounded Distribution). *A distribution ensemble* χ*, supported over the integers, is called B-bounded if*

$$\Pr_{e\leftarrow\chi}[|e| > B] \leq \text{negl}(\lambda).$$

Definition 2.5 (Statistical Indistinguishability). *For two distribution ensembles* X, Y*, over a finite domain* $\mathbf{\Omega}$*.* X, Y *is statistical indistinguishable, denoted by* $\text{X} \overset{stat}{\approx} \text{Y}$*, if*

$$\Delta(\text{X}, \text{Y}) \leq \text{negl}(n).$$

where $\Delta(\text{X},\text{Y}) \overset{def}{=} \frac{1}{2}\sum_{\omega\in\Omega} |\text{X}(\omega) - \text{Y}(\omega)|$.

3 Threshold Decryption via Linear Secret Sharing

We now show how to construct the variant threshold decryption from MFHE by linear secret sharing. It proceeds in 3 parts, which is shown as follow:

1. We show how to perform the threshold decryption and the variant based on linear secret sharing for this scheme.
2. We show the correctness and security of the variant threshold decryption.

3.1 Variant of Threshold Decryption Based on Linear Secret Sharing

This part is to implement the variant threshold decryption for the MFHE construction and its reconstruction on the sharing of one's secret key.

The threshold decryption is implemented by the following 2 functions **PartDec**(…) and **FinDec**(…):

- **PartDec**$(\hat{c}, i, sk_i)$: On input an expanded ciphertext $\hat{c} = \hat{\mathbf{C}} \in \mathbb{Z}_q^{nN \times mN}$ in [8] and the i-th secret key $sk_i = \mathbf{t}_i \in \mathbb{Z}_q^n$ do the following:
  1. Parse $\hat{\mathbf{C}}$ as consisting of N sub-matrices $\hat{\mathbf{C}}^{(i)} \in \mathbb{Z}_q^{n \times mN}$ such that $\hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{C}}^{(1)} \\ \vdots \\ \hat{\mathbf{C}}^{(N)} \end{bmatrix}$
  2. Define $\hat{\mathbf{w}} \in \mathbb{Z}_q^{nN}$ as $\hat{\mathbf{w}} = [0, \ldots, 0, \lceil q/2 \rceil]$.
  3. Then compute $\gamma_i = \mathbf{t}_i \hat{\mathbf{C}}^{(i)} \hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}) \in \mathbb{Z}_q$ and output $p_i = \gamma_i + e_i^{sm} \in \mathbb{Z}_q$ where $e_i^{sm} \xleftarrow{\$} \left[-B_{smdg}^{dec}, B_{smdg}^{dec}\right]$ is a random noise where $B_{smdg}^{dec} = B_\chi 2^{d\lambda \log \lambda}$.
- **FinDec**$(p_1, \ldots, p_N)$: Given $p_1, \ldots, p_N$, compute the sum $p = \sum_{i=1}^{N} p_i$. Output $\mu := \left| Round\left(\frac{p}{q/2}\right) \right|$.

As mentioned in the Sect. 1, the increase and decrease of the node number will affect the encryption and decryption of the MFHE scheme in the multi-hop environment. When it increases, we can make evaluation ciphertexts expanded in the next hop. And if a node leaves, we make use of linear secret sharing scheme to solve it. A new parameter r will be set as $r = r(\lambda, d)$. The variant based on the linear secret sharing consists of the following 3 algorithms:

- **SecSplit**(N, i, sk_i): On input a secret key sk_i, parse $sk_i = \mathbf{t}_i = \left[t_{i,1}, t_{i,2}, \ldots, t_{i,n}\right] \in \mathbb{Z}_q^n$. For $j \in [n]$ compute the sharing of $t_{i,j}$ as follow:
  1. Sample 2 vectors $\mathbf{x}_j = [x_{j,1}, \ldots, x_{j,i-1}, x_{j,i+1}, \ldots, x_{j,N}] \xleftarrow{\$} \mathbb{Z}_q^{N-1}$ and $\mathbf{k}_j = [k_{j,1}, \ldots, k_{j,N-2}] \xleftarrow{\$} \mathbb{Z}_q^{N-2}$ for $\forall k_1 \neq k_2 \in [N] \backslash \{i\}$, $\left|x_{j,k_1} - x_{j,k_2}\right| \geq r$.
  2. Compute the vector $\mathbf{y}_j \in \mathbb{Z}_q^{N-1}$ as follow:

$$\begin{aligned}\mathbf{y}_j =& [(1^{N-1})^T, \mathbf{x}_j^T, (\mathbf{x}_j^2)^T, \ldots, (\mathbf{x}_j^{N-2})^T] \cdot \begin{bmatrix} t_{i,j} \\ \mathbf{k}_j^T \end{bmatrix} \\ =& [y_{j,1}, \ldots, y_{j,N-1}] \in \mathbb{Z}_q^{N-1}\end{aligned}$$

and the sharing is output as follow:

$$\begin{bmatrix} (x_{1,1}, y_{1,1}) & \cdots & \cdots & (x_{n,1}, y_{n,1}) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ (x_{1,N}, y_{1,N}) & \cdots & \cdots & (x_{n,N}, y_{n,N}) \end{bmatrix}$$

These tuples in the same row are the sharing received by the same party, and the tuples in the same column are the sharing split by the same value.

- **SharPartDec**$((x_{i,j}, y_{i,j}), \hat{c}, i, k, N)$: On input a sharing tuple $(x_{i,j}, y_{i,j})$, the expanded ciphertext $\hat{c} = \hat{\mathbf{C}} \in \mathbb{Z}_q^{nN \times mN}$, the index k of the secret key $\mathbf{t}_k$ and the index i of the i-th component scalar $t_{k,i}$, execute the following steps:

  1. Parse $\hat{\mathbf{C}}$ as consisting of $n \times N$ vectors $\hat{\mathbf{c}}^{(i)} \in \mathbb{Z}_q^{mN}$ such that $\hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{c}}^{(1)} \\ \vdots \\ \hat{\mathbf{c}}^{(nN)} \end{bmatrix}$
  2. Difine $\hat{\mathbf{w}} \in \mathbb{Z}_q^{nN}$ as $\hat{\mathbf{w}} = [0, \ldots, 0, \lceil q/2 \rceil]$.
  3. Then compute the partial sharing decryption $(\upsilon_{i,j}, \tau_{i,j})$ as follow:

$$\upsilon_{i,j} = x_{i,j}\hat{\mathbf{c}}^{(kn+i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}) + e_i^{smx} \in \mathbb{Z}_q$$

$$\tau_{i,j} = y_{i,j}\hat{\mathbf{c}}^{(kn+i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}) + e_i^{smy} \in \mathbb{Z}_q$$

  where $e_i^{smx}, e_i^{smy} \xleftarrow{\$} \left[-B_{smdg}^{vdec}, B_{smdg}^{vdec}\right]$ is a random noise where $B_{smdg}^{vdec} = 2^{d\lambda \log \lambda}$.

- **SharFinDec**$((\upsilon_{i,j}, \tau_{i,j})_{i\in[n], j\in[N]\setminus\{k\}})$: Given all the partial sharing decryptions $(\upsilon_{i,j}, \tau_{i,j})_{i\in[n], k\in[N]\setminus\{k\}}$, compute the variant partial decryption as follow:

$$p_j' = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq k}}^{N} \tau_{i,j} \prod_{\substack{h=1 \\ h\neq j \\ h\neq k}}^{N} \upsilon_{i,h}/(\upsilon_{i,h} - \upsilon_{i,j})$$

  and then output p_j'.

3.2 Correctness and Simulation Security

Now, we testify the correctness along with security of our partial sharing decryption

Theorem 3.1. *The above variant procedures of threshold decryption for* MFHE *satisfy correctness and simulation security.*

Correctness. Here the entire scheme is same as MFHE except the variant of threshold decryption based on linear secret sharing. If $(\upsilon_{i,h}, \tau_{i,h})$ and $(\upsilon_{i,j}, \tau_{i,j})$ are the partial sharing decryption of a secret key $\mathbf{t}_k$, then we have

$$\begin{aligned}\frac{\upsilon_{i,h}}{\upsilon_{i,h}-\upsilon_{i,j}} &= \frac{x_{i,h}\hat{\mathbf{c}}^{(kn+i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e_h}{(x_{i,h}-x_{i,j})\hat{\mathbf{c}}^{(kn+i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+(e_h-e_j)} \\ &= \frac{x_{i,h}}{x_{i,h}-x_{i,j}}\cdot\frac{\hat{\mathbf{c}}^{(kn+i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e'}{\hat{\mathbf{c}}^{(kn+i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e''}\end{aligned}$$

where $e' = e_h/x_{i,h}(x_{i,h}-x_{i,j})$, $e'' = (e_h-e_j)/(x_{i,h}-x_{i,j})$. The equation can be generalized into the following form:

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j=1}^{N}\tau_{i,j}\prod_{\substack{h=1\\h\neq j\\h\neq k}}^{N}\upsilon_{i,h}/(\upsilon_{i,h}-\upsilon_{i,j}) \\ &= \sum_{i=1}^{n}\frac{(\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e')^{N-1}}{(\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e'')^{N-2}}\cdot t_{k,i} \\ &= \left(\frac{\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e'}{\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e''}\right)^{N-2}(\mathbf{t}_k\hat{\mathbf{C}}^{(k)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+\mathbf{t}_k\mathbf{e}')\end{aligned}$$

where $\hat{\mathbf{c}}$ is a row vector of $\hat{\mathbf{C}}$. It is easy to see that $\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})$ is much larger than e' and e'', and the value of $\left(\frac{\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e'}{\hat{\mathbf{c}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+e''}\right)^{N-2}$ is very close to 1. So the correctness is primarily determined by $\mathbf{t}_k\hat{\mathbf{C}}^{(k)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}})+\mathbf{t}_k\mathbf{e}'$.

If $\hat{\mathbf{C}}$ is an evaluated ciphertext encrypting a bit μ and the secret key is $\hat{\mathbf{t}} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]$, then we have $\hat{\mathbf{t}}\hat{\mathbf{C}}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}^T) = \mu(q/2)+e$. Now, one can observe that decryption without threshold decryption works correctly as long as $\|e\|_\infty \leq q/4$.

If the threshold decryption with partial sharing decryption is executed, the final result must be correctly decrypted by the function **FinDec**(…). So we take $\mathbf{t}_k$'s variant partial decryption and the other partial decryption as input. And we have

$$\sum\nolimits_i(\mathbf{t}_i\hat{\mathbf{C}}^{(i)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}))+\mathbf{t}_k\mathbf{e}'+e^{sm} = \mu(q/2)+e+\mathbf{t}_k\mathbf{e}'+e^{sm}$$

Lemma 3.2. *Let* $\hat{\mathbf{C}}$ *be the evaluated ciphertext of the above* **MFHE** *scheme and e be the decryption noisy after a homomorphic evaluation of a d-level circuit* $\mathcal{C}$ *. The noisy e has norm upper bound* $B_\chi 2^{O(d\log\lambda)}$.

Proof. We refer the reader to [8] for details.

Lemma 3.3. *Let p be the final decryption of the above* **Threshold Decryption** *scheme generated by function* **FinDec(…)**, *and* e^{sm} *be the "smudging noisy" of p. The noisy* e^{sm} *has norm upper bound* $B_\chi 2^{O(d\lambda \log \lambda)}$.

Proof. We refer the reader to [16] for details.

Lemma 3.4. *Let* p'_k *be the final result of the above* **Variant Partial Decryption** *scheme and* $\mathbf{t}_i\mathbf{e}'$ *be the "variant smudging" noisy. The noisy* $\mathbf{t}_k\mathbf{e}'$ *has norm upper bound* $B_\chi 2^{O(d\lambda \log \lambda)}$.

Proof. Let $\mathbf{t}_i\mathbf{e}'$ be the "variant smudging" noisy. Recall that, $\mathbf{t}_i = [-\mathbf{s}_i, 1]$ with $\mathbf{s}_i \leftarrow \chi^{n-1}$, and $\mathbf{e}' = [e'_1, \ldots, e'_N]$. And for any $i \in [n]$, $e'_i \leq \frac{2^{d\lambda \log \lambda}}{r}$. Therefore, we have $\mathbf{t}_i\mathbf{e}' \leq \frac{nB_\chi 2^{d\lambda \log \lambda}}{r} = B_\chi 2^{O(d\lambda \log \lambda)}$.

So e has norm $|e| \leq B_\chi 2^{O(d \log \lambda)}$, $\mathbf{t}_i\mathbf{e}'$ has norm $|\mathbf{t}_i\mathbf{e}'| \leq B_\chi 2^{O(d\lambda \log \lambda)}$ and e^{sm} has norm $|e^{sm}| \leq B_\chi 2^{O(d\lambda \log \lambda)}$. Since $q = B_\chi 2^{\omega(d\lambda \log \lambda)}$, we have $|e + \mathbf{t}_i\mathbf{e}' + e^{sm}| \leq q/4$ and correctness holds.

Security. We construct the simulator $\mathcal{S}^{vthr}$ as below:

On input sharing $(x_{u,j}, y_{u,j})_{u\in[n], j\in[N]\setminus\{i,k\}}$, an evaluated ciphertext $\hat{c}$ and the secret key $\mathbf{t}_k$ generating secret sharing $(x_{u,j}, y_{u,j})$, outputs the *simulated partial sharing decryption* as the below steps:

1. Construct n matrices $\{\mathbf{MX}_u = [\mathbf{x}_u, \mathbf{x}_u^2, \ldots, \mathbf{x}_u^{N-2}] \in \mathbb{Z}_q^{N-2\times N-2}\}_{u\in[n]}$ and n vectors $\{\mathbf{Vy}_u = [\mathbf{y}_u - \mathbf{t}_{k,u}] \in \mathbb{Z}_q^{N-2}\}_{u\in[n]}$ where $\mathbf{x}_u = [\ldots, x_{u,j}, \ldots]^T_{j\in[N]\setminus\{i,k\}} \in \mathbb{Z}_q^{N-2}$, $\mathbf{y}_u = [\ldots, y_{u,j}, \ldots]^T_{j\in[N]\setminus\{i,k\}} \in \mathbb{Z}_q^{N-2}$ and $\mathbf{t}_{k,u} = [t_{k,u}, \ldots, t_{k,u}]^T \in \mathbb{Z}_q^{N-2}$. And then compute n vectors $\{\mathbf{k}_u = (\mathbf{MX}_u)^{-1} \cdot \mathbf{Vy}_u \in \mathbb{Z}_q^{N-2}\}_{u\in[n]}$.
2. Sample a vector $\mathbf{Sx}_i = [x'_{1,i}, \ldots, x'_{n,i}]^T \xleftarrow{\$} \mathbb{Z}_q^n$ and for each $u \in [n]$ compute $y'_{u,i} = [1^n, x'_{u,i}, (x'_{u,i})^2, \ldots, (x'_{u,i})^{N-2}] \cdot \begin{bmatrix} t_{k,u} \\ \mathbf{k}_u \end{bmatrix}$. And we have $\mathbf{Sy}_i = [y'_{1,i}, \ldots, y'_{n,i}]^T$.
3. For each $u \in [n]$ compute the u-th simulated partial sharing decryption:

$$\upsilon'_{u,i} = x'_{u,i}\hat{\mathbf{c}}^{(kn+u)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}) + e_u^{smx} \in \mathbb{Z}_q, \tau'_{u,i} = y'_{u,i}\hat{\mathbf{c}}^{(kn+u)}\hat{\mathbf{G}}^{-1}(\hat{\mathbf{w}}) + e_u^{smy} \in \mathbb{Z}_q$$

where $e_u^{smx}, e_u^{smy} \xleftarrow{\$} \left[-B_{smdg}^{vdec}, B_{smdg}^{vdec}\right]$. Then output the simulated partial sharing decryption $sp'_i = \{(\upsilon'_{u,i}, \tau'_{u,i})_{u\in[n]}\}$.

The real value sp_i and the simulated sp'_i are almost statistically indistinguishable.

4 Data Aggregation Protocol in VANETs

In this section, we now describe our secure aggregation protocol in VANETs within the cluster through 3 rounds of communication. The following two procedures are supplemented in [16].

Increase. When a new vehicle participants in the cluster, the next hop computation should be executed within $N + 1$ nodes after the final decryption of the last hop.

Decrease. When a vehicle in the cluster leaves, the original protocol will have some changes in the 3rd round which supplements the variant partial decryption of the vehicle's secret key.

4.1 Data Aggregation Protocol Against $N-1$ Corruptions

We have some similar processes with the two-round MPC protocol in [16], so we will not dwell on these. We remind readers to consult [16] for details. And now we describe the additional process. Let $f : (\{0,1\}^{\ell_{im}})^N \rightarrow \{0,1\}^{\ell_{out}}$ be the function to compute.

Round 1. Each party P_k executes the key generation function of the MFHE scheme in [16], and then broadcast the public key pk_k

Round 2. Each party P_k on receiving values $\{pk_i\}_{i\in[N]\setminus\{k\}}$ executes the following steps:

- Split the secret key $\{\{s_j\}_{j\in[N]\setminus\{k\}}\} \leftarrow \text{SecSplit}(N, k, sk_k)$.
- Execute the MFHE encryption function for the secret key sharing $\{cs_{i,g} \leftarrow \text{Encrypt}(pk_i, s_i[g])\}_{i\in[N]\setminus\{k\},g\in[2n\lceil\log q\rceil]}$ bit-by-bit and then broadcast these ciphertexts.

Round 3. On receiving these values $\{cs_{k,g}\}_{g\in[2n\lceil\log q\rceil]}$, if all vehicles are still in the cluster, the final decryption will be executed as [16]. And if the vehicle P_s leaves the cluster, the following steps will be executed:

1. Each P_k decrypts these sharing ciphertexts $\{cs_{k,g}\}_{g\in[2n\lceil\log q\rceil]}$ encrypted by pk_k of the secret key sk_s and reconstructs s_k.
2. Each P_k computes the partial decryption $p_k^{(j)} \leftarrow \text{PartDec}(\hat{c}_j, k, sk_k)$ and the variant partial decryption $(\tau_k^{(j)}, \upsilon_k^{(j)}) \leftarrow \text{SharPartDec}(s_k, \hat{c}_j, k, N)$ of P_s for all $j \in [\ell_{out}]$.
3. Then P_k will broadcast all the above values $\{p_k^{(j)}, \upsilon_k^{(j)}, \tau_k^{(j)}\}_{j\in[\ell_{out}]}$.

Output

1. On receiving the values $\{p_k^{(j)}\}_{j\in[\ell_{out}]}$ run the final decryption to obtain the j-th bit $\{y_j \leftarrow \text{FinDec}(p_1^{(j)}, \ldots, p_N^{(j)})\}_{j\in[\ell_{out}]}$ and then Output $y = y_1 \cdots y_{\ell_{out}}$.
2. On receiving the values $\{p_i^{(j)}, \upsilon_i^{(j)}, \tau_i^{(j)}\}_{j\in[\ell_{out}],i\in[N]\setminus\{s\}}$, run the partial sharing decryption to obtain $\{p_s^{(j)\prime} \leftarrow \text{SharFinDec}(\{\upsilon_i^{(j)}\tau_i^{(j)}\}_{i\in[N]\setminus\{s\}})\}_{j\in[\ell_{out}]}$ and then run the final decryption to obtain $\{y_i \leftarrow \text{FinDec}(p_1^{(j)}, \ldots, p_i^{(j)\prime}, \ldots, p_N^{(j)})\}_{j\in[\ell_{out}]}$.

   Then Output $y = y_1 \cdots y_{\ell_{out}}$.

4.2 Correctness and Security Analysis

Formally we prove the following theorem.

Theorem 4.1. *Let f be a poly-time computable deterministic function with N inputs and 1 output. Let the scheme* MFHE = (Setup, Kengen, Encrypt, Expand, Eval, PartDec, FinDec, SecSplit, SharPartDec, SharFinDec) *be a multi-key FHE scheme with variant threshold decryption. Then the protocol described in Sect. 4.1 UC-realize the function f against any semi-honest adversary corrupting exactly N-1 vehicles in a cluster.*

Proof. The correctness of the protocol follows in a straightforward way from the correctness of the underlying variant threshold MFHE scheme.

To prove the security we construct an efficient (PPT) simulator $\mathcal{S}$ for any adversary corrupting exactly *N-1*. Let A be a semi-honest adversary, P_h be the only honest party and P_s be the vehicle left the cluster.

The Simulator. In round 2, the simulator encrypt 0s as the simulated sharing encryption $\{cs'_{k,g}\}_{g\in[2n\lceil\log q\rceil]}$ instead of the real ones. In round 3, it computes the simulated variant partial decryption $sp_i' \leftarrow \mathcal{S}^{vthr}(sk_s, \hat{c}, i, (s_j)_{j\in[N]\setminus\{s,h\}})$ instead of the correctly computed values generated via SharPartDec(…).

Hybrid Games. We now define a series of *hybrid games* that will be used to prove the indistinguishability of the real and ideal worlds:

$$\mathrm{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}} \overset{comp}{\approx} \mathrm{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$$

The output of each game is always just the out of the environment.

The game $\mathrm{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$**:** This is exactly an execution of the protocol π in the real world with environment $\mathcal{Z}$ and semi-honest adversary $\mathcal{A}$.

The game $\mathrm{HYB}^1_{\pi,\mathcal{A},\mathcal{Z}}$**:** In this game, we modify the real world experiment as follows. Assume that P_h is given the simulated sharing encryption $\{cs'_{k,g}\}_{g\in[2n\lceil\log q\rceil]}$ after round 2. In the 3rd round, instead of broadcasting a correctly generated sharing encryption $\{cs_{k,g}\}_{g\in[2n\lceil\log q\rceil]}$, it broadcasts simulated ones.

The game $\mathrm{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$**:** In this game, we modify the game $\mathrm{HYB}^1_{\pi,\mathcal{A},\mathcal{Z}}$ as follows. Assume that P_h is given all the sharing $\{s_j\}_{j\in[N]\setminus\{s,h\}}$ of the secret keys $\mathbf{t}_s$ after round 2. In the 3rd round, instead of broadcasting a correctly generated variant partial decryption sp_i generated via SharPartDec(…), it broadcasts simulated ones $sp_i' \leftarrow \mathcal{S}^{vthr}(sk_s, \hat{c}, i, \{s_j\}_{j\in[N]\setminus\{s,h\}})$.

Claim 4.2. $\mathrm{REAL}_{\pi,\mathcal{A},\mathcal{Z}} \overset{stat}{\approx} \mathrm{HYB}^1_{\pi,\mathcal{A},\mathcal{Z}}$

Proof. The only changes between those experiments are in generating encryption of party P_h. We have the following lemma:

Lemma 4.3. *The* MFHE *scheme described in Sect. 3.1 satisfies semantic security.*

The semantic security of the above MFHE scheme has been proved in detail in reference [8]. We refer the reader to [8] for details. So the encryptions are also computationally indistinguishable.

Claim 4.4. $\mathrm{HYB}^{1}_{\pi,\mathcal{A},\mathcal{Z}} \overset{comp}{\approx} \mathrm{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$

Proof. The only changes between those experiments are that the variant partial decryption of party P_h is generated through simulator $\mathcal{S}^{vthr}$ instead of correctly using SharPartDec(…). By simulation security the variant partial decryptions are statistically indistinguishable hence so are the experiments.

This concludes the proof of the theorem.

4.3 Complexity Analysis

In this section, we analyze the communication complexity and computational complexity of our protocols. And for simplicity, we will take the vehicle P_h as the example to carry out the analysis.

In round 1, the public keys are generated and broadcasted in the cluster. So for fixed parameters, the communication complexity is $\omega(d^2\,\lambda^2\,(\log\,\lambda)^2)$. In round 2, it is $\omega(\ell_{in}d^2\,\lambda^2\,(\log\,\lambda)^2)+\omega(d^3\,\lambda^3\,(\log\,\lambda)^3)$. In round 3, it is $\omega\,(\ell_{out}d\,\lambda\,(\log\,\lambda))$. As described above, the total communication complexity is

$$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)+\omega(d^5\,\lambda^5\,(\log\,\lambda)^5).$$

In the execution of the entire protocol, the function Encrypt(…) has been invoked for $\ell_{in}+2n(N-1)\log\,q$ times. And the function performs nm^4 multiplication operations every time. So the computation complexity is

$$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)+\omega(d^5\,\lambda^5\,(\log\,\lambda)^5).$$

We list the differences in complexity between our scheme and some other related scheme in Table 1. Compared with the previous scheme.

Table 1. Complexity comparison.

	Communication complexity	Computation complexity
Clear and Mcgoldrick [8]		$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)$
Mukherjee and Wichs [16]	$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)$	$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)$
Our scheme	$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)$ + $\omega(d^5\,\lambda^5\,(\log\,\lambda)^5)$	$\omega(\ell_{in}d^4\,\lambda^4\,(\log\,\lambda)^4)$ + $\omega(d^5\,\lambda^5\,(\log\,\lambda)^5)$

5 Conclusion

This paper main contributes to the data aggregation protocol based on MFHE in VANETs. To adapt the existed schemes to the new situation, a novel protocol based on MFHE is proposed. The main conclusion as follow:

Considering the dynamic structure of the vehicle cluster, after the variant partial decryption, we can realize the data aggregation in the more complex situation. And the multi-hop evaluation can be performed in this environment. On the other hand, because too many cryptographic suites and matrix operations are invoked, the performance of the proposed scheme is much lower than that of the previous one. The above will be the focus of our future research.

Acknowledgement. This work is supported by National Science Foundation of China P.R. (NSFC) under Grants 61573076, 61703063, 61663008; the Scientific Research Foundation for the Returned Overseas Chinese Scholars under Grant 2015-49; the Program for Excellent Talents of Chongqing Higher School under Grant 2014-18; Science and Technology Research Project of Chongqing Municipal Education Commission of China P.R. under Grants KJ1705121, KJ1600518, KJ1705139, KJ1605002, KJZD-K201800701; Chongqing Research Program of Basic Research and Frontier Technology under Grant CSTC2017jcyjAX0411; Chongqing Municipal Social Livelihood Science and Technology Innovation Project under Grant CSTC2016shmszx30026; Program of Chongqing Innovation and Entrepreneurship for Returned Overseas Scholars of China P.R under Grant cx2018110; Graduate Education and Innovation Foundation Project of Chongqing Jiaotong University under Grant 2018S0145.

References

1. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009, vol. 9, no. 4, pp. 169–178 (2009)
2. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Foundations of Computer Science, pp. 97–106. IEEE (2011)
3. Plantard, T., Susilo, W., Zhang, Z.: Fully homomorphic encryption using hidden ideal lattice. IEEE Trans. Inf. Forensics Secur. **8**(12), 2127–2137 (2013)
4. Gentry, C., Halevi, S.: Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In: Annual Symposium on Foundations of Computer Science, vol. 47, no. 10, pp. 107–109 (2011)
5. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory **6**(3), 1–36 (2014)
6. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5
7. Lópezalt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the Annual ACM Symposium on Theory of Computing, pp. 1219–1234 (2012)
8. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 630–656. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_31

9. Wan, S., Zhang, Y., Chen, J.: On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks. IEEE Sens. J. **16**(20), 7433–7440 (2016)
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: DBLP, pp. 197–206 (2008)
11. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. ACM (2013)
12. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. ACM (2009)
13. Wan, S.: Energy-efficient adaptive routing and context-aware lifetime maximization in wireless sensor networks. Int. J. Distrib. Sens. Netw. **2014**(2), 112–116 (2014)
14. Wan, S., Zhang, Y.: Coverage hole bypassing in wireless sensor networks. Comput. J. **60** (10), 1536–1544 (2017)
15. Peikert, C., Shiehian, S.: Multi-key FHE from LWE, revisited. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9986, pp. 217–238. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_9
16. Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 735–763. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_26
17. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, pp. 84–89. ACM (2005)
18. Gopinath, V., Bhuvaneswaran, R.S.: Design of ECC based secured cloud storage mechanism for transaction rich applications. CMC: Comput. Mater. Continua **57**(2), 341–352 (2018)
19. Zhong, J., Liu, Z., Xu, J.: Analysis and improvement of an efficient controlled quantum secure direct communication and authentication protocol. CMC: Comput. Mater. Continua **57**(3), 621–633 (2018)