Chen, Jingxue; Liu, Gao; Liu, Yining

Lightweight Privacy-preserving Raw Data Publishing Scheme

Published in:
IEEE Transactions on Emerging Topics in Computing

DOI:
10.1109/TETC.2020.2974183

Published: 01/12/2021

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Chen, J., Liu, G., & Liu, Y. (2021). Lightweight Privacy-preserving Raw Data Publishing Scheme. *IEEE Transactions on Emerging Topics in Computing*, *9*(4), 2170-2174. https://doi.org/10.1109/TETC.2020.2974183

Lightweight Privacy-preserving Raw Data Publishing Scheme

Jingxue Chen, Gao Liu, and Yining Liu

Abstract—Data publishing or data sharing is an important part of analyzing network environments and improving the Quality of Service (QoS) in the Internet of Things (IoT). In order to incentives data providers (i.e., IoT end-users) to contribute their data, privacy requirement is necessary when data is collected and published. In traditional privacy preservation techniques, such as k-anonymity, data aggregation and differential privacy, data is modified, aggregated, or added noise, the utility of the published data are reduced. Privacy-preserving raw data publishing is a more valuable solution, and n-source anonymity based raw data collection is most promising by delinking raw data and their sources. In this paper, a lightweight raw data collection scheme for publishing is proposed, in which the rawness and the unlinkability of published data are all really guaranteed with Shamir's secret sharing, and shuffling algorithm. Moreover, it is lightweight and practical for the IoT environment by the performance evaluation.

Index Terms—data collection, privacy, rawness, unlinkability, lightweight

1 Introduction

With the rapid development of the Internet of Thing (IoT), various IoT devices are used in many applications [1], such as smart grid [2], vehicle network [3], body area network [4]. These IoT devices really facilitate daily life, however, data privacy [5] [6] concerns should be addressed since the data from IoT devices consist of the sensitive information [7].

In past decades, k-anonymity [8] [9] and differential privacy [10] are widely researched to guarantee privacy when data is published. Specifically, k-anonymity guarantees that each person cannot be distinguished from at least other $k-1$ individuals by modifying corresponding attributes, meanwhile, differential privacy adds noise to the published data to avoid the disclosure of private information records.

However, both k-anonymity and differential privacy are used to protect the privacy of the data that has been collected and stored in data center, in fact, it is under the assumption that the data center is fully trusted since it owns or knows all stored data. However, the assumption that a data center or edge nodes connecting to IoT devices are fully trusted is not practical. Therefore, the edge nodes and data center should not directly obtain raw data from IoT devices, instead, the raw data collected with IoT devices should be masked before it is sent to other nodes. Two requirements are necessary, namely data privacy and utility.

Data aggregation allows a data center to obtain the average, maximum or minimum of data in an area without knowing individual data [11]. However, in some application scenarios, the average, maximum or minimum of data cannot meet the needs, a variety of fine-grained data is required. Recently, a privacy-preserving computing function library is designed based on Intel Software Guard Extensions (SGX) [12]. However, Intel SGX may suffer from attack such as side-channel attack [13]. For data utilization, n-source anonymity is a feasible solution by delinking data and its source [14], where a piece of data is protected from an n-member group and simultaneously the rawness of data is ensured. Current n-source anonymity based raw data collection schemes mainly use virtual rings [15], a trusted third party (TTP) [14] and shuffling [16] to reserve slots for loading data. However, the sensitive data of an IoT device in virtual rings can be derived due to the collusion attack of its upstream and downstream devices. In addition, it is hard to deploy a TTP in practice. Hence, shuffle is used to replace the role of TTP, and simultaneously to ensure the rawness and unlinkability. Unfortunately, when n IoT devices construct a group for masking their data, in [15] each IoT device of virtual ring reserves $n/2$ slots on average, and in [14], [16], n slots are reserved. As a consequence, the heavy storage cost is brought to each IoT device when n is large.

In this paper, a lightweight raw data publishing scheme is proposed using secret sharing and shuffle, and two contributions are achieved as follows.

- Data privacy and utility are balanced by guaranteeing the unlinkability and the rawness.

- The lightweight requirement is achieved to make it more suitable for IoT devices.

The rest of this paper is organized as follows. Preliminaries are introduced in Section 2, problem definition is discussed in Section 3, our scheme and its analysis are presented in Section 4 and Section 5 respectively. Finally, the paper is concluded in Section 6.

This work was supported by the National Natural Science Foundation of China under grant no. 61662016, and Key projects of Guangxi Natural Science Foundation under grant no. 2018JJD170004.

Jingxue Chen and Yining Liu are with Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, 541004, China. Gao Liu is with State Key Laboratory on Integrated Services Networks and School of Cyber Engineering, Xidian University, Xi'an, 710126, China and with the Department of Communications and Networking, Aalto University, Espoo, Finland. (Corresponding author: Yining Liu, E-mail: ynliu@guet.edu.cn)

2 Preliminaries

2.1 Bilinear Pairings

The p order cyclic additive group G_1, and p order cyclic multiplicative group G_2, and a mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ ($\hat{e}$ is a bilinear pairing) are defined. Assume that P is a generator of G_1, which satisfies:

(1) Bilinearity: $\forall a, b \in Z_p^*$, $\hat{e} : (aP, bP) = \hat{e}(P, P)^{ab}$;
(2) Non-degeneracy: $\hat{e} : (P, P) \neq 1_{G_1}$;
(3) Computability: For any $P, Q \in G_1$, exists an efficient algorithm to compute $\hat{e} : (P, Q)$.

2.2 Shuffle

In [16], a shuffle algorithm is used to allocate slot locations, in which a ciphertext is re-randomized without changing the corresponding plaintext. The shuffle algorithm is defined as follows:

(1) Input $c_i, (i \in [1, n])$ that is then encrypted into c_i^*, and rearranged using random permutation;

(2) Output the new permutated list.

More details can refer [16].

2.3 Shamir's secret sharing

In [17], assuming there are n users $\{U_1, U_2, ..., U_n\}$, and a trusted dealer D.

Shares generation

D chooses $t-1$ random numbers $a_1, a_2, \ldots, a_{t-1}$ and generates a polynomial $f(x) = s + a_1 x + \cdots + a_i x^i + \cdots + a_{t-1} x^{t-1}$ over a finite field F_p, where p is a prime number, s is a secret, and t is a threshold value. D sends $y_i = f(x_i)$ to $U_i, (i \in [1, n])$ via a secure channel.

Secret reconstruction

The secret is recovered by computing

$$s = f(0) = \sum_{i=1}^{t} \prod_{j=1, j \neq i}^{t} \frac{0 - x_j}{x_i - x_j} y_i.$$

2.4 Secret sharing homomorphism

Secret sharing homomorphism [18] is a useful tool for privacy-preserving computation. For example, in e-voting, an election center obtains the sum of voters' ballots without knowing individual ballots [19]. Assume there are two polynomials $f(x)$ and $g(x)$, and s_1, s_2 are their secrets to be shared respectively.

(1) Dealer D sends $f(i)$ and $g(i)$ to the corresponding user U_i, where $i \in [1, n]$.

(2) U_i computes and sends $f(i) + g(i)$, $(i \in [1, n])$ to D.

(3) D recovers the secret $s_1 + s_2$ due to the additive homomorphism.

3 Problem Definition

3.1 System model

As shown in Fig.1, the system model consists of Cloud Server (CS), Fog Node (FN) and users $U_i, (i \in [1, n])$ in group.

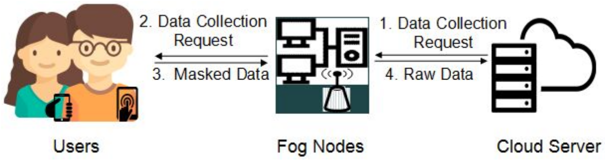

Fig. 1. System model

(1) CS sends a data collection request to FN, then FN forwards the request to $U_i, (i \in [1, n])$.
(2) After receiving the request, $U_i, (i \in [1, n])$ collects and masks the data, then sends the masked data to FN.
(3) The raw data collected from $U_i, (i \in [1, n])$ are extracted by FN, then sent to CS, meanwhile the relation between data and its source is privacy for all.

3.2 Assumptions and threat model

Only privacy issues are concerned under the assumption that secure communication channels have been established using cryptographic techniques among these entities [11] [16], and the semi-honest model is followed, i.e., CS, FN and users follow the protocols, meantime, they are curious to know the source of a piece of data.

• Data collectors (CS, FN) want to infer the data source from the information they received.

• Besides the data collectors, other users also attempt to collude to infer a user's data.

4 Our scheme

Our scheme consists of *Configuration Phase* and *Data Collection Phase*. Details are as follows.

4.1 Configuration phase

Key generation

Step1: CS selects a security parameter γ and generates $\{p, P, G_1, G_2, \hat{e}\}$, where G_1, G_2 are p order cyclic groups, P is a generator of G_1, and $\hat{e}$ is a bilinear pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$.

CS chooses two secure cryptographic hash functions, $H : Z_p^* \rightarrow Z_p^*$ and $H_1 : \{0, 1\}^* \rightarrow G_1$.

CS publishes $\{p, P, G_1, G_2, \hat{e}, H, H_1\}$.

Step2: Each entity selects a random number $sk_{entity} \in Z_p^*$ as its secret key, then calculates and publishes $PK_{entity} = sk_{entity} P$.

Setup for data collection

U_i selects β_i partners in a group, generates a session key k_{ij} with each partner U_j, and obtains its session list $\{k_{i1}, k_{i2}, ..., k_{i\beta_i}\}$, $(\beta_i \in [1, n-1]), (i, j \in [1, n], i \neq j)$. For example, three users $\{U_1, U_2, U_3\}$ are in a group, U_1 and U_2 share the session key k_{12}, U_1 and U_3 share the session key k_{13}.

4.2 Data collection phase

Each user is assigned a position using shuffle [16], and this position corresponds to a coefficient in n-order degree polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$, and the data from $U_i's$ is placed in this position. U_i only knows its position, meanwhile knows nothing about others.

Data collection initialization

CS sends a data collection request $Req = \{ID_{CS}, ID_{FN}, T, \omega_{CS}\}$ to FNs, where ID_{CS} and ID_{FN} are identities of CS and FN, T is timestamp and $\omega_{CS} = sk_{CS}H_1(ID_{CS}||ID_{FN}||T)$.

Forwarding request

FN checks the timestamp T and the equation $\hat{e}(\omega_{CS}, P) = \hat{e}(H_1(ID_{CS}||ID_{FN}||T), PK_{CS})$. If yes, FN broadcasts the request.

Masking raw data

Step1: $U_i, (i \in [1, n])$ generates its polynomial

$$f_i(x) = a_{i,0}+\lambda_{i,1}x+\cdots+(RD_i+\lambda_{i,\ell})x^{\ell}+\cdots+\lambda_{i,n}x^n, \quad (1)$$

where $a_{i,0}$ is a random number selected by U_i, RD_i represents the data from U_i, and the masking information $\lambda_{i,\ell}$,$(\ell \in [1, n])$ is computed as follow:

$$\lambda_{i,\ell} = \sum_{j=1,j\neq i}^{\beta_i} (ID_i - ID_j)H(k_{ij}|\ell). \quad (2)$$

Step2: U_i generates $n + 1$ shares $Y_{i,j} = f_i(ID_j), (j = 1, ..., n), Y_{i,FN} = f_i(ID_{FN})$, where ID_i is $U_i's$ identification. U_i keeps $Y_{i,i}$ and sends $Y_{i,j}$ to $U_j, (j = 1, ..., n, j \neq i)$, sends $Y_{i,FN}$ and $a_{i,0}$ to FN.

Step3: U_i and FN obtain the masked data MSD_i and MSD_{FN} by adding the received shares.

Extracting raw data

FN checks if the equation

$$\hat{e}(P, \sum_{i=1}^{n} \omega_i) = \prod_{i=1}^{n} \hat{e}(PK_i, H_1(MSD_i||ID_i||ID_{FN}||T_i||a_{i,0})) \quad (3)$$

holds, then recoveries

$$f(x) = a_0 + a_i x^i + \cdots + a_n x^n \quad (4)$$

using (ID_1, MSD_1), (ID_2, MSD_2),$\cdots$, (ID_{FN}, MSD_{FN}).

$\{a_1, \ldots, a_n\}$ are collected by FN, when the equation $a_0 = \sum_{i=1}^{n} a_{i,0}$ holds. Moreover, $\{a_1, \ldots, a_n\}$ are raw, which is not aggregated or averaged, not added noise, not modified. Nobody can link the element in two sets $\{a_1, \ldots, a_n\}$, $\{U_i, ..., U_n\}$.

An example of raw data collection

Assume that $p = 137$, U_1, U_2, U_3 and FN (their identifications are $1, 2, 3, 4$) collaborate to collect the heartbeats, U_1, U_2, U_3 correspond to three coefficients of a 3-degree polynomial after shuffling, such as x^2, x and x^3, U_1 and U_2 share $k_{12} = 2$, U_1 and U_3 share $k_{13} = 3$.

U_1 calculates $Y_{1,1} = f_1(1)$, $Y_{1,2} = f_1(2)$, $Y_{1,3} = f_1(3)$ and $Y_{1,FN} = f_1(4)$, then keeps $Y_{1,1}$ secret and sends $Y_{1,2}$ to U_2, $Y_{1,3}$ to U_3, and $Y_{1,FN}$ to FN. The similar computation is for U_2, U_3. The shares received by each user and FN are listed in TABLE 1.

TABLE 1
The shares distribution

		Receiver			
		U_1	U_2	U_3	FN
Sender	U_1	$Y_{1,1}$	$Y_{1,2}$	$Y_{1,3}$	$Y_{1,FN}$
	U_2	$Y_{2,1}$	$Y_{2,2}$	$Y_{2,3}$	$Y_{2,FN}$
	U_3	$Y_{3,1}$	$Y_{3,2}$	$Y_{3,3}$	$Y_{3,FN}$

U_i computes its masked data $MSD_i = Y_{1,i}+Y_{2,i}+Y_{3,i}$, and sends $\{MSD_i, ID_i, ID_{FN}, T_i, a_{i,0}\}$ to FN, where T_i is a timestamp and $\omega_i = sk_i H_i(MSD_i||ID_i||ID_{FN}||T_i||a_{i,0})$, $(i = 1, 2, 3)$.

Table 2 illustrates the following computation:

(1) The heartbeats of U_1, U_2 and U_3 are 78, 60 and 85 respectively.

(2) $a_{1,0} = 5$, $a_{2,0} = 2$, and $a_{3,0} = 1$.

(3) $\lambda_{1,1} = (1 - 2) \cdot H(2|1) + (1 - 3) \cdot H(3|1)$=80,
$\lambda_{1,2} = (1 - 2) \cdot H(2|2) + (1 - 3) \cdot H(3|2)$=52,
$\lambda_{1,3} = (1 - 2) \cdot H(2|3) + (1 - 3) \cdot H(3|3)$=106,
$\lambda_{2,1} = (2 - 1) \cdot H(2|1)$=27,
$\lambda_{2,2} = (2 - 1) \cdot H(2|2)$=69,
$\lambda_{2,3} = (2 - 1) \cdot H(2|3)$=114, and
$\lambda_{3,1} = (3 - 1) \cdot H(3|1)$=30,
$\lambda_{3,2} = (3 - 1) \cdot H(3|2)$=16,
$\lambda_{3,3} = (3 - 1) \cdot H(3|3)$=54.

(4) 94, 24, 34 are the sum of the received shares of U_1, U_2, U_3, they are sent to FN.

FN recovers the polynomial $f(x) = 8 + 60x + 78x^2 + 85x^3$ mod 137, and ensures $\{60, 78, 85\}$ is the heartbeats set of $\{U_1, U_2, U_3\}$ when the equation $a_{1,0}+a_{2,0}+a_{3,0} = a_0$ holds, namely 5+2+1=8.

5 Analysis

In this section, the proposed scheme is analyzed to really achieve the rawness and unlinkability. Moreover, the performance in terms of storage and computational burdens are evaluated, and the comparison with the excellent techniques are shown in TABLE 3.

5.1 Security analysis

Rawness

Raw data RD_i can be obtained by reconstructing secrets on public shares.

Proof: Assuming that U_i is assigned an unique coefficient of x^{ℓ}, $(l = 1, ..., n)$ after shuffling, before no masking information is added, the polynomial is

$$f_i(x) = a_{i,0} + RD_i x^{\ell}, (i = 1, ..., n) \quad (5)$$

Then, the set of coefficient of $a_1, \cdots, a_n$ of $\sum_{i=1}^{n} f_i(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ corresponds the raw data RD_i, $(i = 1, ..., n)$ from $U_i, (i = 1, ..., n)$.

When masking information is added, $U_i's$ polynomial is $f_i(x) = a_{i,0}+\lambda_{i,1}x+\cdots+(RD_i+\lambda_{i,\ell})x^{\ell}+\cdots+\lambda_{i,n}x^n$, where $\lambda_{i,\ell} = \sum_{j=1,j\neq i}^{\beta_i}(ID_i - ID_j)H(k_{ij}|\ell)$ and $\sum_{i=1}^{n} \lambda_{i,\ell} = 0$.

The shares of each user's polynomial are generated and sent to others, which guarantees that the sum of received shares $MSD_1, \ldots, MSD_n, MSD_{FN}$ of $U_i, (i = 1, ..., n)$ and FN to correctly recover $a_0 + a_1x + \cdots + a_nx^n$ that is same as Equation 5 using secret sharing homomorphism [18].

$\{a_1, \ldots, a_n\}$ are ensured to be the raw data from $\{U_i, (i = 1, ..., n)\}$ by FN when $a_0 = \sum_{i=1}^{n} a_{i,0}$.

Unlinkability

Nobody can link Raw data RD_i and its source $U_i, (i = 1, ..., n)$, even if one honest partner exists.

Proof: U_i generates $n + 1$ shares with its polynomial $f_i(x) = a_{i,0} + \lambda_{i,1}x + \cdots + \lambda_{i,\ell-1}x^{\ell-1} + (RD_i + \lambda_{i,\ell})x^{\ell} + \cdots + \lambda_{i,n}x^n$, where $n - 1$ shares are sent to other users

TABLE 2
An example over F_p, and $p = 137$

	ID	polynomial	p_1	p_2	p_3	FN
p_1	1	$f_1(x) = 5 + (80)x + (78 + 52)x^2 + (106)x^3$	(1,47)	(2,26)	(3,30)	(4,10)
p_2	2	$f_2(x) = 2 + (60 + 27)x + 69x^2 + 114x^3$	(1,135)	(2,131)	(3,126)	(4,119)
p_3	3	$f_3(x) = 1 + 30x + 16x^2 + (85 + 54)x^3$	(1,49)	(2,4)	(3,15)	(4,94)
MSD_i			(1,94)	(2,24)	(3,34)	(4,86)

in the group, and $a_{i,0}$ and one share are sent to FN. In our hypothesis, there is at least one honest partner (except U_i) in the group who as a user will not betray U_i. Even if other $n-2$ users collude with FN to recover $f_i(x)$, however, $n-$degree polynomial cannot be recovered only with them, since at least $n+1$ points are necessary. Therefore, $f_i(x)$ is still unknown for all.

Furthermore, the masking information $\lambda_{i,\ell} = \sum_{j=1,j\neq i}^{\beta_i}(ID_i - ID_j)H(k_{ij}|\ell)$ is still private, since at least one session key cannot be obtained by an adversary under the assumption that U_i's honest partner exist. Therefore, U_i's masking information, $\lambda_{i,1}$, $\cdots$ $\lambda_{i,\ell}$, $\cdots$, $\lambda_{i,n}$ is private even if $\beta_i - 1$ partners collude with FN. As a consequence, the collected raw data RD_i cannot be obtained by calculating $(RD_i + \lambda_{i,\ell}) - \lambda_{i,\ell}$, so that its privacy is preserved.

5.2 Performance evaluation

The performance of our scheme is evaluated and compared with [14] and [16].

Storage burden

Assume that the size of each data in [14] [16] is $L\ bits$, the masked data $\Re_i = e_i^1|e_i^2|...|e_i^{slot(i)} \oplus m_i|\cdots|e_i^n$ occupies $nL\ bits$. When n grows, the storage cost increases linearly. The masked data MSD_i of user U_i is the sum of all received shares, is only $L\ bits$.

Computational complexity

Assume that there are n users with an FN, and a simulation is executed using a laptop with Intel i5 2.5 GHz CPU and 8.00 GB memory, and computational complexity in shuffle and data collection is evaluated.

(1) **Computational complexity in shuffle**

The total computational complexity of shuffle is $\mathcal{O}(nlogp)$, since the shuffle technique adopts the ElGamal encryption [16]. U_i encrypts its own pseudonym, shuffles a cipher list, and sends the new cipher list to its successor. The efficiency in shuffle depends on the length of pseudonym and users′ position (i.e., the order in a transmission sequence) in a group. In our experiment, the position ranges from $[200, 1000]$, and the length of pseudonyms is 128 $bits$ and 256 $bits$ respectively. The result is shown in TABLE 4. When the group size is 1000, the computation time of U_{1000} is only 394 ms.

(2) **Computational complexity in data collection**

Only the bilinear pairing operations are counted in data collection. Specifically, FN needs to execute two bilinear pairing operations to verify CS's data collection request, and $n+1$ bilinear pairing operation when checking the equation $\hat{e}(P, \sum_{i=1}^{n}\omega_i) = \prod_{i=1}^{n}\hat{e}(PK_i, H_1(MSD_i||ID_i||ID_{FN}||T_i||a_{i,0}))$ to extract the raw data. Thus, FN takes $n+3$ [20] pairing operation. In addition, pairing operation execution T_{pair} is about 2.187ms.

TABLE 4
The computation time of shuffle (ms)

		Group size				
		200	400	600	800	1000
The length of pseudonym	128$bits$	106	176	248	320	388
	256$bits$	120	187	262	321	394

5.3 Comparison

Our scheme is compared with other well known privacy technique in TABLE 3, including TTP reliance, unlinkability, rawness, and computational complexity, etc.

6 Conclusion

In this paper, a lightweight raw data collection for data publishing is proposed based on secret sharing and shuffling algorithm, which is proved and evaluated to be more practical due to the efficiency and the privacy.

References

[1] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Estimating age privacy leakage in online social networks," in *proc. of ACC*, 2012, pp. 21–29.

[2] Z. Erkin, J. R. Troncoso-Pastoriza, R. L. Lagendijk, and F. Pérez-González, "Privacy-preserving data aggregation in smart metering systems: An overview," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 75–86, 2013.

[3] Y. Liu, S. Lv, M. Xie, Z. Chen, and P. Wang, "Dynamic anonymous identity authentication (daia) scheme for vanet," *International Journal of Communication Systems*, vol. 32, no. 5, p. e3892, 2019.

[4] M. Rabbi, S. Ali, T. Choudhury, and E. Berke, "Passive and in-situ assessment of mental and physical well-being using mobile sensors," in *proc. of UBIC*, 2011, pp. 385–394.

[5] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving svm for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[6] Y. Zhu, Y. Zhang, X. Li, H. Yan, and J. Li, "Improved collision-resisting secure nearest neighbor query over encrypted data in cloud," *Concurrency and Computation: Practice and Experience*, p. e4681, 2018.

[7] R. Dey, C. Tang, K. Ross, and N. Saxena, “Estimating age privacy leakage in online social networks,” in *proc. of INFOCOM*, 2012, pp. 2836–2840.

[8] P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression,” technical report, SRI International, Tech. Rep., 1998.

[9] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[10] C. Dwork, “Differential privacy,” *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011, doi:10.1007/978-1-4419-5906-5_752.

[11] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, “A practical privacy-preserving data aggregation (3pda) scheme for smart grid,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1767–1774, 2019.

[12] S. Li and K. Xue, “Secgrid: A secure and efficient sgx-enabled smart grid system with rich functionalities,” *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1810.01651

[13] Y. Wu, W. Zheng, B. Mao, and X. Wu, “Leaks or not: A framework for evaluating cache timing side channel attacks in sgx,” in *proc. of IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IO*, 2018, pp. 1467–1470.

[14] Y. Zhang, Q. Chen, and S. Zhong, “Privacy-preserving data aggregation in mobile phone sensing,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 980–992, 2016.

[15] M. Badra and S. Zeadally, “Design and performance analysis of a virtual ring architecture for smart grid privacy,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 321–329, 2014.

[16] Y. Liu, Y. Wang, X. Wang, Z. Xia, and J. Xu, “Privacy-preserving raw data collection without a trusted authority for iot,” *Computer Networks*, vol. 148, pp. 340 – 348, 2019.

[17] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[18] J. C. Benaloh, “Secret sharing homomorphisms: keeping shares of a secret secret (extended abstract),” in *proc. of EUROCRYPT*, 1987, pp. 251–260.

[19] Y. Liu and Q. Zhao, “E-voting scheme using secret sharing and k-anonymity,” *World Wide Web*, vol. 22, no. 4, pp. 1657–1667, 2019.

[20] Y. Zhang, J. Zhao, D. Zheng, K. Deng, F. Ren, and X. Zheng, “Privacy-aware data collection and aggregation in iot enabled fog computing,” in *proc. of ICA3PP*, 2018, pp. 581–590.

TABLE 3
Comparison of mainstream privacy preservation techniques and our scheme

	Mainstream privacy preservation techniques					
	k-anonymity	Differential privacy	Data aggregation	n-source anonymity		
	[9]	[10]	[11]	[14]	[16]	Our scheme
Data center	Fully trusted	Fully trusted	Semi-honest	Semi-honest	Semi-honest	Semi-honest
TTP	NO	NO	NO	YES	NO	NO
Rawness	NO	NO	NO	YES	YES	YES
Unlinkability	YES	YES	YES	YES	YES	YES
Masked data storage	-	-	$2Lbits$	$nLbits$	$nLbits$	$Lbits$
Computational complexity of generating masked data	-	-	-	$2n$ hashes $2nL$-bit XORs	$n\beta_i$ hashes $n(\beta_i - 1)L$-bit XORs	$n\beta_i$ hashes secret shares generation
Computational complexity of extracting raw data	-	-	-	$(n-1)nL$ -bit XORs	$(n-1)nL$ -bit XORs	Secret recovery

$^{-}$ this is not considered