SoK: Privacy-preserving Clustering (Extended Abstract)*

Aditya Hegde
Intern at TU Darmstadt

Helen Möllering, Thomas Schneider, Hossein Yalame
TU Darmstadt

Abstract

Clustering is a popular unsupervised machine learning technique that groups similar input elements into clusters. In many applications, sensitive information is clustered that should not be leaked. Moreover, nowadays it is often required to combine data from multiple sources to increase the quality of the analysis as well as to outsource complex computation to powerful cloud servers. This calls for efficient privacy-preserving clustering. In this work, we systematically analyze the state-of-the-art in privacy-preserving clustering. We implement and benchmark today's four most efficient fully private clustering protocols by Cheon et al. (SAC'19), Meng et al. (ArXiv'19), Mohassel et al. (PETS'20), and Bozdemir et al. (ASIACCS'21) with respect to communication, computation, and clustering quality.

1 Introduction

Many large IT companies, including Microsoft, Facebook, Google, and Apple, collect massive amounts of data to perform analyses for their commercial benefit [2]. Clustering is a popular unsupervised learning technique and plays a crucial role in data processing and analysis. The regulations like GDPR emphasize the need for privacy-preserving clustering to preserve the privacy of data. Consequently, a series of efforts have been made through two paradigms for secure computation, homomorphic encryption (HE) [3–5] and secure multi-party computation (MPC) [6, 7], that can also be combined. However, these works only cover a few clustering algorithms so far: K-means, K-medoid, Mean-shift, Gaussian Mixture Models Clustering (GMM), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), hierarchical clustering (HC), Affinity Propagation, and Mean-shift. Moreover, we found that only ten works (cf. [1, Tab. 1]) provide full privacy protection according to the ideal functionality for private clustering, i.e., they leak nothing beyond the output.

Our Contributions. Our Systematization of Knowledge (SoK) paper provides the following core contributions:

– The first comprehensive review and analysis of existing techniques used for privacy-preserving clustering with respect to security models, privacy limitations, efficiency, and further aspects.
– An empirical evaluation of the four most efficient and fully private clustering schemes [8, 7, 5, 9].
– An open source implementation of the clustering protocol of [5] and [9] in C++17. Implementations of the remaining two protocols that we also evaluate [8, 7] are publicly available.

2 Existing Protocols

Tab. 1 contains on overview of all 59 works on privacy-preserving clustering with secure computation techniques that we are currently aware of. It indicates the respective security model, used secure computation techniques, common types of leakages of intermediate values, the type of output, which and how many parties are involved in the protocol, the data partition, and other issues.

*The full version of this paper published at PET'21 [1].

Algorithm	Scheme	Privacy	Security	PETs	L1	L2	L3	L4	O1	O2	O3	Interactivity (Scenario)	Data	Other issues
K-means	[10, KDD'03]	✗	◐	HE+blinding	✗	✗	✗	✗	✗	✓	✗	all data owners (≥ 3)	v	
	[11, KDD'05]	✗	◐	HE+ASS+GC	✓	✓	✗	✗	✓	✓	✗	2PC	a	wrong division
	[12, ESORICS'05]	✗	◐	HE *or* OPE	✗	✓	✓	✗	✗	✓	✗	2PC	h	
	[13, CCS'07]	✓	◐	HE+ASS	✓	✓	✓	✗	✗	✓	✗	2PC	a	
	[14, SECRYPT'07]	✗	◐	blinding	✗	✓	✗	✗	✓	✓	✗	all data owners	v/h	
	[15, AINAW'07]	✗	◐	HE+ASS+OPE	✓	✗	✗	✗	✓	✓	✗	2PC	h	
	[16, PAIS'08]	✗	◐	ASS	✓	✓	✗	✗	✓	✓	✗	all data owners (≥ 4)	v	
	[17, WIFS'09]	✗	◐	HE	✗	✓	✗	✓	✓	✗	✗	data owners + 1 server	h	
	[18, KAIS'10]	✗	◐	HE+ASS	✓	✓	✗	✗	✓	✗	✗	all data owners	h	
	[19, PAISI'10]	✗	◐	SS	✓	✗	✗	✗	✓	✓	✗	Outsourcing ≥ 3 servers	a	
	[20, ISPA'10]	✗	◐	HE	✓	✓	✗	✗	✗	✓	✗	all data owners	v/h	
	[21, WIFS'11]	✗	◐	HE+GC	✓	✗	✓	✓	✓	✗	✗	Outsourcing, 3 servers	h	
	[22, ISI'11]	✗	◐	HE+ASS	✗	✗	✗	✗	✓	✗	✗	2PC	v	
	[23, TM'12]	✗	◐	SSS	✗	✗	✓	✗	✗	✓	✗	all data owners	h	distance calculation unclear
	[24, JIS'13]	✗	◐	HE	✗	✓	✓	✗	✓	✗	✗	data owners + 2 servers	h	
	[25, ICDCIT'13]	✗	●	SSS+ZKP	✗	✗	✓	✗	✗	✓	✗	all data owners	h	
	[26, ASIACCS'14]	✗	◐	HE	✗	✗	✗	✗	✓	✓	✗	outsourcing, 1 data owner + 1 server	–	insecure HE [27]
	[28, MSN'15]	✗	◐	HE	✗	✗	✗	✓	✗	✗	✗	outsourcing, data owners + 1 server	h	insecure HE [27]
	[29, IJNS'15]	✗	◐	HE	✗	✗	✗	✗	✗	✓	✗	all data owners	h	
	[30, CIC'15]	✓	◐	HE	✓	✓	✓	✗	✗	✓	✗	Outsourcing, 2 servers	h	
	[31, ICACCI'16]	✗	N/A	SS	✗	✗	✗	✗	✓	✗	✗	arbitrary number of servers	a	
	[32, ISPA'16]	✗	◐	blinding	✗	✗	✗	✓	✗	✓	✗	all data owners (≥ 3)	h	
	[33, SecComm'17]	✗	◐	HE	✓	✗	✗	✓	✗	✓	✗	outsourcing, ≥ 4 servers	h	
	[34, TII'17]	✗	◐	HE	✗	✗	✗	✗	✗	✗	✗	data owners + 1 server	h	
	[35, SAC'18]	✓	◐	HE	✓	✓	✓	✓	✗	✓	✗	Outsourcing, 1 server	–	
	[36, CLOUD'18]	✓	◐	HE	✓	✓	✓	✗	✗	✓	✗	Outsourcing, 2 servers	–	distance calculation unclear
	[37, CCPE'19]	✗	N/A	HE	✗	✗	✗	✗	✗	✓	✗	Outsourcing, 2 data owners + 1 server	h	insecure HE [27]
	[38, TCC'19]	✗	◐	HE	✓	✗	✗	✓	✓	✗	✗	Outsourcing	–	
	[39, Inf. Sci.'20]	✗	◐/●	HE+GC	✗	✗	✗	✗	✗	✓	✗	Outsourcing, 2 data owners + 1 server	h	
	[40, SCN'20]	✗	◐	HE+SKC	✓	✗	✗	✓	✗	✓	✗	Outsourcing, 3 servers	h	
	[7, PETS'20]	✓	◐	GC	✓	✓	✓	✗	✗	✓	✗	2PC/Outsourcing	h	
	[4, TKDE'20]	✗	◐	HE	✓	✗	✓	✗	✗	✓	✗	Outsourcing, 2 servers	a	
Kernel K-means	[41, KAIS'16]	✗	N/A	PKC	✓	✗	✗	✗	✓	✗	✗	Outsourcing, 1 server	–	security model
Possibilistic C-means	[42, TBD'17]	✗	N/A	HE	✗	✗	✗	✗	✓	✓	✗	Outsourcing, 1 data owner + 1 server	–	
K-medoids	[43, SMC'07]	✗	N/A	HE+blinding	✓	✗	✗	✓	✗	✗	✗	all data owners	v	exhaustive search
	[44, CCSEIT'12]	✗	N/A	HE+blinding	✓	✗	✗	✓	✗	✗	✗	all data owners	v	exhaustive search
GMM	[45, KAIS'05]	✗	◐	blinding	✓	✓	✗	✗	✓	✗	✗	all data owners	h	
	[46, DCAI'19]	✗	◐	ASS	✓	✓	✗	✗	✓	✗	✗	all data owners (> 2)	v/h	
Affinity Propagation	[47, INCoS'12]	✗	◐	HE + blinding	✓	✓	✗	✓	✓	✗	✗	all data owners	v	
	[48, SECRYPT'21]	✓	◐/●	ASS+GC	✓	✓	✓	✓	✓	✗	✗	all data owners/Outsourcing	a	
Mean-shift	**[5, SAC'19]**	✓	◐	HE	✓	✓	✓	✓	✓	✗	✗	Outsourcing, 1 server	–	
DBSCAN	[49, ISI'06]	✗	◐	blinding	✓	✓	✗	✓	✗	✗	✗	all data owners	v	lack of complete protocol
	[50, ADMA'07]	✗	◐	HE+blinding	✓	✗	✗	✓	✓	✗	✗	2PC	v/h	
	[51, IJSIA'07]	✗	◐	PKC+blinding	✓	✓	✗	✓	✓	✗	✗	all data owners	v	
	[52, ITME'08]	✗	◐	HE+blinding	✓	✗	✗	✓	✓	✗	✗	data owners + 1 server	h	
	[53, TDP'13]	✗	◐	HE+blinding	✓	✗	✗	✓	✓	✗	✗	2PC	a	
	[54, S&P'12]	✓	◐/●	GC	✓	✓	✓	✓	✓	✓	✗	2PC	h	
	[55, SIBCON'17]	✗	◐	HE+PKC	✓	✓	✗	✓	✓	✗	✗	all data owners	v	cluster expansion missing
	[56, PRDC'17]	✗	◐	HE	✓	✗	✗	✓	✗	✗	✗	all data owners + 1 server	h	
	[57, AI'18]	✗	◐	HE	✓	✗	✗	✓	✓	✗	✗	data owners + 1 server	a	uses absolute distance
	[8, ASIACCS'21]	✓	◐	ASS+GC	✓	✓	✓	✓	✓	✓	✗	2PC/Outsourcing	a	
HC	[58, SDM'06]	✗	◐	HE+ASS+GC	✓	✓	✗	✓	✗	✓	✗	2PC	h	
	[59, TKDE'07]	✗	◐	SKC	✓	✓	✗	✓	✓	✗	✗	data owners + 1 server	h	SKC not semantically secure
	[60, TDP'10]	✗	◐	HE+GC	✓	✓	✗	✓	✓	✓	✗	2PC	h	
	[61, ISI'14]	✗	N/A	HE	✓	✗	✗	✓	✓	✓	✗	2PC	v	
	[62, ISCC'17]	✗	◐	HE	✓	✓	✗	✓	✗	✗	✓	2PC	v/h	
	[9, ArXiv'19]	✓	◐	HE & GC	✓	✓	✓	✓	✗	✓	✓	2PC	h	
BIRCH	[63, SDM'06]	✗	◐	HE+ASS	✓	✓	✗	✓	✗	✗	✗	2PC	v	
	[64, ADMA'07]	✗	◐	HE+ASS	✓	✓	✗	✓	✗	✗	✗	2PC	a	

Table 1: History overview of privacy-preserving clustering using secure computation techniques. Privacy indicates if fully privacy protection according to the ideal functionality for privacy-preserving clustering is provided (✗: leakage; ✓: no leakage). ◐ is the semi-honest security model, ● is the malicious security model, N/A indicates that no security model was defined. HE is homomorphic encryption, ASS additive secret sharing, SSS Shamir's secret sharing, GC garbled circuits, OPE oblivious polynomial evaluation, PKC public-key cryptography, SKC symmetric-key cryptography, ZKP zero-knowledge proof, blinding is the use of random values for blinding, and other types of secret sharing are summarized by SS. v indicates that the data that shall be clustered is vertically distributed, i.e., the data owners hold the values for a subset of parameters from all data records. h indicates horizontally partitioned data where the data owners hold complete data records with all parameters, and a is arbitrary data partitioning. L1 leaks intermediate centroids, L2 intermediate cluster sizes, L3 other intermediate values (e.g., intermediate cluster assignments or distance comparison results), and L4 the number of clustering iterations. O1 outputs the final cluster labels/assignments, O2 outputs the final centroids, and O3 outputs the final dendogram/tree structure. The schemes with the best privacy guarantees are marked in bold (we do not consider the number of clustering iterations as a severe leakage as it can be easily avoided. The efficient and fully private schemes that we implemented and benchmarked are [5, 9, 8, 7].

3 Evaluation

Software Details. We implemented all four protocols in C++17 and instantiate all cryptographic building blocks with a security level of 128 bits.

Datasets. We use nine datasets from the well-known FCPS [65] and Graves [66] collections designed for benchmarking clustering algorithms. They also include the ground truth separation [67].

Discussion. ppDBSCAN consistently achieves the highest scores and is able handle different shapes, and non-linear clusters well. PCA and OPT achieve a relatively good clustering quality on eight out of nine datasets, but they (completely) fail on the Dense dataset. The K-Means and Mean-shift protocols have comparable clustering quality that heavily varies between different datasets. The K-means-based protocols can only cluster very specific datasets that do not contain non-convexly shaped and non linearly-separable clusters. HE-Meanshift tends to have large standard deviations which indicate a strong dependency on dust initialization. However, the highest score achieved by HE-Meanshift is comparable to that of plaintext Mean-shift which indicates that the modifications introduced for its HE-friendly computation do not decrease accuracy. In contrast, MPC-KMeans has a small standard deviation and achieves a similar clustering quality to KMeans++, which shows that the randomness used for centroid initialization has a smaller impact on final output.

Security Model w.r.t Scenario. All four works are in the static semi-honest security model i.e., the adversary can corrupt some of the parties at the onset of the computation and correctly follows the protocol description, but attempts to learn information about the private inputs of the honest parties. MPC-KMeans, PCA/OPT, and ppDBSCAN consider the outsourced two-party computation setting where multiple data owners secret share their input among two non-colluding servers to privately cluster the dataset. In contrast, in HE-Meanshift, a *single* data owner outsources its computation to a *single* server.

MPC-KMeans [7] and PCA/OPT [9] provide a formal proof of security by using a *simulator* which generates a view that is indistinguishable from a real protocol execution given the party's input and output. The security of HE-Meanshift [5] follows directly from the security of the used CKKS encryption scheme since only the input and final output are sent. We note that the recent attack on the CKKS scheme by Li and Micciancio [68] does not affect the security of HE-Meanshift, as discussed by Cheon et al. [69]. Similarly, the security of ppDBSCAN [8] follows directly from the security of the employed secure two-party computation techniques, specifically garbled circuit [70] and secret sharing [71].

Leakage from Outputs. The information leaked from the clustering *output* is not captured in the security definition. HE-Meanshift outputs the cluster labels for every record in the dataset. However, this is not a privacy concern since the protocol is intended to be used in the outsourced single-server computation setting where the entire dataset is known to the client. MPC-KMeans and ppDBSCAN can be adapted to output either the cluster centroids or cluster labels. MPC-KMeans also outputs the number of iterations for the clustering to converge which is related to the distribution of the underlying dataset. The PCA/OPT algorithms output a *point-agnostic dendrogram* in addition to the cluster centroids. The point-agnostic dendrogram is intended to be a privacy-preserving variant of the dendrogram output by a plaintext HC algorithm since the latter provides the complete merging history which leaks information in a setting with multiple data owners. The point-agnostic dendrogram is computed by first applying a random and private permutation on the input records to fuzz the merging history and by retaining the metadata of only sufficiently large clusters. Intuitively, this allows obtaining useful metadata akin to the plaintext computation while still preserving privacy.

3.1 Efficiency

Asymptotic Analysis. First, we compare the asymptotic runtime, communication, and round complexity of the four investigated private clustering protocols and depict the results in Tab. 2. Asymptotically, MPC-KMeans is the most efficient with respect to communication and runtime in terms of dataset size N, input records' dimension d and number of clusters K.

Protocol	Runtime	Communication	Rounds
MPC-KMeans [7]	$\Theta(NK(d+\ell)t)$	$\Theta\left(NK(d\ell^2+\ell\kappa)t\right)$	$\Theta(\lceil \log K \rceil t)$
HE-Meanshift [5]	$\Theta\left((NK_d d^2 t)/(N_c \log d)\right)$	$\Theta(NdK_d\kappa)$	2
PCA [9]	$\Theta(N^3\lambda)$	$\Theta(N^3\lambda\kappa)$	$\Theta(N^2)$
OPT [9]	$\Theta(N^2(\lambda+d))$	$\Theta(N^2(\lambda\kappa+\kappa_{\text{pub}}))$	$\Theta(N^2)$
ppDBSCAN [8]	$\Theta(N^2(N+d))$	$\Theta(N^2\ell\kappa)$	$\mathcal{O}(N^3)$

Table 2: Asymptotic complexity of the private clustering protocols. N is the dataset size, d is the dimension, ℓ is the bitlength of the data records, K is the number of clusters, K_d is the number of dusts used in HE-Meanshift, $\kappa = 128$, $\lambda = 40$, N_c is the number of plaintext slots in CKKS, $\kappa_{\text{pub}} = 2048$.

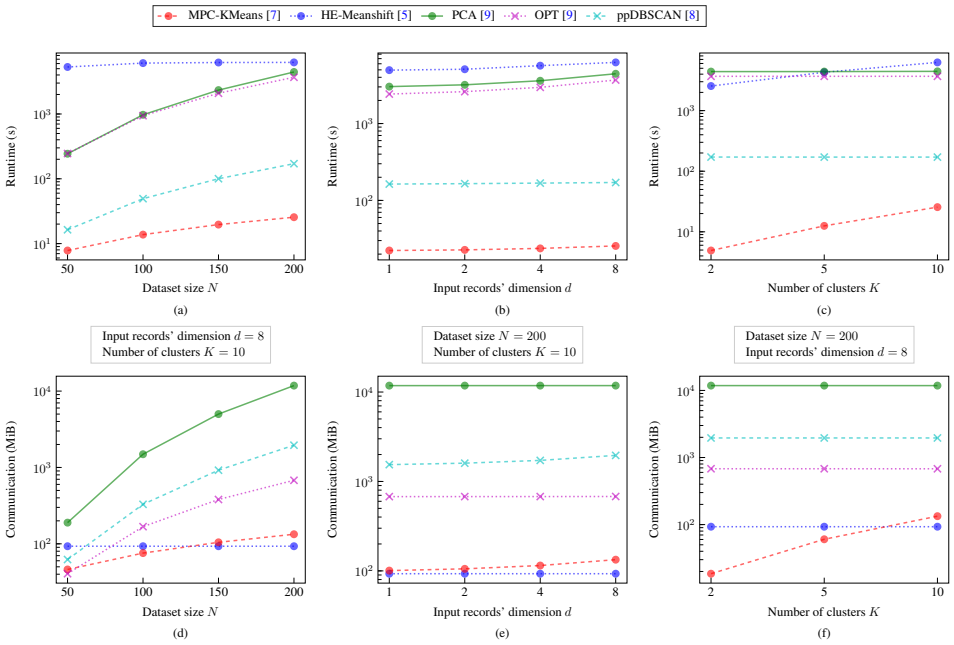

Figure 1: **LAN runtimes** in seconds (top row) and **communication** in MiB (bottom row) of the fully-private clustering protocols for varying dataset size N, input records' dimension d, number of clusters K, and bitlength $\ell = 32$. In (a) and (d) $d = 8$ and $K = 10$, in (b) and (e) $N = 200$ and $K = 10$, and in (c) and (f) $N = 200$ and $d = 8$

Communication. We plot the communication costs in the bottom rows of Fig. 1. The communication cost for HE-Meanshift is identical across different datasets (Fig. 1) because the entire dataset can be encrypted in one ciphertext. The communication cost of PCA is $6\times$ higher than that of ppDBSCAN on average while the communication of ppDBSCAN is $2\times$ higher than that of OPT on average. While the communication cost increases linearly in the input records' dimension d for HE-Meanshift, d does not have a significant effect on the communication of MPC-KMeans since the communication during the assignment of input records to clusters is independent of d. This phase has the highest communication complexity and dominates the overall communication cost. The communication costs of PCA/OPT are independent of the input records' dimension d while the communication costs of PCA/OPT and ppDBSCAN are independent of the number of clusters K.

Runtimes. We plot the LAN runtimes in the top row of Fig. 1, all averaged over 10 runs. MPC-KMeans has the lowest runtime and is up to $700\times$ faster than HE-Meanshift over LAN. On the other hand, the runtime of HE-Meanshift is linear in d since it directly affects the number of ciphertexts and the efficiency of the bootstrapping operation. ppDBSCAN's average runtime is $14\times$ higher. However, since ppDBSCAN's runtime is independent of the number of clusters, this gap in runtime diminishes with the increase in number of clusters. OPT and PCA have similar runtimes wich are $15\times$ higher than that of ppDBSCAN on average over LAN, even though OPT has less communication.

References

[1] A. Hegde, H. Möllering, T. Schneider, and H. Yalame, "Sok: Efficient privacy-preserving clustering." PETS, 2021.

[2] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," in *The adaptive web*, 2007.

[3] C. Gentry, *A fully homomorphic encryption scheme*. Stanford University PhD Thesis, 2009.

[4] W. Wu, J. Liu, H. Wang, J. Hao, and M. Xian, "Secure and efficient outsourced K-means clustering using fully homomorphic encryption with ciphertext packing technique," in *TDKE*, 2020.

[5] J. H. Cheon, D. Kim, and J. H. Park, "Towards a practical cluster analysis over encrypted data," in *SAC*, 2019.

[6] D. Demmler, T. Schneider, and M. Zohner, "ABY–a framework for efficient mixed-protocol secure two-party computation," in *NDSS*, 2015.

[7] P. Mohassel, M. Rosulek, and N. Trieu, "Practical privacy-preserving K-means clustering," in *PETS*, 2020.

[8] B. Bozdemir, S. Canard, O. Ermis, H. Möllering, M. Önen, and T. Schneider, "Privacy-preserving density-based clustering," in *ASIACCS*, 2021.

[9] X. Meng, D. Papadopoulos, A. Oprea, and N. Triandopoulos, "Private two-party cluster analysis made formal & scalable," *arXiv:1904.04475v2*, 2019.

[10] J. Vaidya and C. Clifton, "Privacy-preserving K-means clustering over vertically partitioned data," in *SIGKDD*, 2003.

[11] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed K-means clustering over arbitrarily partitioned data," in *SIGKDD*, 2005.

[12] S. Jha, L. Kruger, and P. McDaniel, "Privacy preserving clustering," in *ESORICS*, 2005.

[13] P. Bunn and R. Ostrovsky, "Secure two-party K-means clustering," in *CCS*, 2007.

[14] S. Samet, A. Miri, and L. Orozco-Barbosa, "Privacy preserving K-means clustering in multi-party environment," in *SECRYPT*, 2007.

[15] C. Su, F. Bao, J. Zhou, T. Takagi, and K. Sakurai, "Privacy-preserving two-party K-means clustering via secure approximation," in *AINA*, 2007.

[16] M. C. Doganay, T. B. Pedersen, Y. Saygin, E. Savaş, and A. Levi, "Distributed privacy preserving K-means clustering with additive secret sharing," in *International Workshop on Privacy and Anonymity in Information Society*, 2008.

[17] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Privacy-preserving user clustering in a social network," in *Information Forensics and Security*, 2009.

[18] J. Sakuma and S. Kobayashi, "Large-scale k-means clustering with user-centric privacy-preservation," in *Knowledge and Information Systems*, 2010.

[19] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, "Efficient privacy preserving K-means clustering," in *Pacific-Asia Workshop on Intelligence and Security Informatics*, 2010.

[20] T.-K. Yu, D. Lee, S.-M. Chang, and J. Zhan, "Multi-party K-means clustering with privacy consideration," in *ISPA*, 2010.

[21] M. Beye, Z. Erkin, and R. L. Lagendijk, "Efficient privacy preserving K-means clustering in a three-party setting," in *Information Forensics and Security*, 2011.

[22] Z. Lin and J. W. Jaromczyk, "Privacy preserving two-party K-means clustering over vertically partitioned dataset," in *ISI*, 2011.

[23] S. Patel, S. Garasia, and D. Jinwala, "An efficient approach for privacy preserving distributed K-means clustering based on shamir's secret sharing scheme," in *Trust Management VI*, 2012.

[24] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Privacy-preserving distributed clustering," in *EURASIP Journal on Information Security*, 2013.

[25] S. Patel, V. Patel, and D. Jinwala, "Privacy preserving distributed K-means clustering in malicious model using zero knowledge proof," in *Distributed Computing and Internet Technology*, 2013.

[26] D. Liu, E. Bertino, and X. Yi, "Privacy of outsourced K-means clustering," in *ASIACCS*, 2014.

[27] Y. Wang, "Notes on two fully homomorphic encryption schemes without bootstrapping." Cryptology ePrint Archive, Report 2015/519.

[28] X. Liu, Z. L. Jiang, S. M. Yiu, X. Wang, C. Tan, Y. Li, Z. Liu, Y. Jin, and J. Fang, "Outsourcing two-party privacy preserving K-means clustering protocol in wireless sensor networks," in *MSN*, 2015.

[29] S. J. Patel, D. Punjani, and D. C. Jinwala, "An efficient approach for privacy preserving distributed clustering in semi-honest model using elliptic curve cryptography," *International Journal of Network Security*, 2015.

[30] F.-Y. Rao, B. K. Samanthula, E. Bertino, X. Yi, and D. Liu, "Privacy-preserving and outsourced multi-user K-means clustering," in *CIC*, 2015.

[31] V. Baby and N. S. Chandra, "Distributed threshold K-means clustering for privacy preserving data mining," in *ICACCI*, 2016.

[32] Z. Gheid and Y. Challal, "Efficient and privacy-preserving K-means clustering for big data mining," in *IEEE TrustCom/BigDataSE/ISPA*, 2016.

[33] H. Rong, H. Wang, J. Liu, J. Hao, and M. Xian, "Outsourced k-means clustering over encrypted data under multiple keys in spark framework," in *Security and Privacy in Communication Networks*, 2017.

[34] K. Xing, C. Hu, J. Yu, X. Cheng, and F. Zhang, "Mutual privacy preserving K-means clustering in social participatory sensing," in *TII*, 2017.

[35] A. Jäschke and F. Armknecht, "Unsupervised Machine Learning on Encrypted Data," in *SAC*, 2018.

[36] H. Kim and J. Chang, "A privacy-preserving k-means clustering algorithm using secure comparison protocol and density-based center point selection," in *International Conference on Cloud Computing*, 2018.

[37] Y. Cai and C. Tang, "Privacy of outsourced two-party k-means clustering," *Concurrency and Computation: Practice and Experience*, 2019.

[38] J. Yuan and Y. Tian, "Practical privacy-preserving MapReduce based K-means clustering over large-ccale dataset," in *TCM*, 2019.

[39] Z. L. Jiang, N. Guo, Y. Jin, J. Lv, Y. Wu, Z. Liu, J. Fang, S. Yiu, and X. Wang, "Efficient two-party privacy-preserving collaborative k-means clustering protocol supporting both storage and computation outsourcing," *Information Sciences*, 2020.

[40] Y. Zou, Z. Zhao, S. Shi, L. Wang, Y. Peng, Y. Ping, and B. Wang, "Highly secure privacy-preserving outsourced k-means clustering under multiple keys in cloud computing," in *Security and Communication Networks*, 2020.

[41] K.-P. Lin, "Privacy-preserving kernel K-means clustering outsourcing with random transformation," *Knowledge and Information Systems*, 2016.

[42] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "PPHOPCM: privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing," *IEEE Transactions on Big Data*, 2017.

[43] J. Zhan, "Privacy preserving K-medoids clustering," in *SMC*, 2007.

[44] S. K. Dash, D. P. Mishra, R. Mishra, and S. Dash, "Privacy preserving K-medoids clustering: An approach towards securing data in mobile cloud architecture," in *Conference on Computational Science, Engineering and Information Technology*, 2012.

[45] X. Lin, C. Clifton, and M. Zhu, "Privacy-preserving clustering with distributed EM mixture modeling," in *Knowledge and Information Systems*, 2005.

[46] M. Hamidi, M. Sheikhalishahi, and F. Martinelli, "Privacy preserving Expectation Maximization (EM) clustering construction," in *DCAI*, 2019.

[47] X. Zhu, M. Liu, and M. Xie, "Privacy-preserving affinity propagation clustering over vertically partitioned data," in *International Conference on Intelligent Networking and Collaborative Systems*, 2012.

[48] H. Keller, H. Möllering, T. Schneider, and H. Yalame, "Balancing quality and efficiency in private clustering with affinity propagation," in *SECRYPT*, 2021.

[49] A. Amirbekyan and V. Estivill-Castro, "Privacy preserving DBSCAN for vertically partitioned data," in *Intelligence and Security Informatics*, 2006.

[50] K. A. Kumar and C. P. Rangan, "Privacy preserving DBSCAN algorithm for clustering," in *Advanced Data Mining and Applications*, 2007.

[51] W.-j. Xu, L.-s. Huang, Y.-l. Luo, Y.-f. Yao, and W. Jing, "Protocols for privacy-preserving DBSCAN clustering," in *International Journal of Security and Its Applications*, 2007.

[52] D. Jiang, A. Xue, S. Ju, W. Chen, and H. Ma, "Privacy-preserving DBSCAN on horizontally partitioned data," in *International Symposium on IT in Medicine and Education*, 2008.

[53] J. Liu, L. Xiong, J. Luo, and J. Z. Huang, "Privacy preserving distributed DBSCAN clustering," in *Transactions on Data Privacy*, 2013.

[54] S. Zahur and D. Evans, "Circuit structures for improving efficiency of security and privacy tools," in *IEEE S&P*, 2013.

[55] I. V. Anikin and R. M. Gazimov, "Privacy preserving DBSCAN clustering algorithm for vertically partitioned data in distributed systems," in *International Siberian Conference on Control and Communications*, 2017.

[56] M. S. Rahman, A. Basu, and S. Kiyomoto, "Towards outsourced privacy-preserving multiparty DBSCAN," in *PRDC*, 2017.

[57] N. Almutairi, F. Coenen, and K. Dures, "Secure third party data clustering using ϕ data: Multi-user order preserving encryption and super secure chain distance matrices," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 2018.

[58] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A new privacy-preserving distributed K-clustering algorithm," in *SDM*, 2006.

[59] A. İnan, S. V. Kaya, Y. Saygın, E. Savaş, A. A. Hintoğlu, and A. Levi, "Privacy preserving clustering on horizontally partitioned data," in *TDKE*, 2007.

[60] G. Jagannathan, K. Pillaipakkamnatt, R. Wright, and D. Umano, "Communication-efficient privacy-preserving clustering," in *Transactions on Data Privacy*, 2010.

[61] I. De and A. Tripathy, "A secure two party hierarchical clustering approach for vertically partitioned data set with accuracy measure," in *Recent Advances in Intelligent Informatics*, 2014.

[62] M. Sheikhalishahi and F. Martinelli, "Privacy preserving clustering over horizontal and vertical partitioned data," in *Symposium on Computers and Communications*, 2017.

[63] P. K. Prasad and C. P. Rangan, "Privacy preserving birch algorithm for clustering over vertically partitioned databases," in *Workshop on Secure Data Management*, 2006.

[64] K. Prasad and P. Rangan, "Privacy preserving birch algorithm for clustering over arbitrarily partitioned databases," *ADMA*, 2007.

[65] A. Ultsch, "Clustering with SOM," in *Workshop on Self-Organizing Maps*, 2005.

[66] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," in *Fuzzy Sets and Systems*, 2010.

[67] M. Gagolewski, "Benchmark suite for clustering algorithms version 1," 2020, https://github.com/gagolews/clustering_benchmarks_v1.

[68] B. Li and D. Micciancio, "On the security of homomorphic encryption on approximate numbers," Cryptology ePrint Archive, Report 2020/1533.

[69] J. H. Cheon, S. Hong, and D. Kim, "Remark on the security of CKKS scheme in practice," Cryptology ePrint Archive, Report 2020/1581.

[70] A. C.-C. Yao, "How to generate and exchange secrets," in *FOCS*, 1986.

[71] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *STOC*, 1987.