STM JOURNALS

Journal of
Network Security

ISSN: 2395-6739 (Online)
ISSN: 2321-8517 (Print)
Volume 12, Issue 1, 2024
January–April
DOI (Journal): 10.37591/JoNS

http://computerjournals.stmjournals.in/index.php/JoNS/index

Review JoNS

Advances in Data Security in Cryptography

Mohit Shrivastava[1], Kanchan Mishra[2,*]

Abstract

In the ultra-modern period, evaluation of networking and wireless networks within information and communication technology has brought many changes to deal with this technology using internet, growing strongly over the past several decades, data security has come a main concern for anyone connected to the web. Data security ensures that our data can only be accessed by authorized recipients and prevents any unauthorized access or alteration of the data. We punctuate the strengths and sins of state-of-the-art ways, and conclude that, while no single approach is likely to approach as a tableware pellet. Therefore, the key is to combine different tools and software approaches to work in confluence using partitioned computing, where a computation is resolved across different cryptographic ways precisely, so as not to compromise security. We punctuate some recent work in that direction. The significant volume, rapid pace, and diverse nature of this data pose a security risk. Given the constraints of IoT devices such as low power and limited computational speed, traditional encryption methods like DES, 3DES, and AES are too complex for implementation. Therefore, there is a necessity to create lightweight encryption algorithms tailored for IoT devices to ensure secure communication and data transmission within IoT environments. Cryptographic and steganographic techniques are employed to safeguard data transmitted over the internet.

Keywords: Cryptography, security, encryption, decryption, big data

INTRODUCTION

With cloud emerging as the dominant computing platform, secure data outsourcing, originally described in significant data management challenge [1–10]. Initial work on secure computing focused extensively on encryption techniques to allow operations to execute on the cloud. With the very fast growth of internet technology, our lives are steadily led into a virtual world. We are going to live in an imaginary space.

In the virtual realm, individuals have the freedom to engage in various activities such as shopping, chatting, and working. Internet of things is an emerging domain which will be the main reason for generating the Big Data. Big Data is also an emerging field and the latest topic of interest. IoT devices are increasing day by day, and the data being generated by them also increases as shown in Figure 1.

***Author for Correspondence**
Kanchan Mishra
E-mail: kanchanmishra778@gmail.com

[1]Head, Department of Computer and Science, Vipra Kala, Vanijya Avam Sharirik Shiksha Mahavidyalaya, Raipur, Chhattisgarh, India
[2]Assistant Professor, Department of Mathematics, Vipra Kala, Vanijya Avam Sharirik Shiksha Mahavidyalaya, Raipur, Chhattisgarh, India

Received Date: January 25, 2024
Accepted Date: March 14, 2024
Published Date: April 04, 2024

Citation: Mohit Shrivastava, Kanchan Mishra. Advances in Data Security in Cryptography. Journal of Network Security. 2024; 12(1): 1–7p.

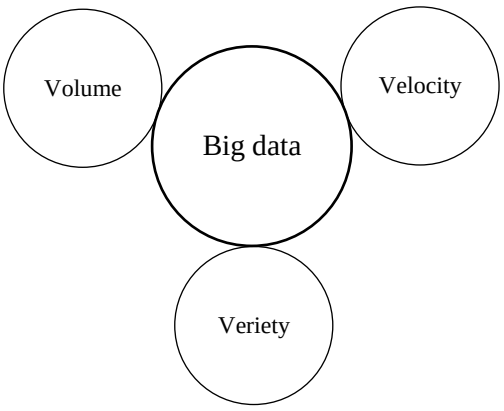

Figure 1. Big data flow graph.

LITERATURE REVIEW

1. *Saini and Susan [11]:* It has been observed that network and computer security represent a rapidly evolving field within computer science, where the pursuit of secure computing is an ongoing challenge. Courses in computer security primarily emphasize on algorithmic and mathematical elements such as hashing methods and encryption. As hackers devise new ways to breach network systems, educational programs adapt by introducing courses that address the latest attack methods. However, these attack techniques quickly become obsolete as new security software respond to them. The terminology and techniques in security are continually evolving, leading to the emergence of new skills and practices in various domains such as business operations, network optimization, security architecture, and legal frameworks.
2. In a study concerning network security and cryptography, Tayal highlighted the increasing volume of data generated daily by global organizations due to the rise of social networks and e-commerce applications [12]. Consequently, ensuring secure data transfer over the internet has become a significant concern. The growing number of internet users underscores the importance of employing cryptography methods to address this issue. The paper offers a summary of different approaches employed by networks to bolster security, including cryptography.
3. *Qadir and Varol [13]:* The research focused on symmetric encryption, utilized for encrypting specific text or speech. In this investigation, the data intended for encryption undergoes initial conversion into an encapsulated cipher format, rendering it indecipherable by a cipher algorithm.
4. *Chachapara and Bhadlawala [14]:* They explored secure data sharing through cryptography in cloud computing, presenting a framework that employs cryptographic algorithms such as RSA and AES. Among these algorithms, AES is identified as the most robust cryptographic method. Within this framework, cloud users have the capability to generate keys tailored for various users, each with different levels of permissions for accessing their files.
5. *Orman [15]:* The author noted that extensive debates and advancements are ongoing in the field of cryptography. They emphasized the crucial role of hash functions in cryptography, which assign a unique numerical value to each piece of data. As vulnerabilities in MD5 became apparent over time, it sparked uncertainty regarding the appropriate design of hash functions.
6. *Gennaro et al. [16]:* They explored the concept of randomness in cryptography, highlighting that a random process is characterized by unpredictable outcomes. The discussion emphasized the significance of randomness in cryptography as it enables the generation of information that remains inaccessible to adversaries, preventing them from learning or predicting it.
7. *Preneel et al. [17]:* The authors showcased the application of cryptography and information security in the aftermath of the Snowden revelations, addressing issues such as mass surveillance and the vulnerability of ICT systems. Additionally, they examined various methods through which advanced attackers can circumvent or weaken cryptographic measures.

BACKGROUND OF SECURITY TECHNIQUES

Steganography

"Steganography derived from two Greek words 'steganos' which means either secret or covered and 'graphein' which means writing or drawing". Which shows that steganography means secret notation, secret delineation, covered notation, or covered delineation. About 2000 times agone, the Greek used this fashion to transfer secret dispatches [18–20]. Steganography means hiding a secret communication within another communication. Steganography is used to hide secret data into the non-secret data. The train in which the secret data is concealed is called the carrier. After concealing the secret data, the modified carrier looks like an original carrier. Some swish carriers are images, audio and video lines. In steganography, we do not reckon the data, we simply conceal our data within image, audio or video lines.

Cryptography

Cryptography is used to make security. Cryptography is an important tool to cover the data. Our computer word is secured by cryptographic hash function. When we shoot a dispatch, it is also secured by cryptography fashion SSL [20–30].

Cryptography concerns with confidentiality (No one can pierce the data except the person to whom it may concern), integrity (No one can change or modify the data), and authenticate (sender and receiver conform their identity). Cryptography is used to store and transfer the data in such a form that only sender and receiver can understand it or reuse it. A meddler cannot pierce or understand that data. Cryptography depends upon the algorithm and the key. Two main terms used in cryptography are encryption and decryption. Encryption is a process to convert a plain textbook into cipher textbook and decryption is the process to convert cipher textbook to plain textbook [31–40].

Featherlight Cryptography in Internet of Effects machines connected to machines or machines connected to mortal to communicate or to shoot data. IoT bias have lower computational power, so we need to design featherlight encryption ways to cipher the data. Featherlight Cryptography is used in a constrained terrain like RFID markers, healthcare bias, and detectors. While developing Lightweight Cryptography ways, we must have to take into consideration the software and tackle specification.

CRYPTOGRAPHIC SECURE DATA PROCESSING

Inimical and security models for the Cloud: Cryptographic ways are designed to deal with different types of inimical public shadows, videlicet honest-but-curious (HBC) and vicious shadows. An HBC inimical pall, which is considered extensively in numerous cryptographic algorithms, stores an outsourced dataset without tampering, rightly computes assigned tasks, and returns answers; still, it may exploit lateral knowledge.

To acquire significant insights into sensitive data, investigators focus on querying prosecution, background information, and the scale of the affair. A vicious adversary may diverge from the algorithm and may execute a task that he/she wishes (e.g., delete tuples from the relation). To ensure security in HBC and vicious inimical models, IND-CKA and IND-CKA2 are star security parcels, independently. Further, to deal with securely streamlining and fitting data, forward and backward security parcels were introduced. In this study, we discuss different inimical models and security parcels. Secret-Participating-grounded Data Outsourcing Secret-Participating was constructed singly. In using secret-sharing, the database (DB) proprietor divides a secret value into different fractions, called shares, and sends each share to a set of non-communicating actors/waiters. These waiters cannot know the secret value until they collect shares. Note that the threshold value c is often set to be greater than c to accommodate potentially malicious adversaries which may attempt to alter the value of their shares. We discuss fresh hypotheticals about the inimical model when using secret-sharing, e.g., the adversary cannot machinate with all (or conceivably the maturity of) the waiters. Also, the adversary cannot overhear on a maturity of communication channels between the DB proprietor and the waiters. Note that if the adversary could either machinate with or successfully overhear on the communication channels between the maturity of waiters and the DB proprietor, the secret-sharing fashion will not apply [41–50].

Order-conserving secret-sharing (OP-SS), maintains the order of the values in secret-shares too. Due to maintaining the ordering of values, chancing records with outside/minimum values using OP-SS is trivial, while revealing ordering information to the adversary. In 2006, [37] introduced the first work for data outsourcing using SSS and OP-SS for executing sum, outside, and minimal queries. Another paper by [38]. using OP-SS for aggregation queries requires the DB proprietor to retain each polynomial, which was used to produce database shares. The similar approach, suggesting Secret Sharing Scheme (SSS)-based sum and average queries; still, they bear the DB proprietor to retain tuple-ids of qualifying tuples introduced a new approach for searching over the secret-participated data without taking help from the DB proprietor, utilizing the string-matching operation over the shares at the garçon by applying a MapReduce job. In short, these results offer a limited form of selection or aggregation queries, but overload the DB proprietor (by storing enough data related to polynomials and completely sharing in a query prosecution), are insecure due to OP-SS, or reveal access-patterns. Recent work, OBSCURE eliminates all similar limitations and provides a completely secure and effective result

for enforcing aggregation queries with selections. OBSCURE exploits OP-SS, while OP-SS is not secure (it is prone to background knowledge attacks, for case) [51, 52].

The way OBSCURE cleverly uses OP-SS, it prevents similar attacks by meetly partitioning data, while still being suitable to exploit OP-SS for effectiveness. SMCQL and Conclave are two systems that allow executing SQL queries among different DB possessors, while pushing most corridor of the calculation in cleartext. Recent performances of Intel CPUs introduced SGX, a collection of microarchitectural mechanisms aimed to cover third-party pall operations from the software mound of an untrusted system. SGX allows us to produce a small, trusted prosecution terrain that is insulated and defended from the rest of the system. For illustration, in the pall, SGX protects the calculation from the operating system controlled by the tenant, but vulnerable to multitudinous operations, system level attacks, and the hypervisor controlled by the pall. In addition to guarding against software attacks, SGX provides encryption of enclave's memory having law and data and integrity is defended by the CPU, when the data leaves the last position of the caching scale. This protects SGX operations from tackle attacks like memory poking. While running on the waiters controlled by the call, from a trust point of view, SGX enclaves are effectively controlled by the customer. 1799 Failing of the being SGX armature. Unfortunately, being executions of SGX are prone to a range of side-channel attacks that exploit one of the microarchitectural factors of the CPU, e.g., branch target buffers pattern-history table caches, DRAM row buffer, runner-tables, runner-fault exceptions, instructors, and academic prosecution capabilities to exfiltrate sensitive data and occasionally the entire memory of the enclave. Runner-fault attacks calculate on the operating system that is under control of the bushwhacker to exfiltrate sensitive data from the enclave by driving runner-faults and tracking enclave's memory accesses at the granularity of memory runners. Branch shadowing attacks allow reconstructing control inflow inside the enclave, and hence, secret data by covering all taken branches through a side-channel in the branch vaticination unit. Cache based side-channel attacks allow bushwhackers to trace all memory accesses within the enclave. Foreshadow and Meltdown-type attacks on SGX allow complete access to enclave's memory and registers. Unexpectedly, the side-channel attacks allow a bushwhacker to reconstruct a significant bit of the sensitive dataset, frequently gaining complete access to it in cleartext [53–59].

To illustrate the power of side-channel attacks, we bandy several recent case studies that is cryptographic keys, graphical images, and sensitive genomic data from the enclave in a practical and realistic script). Data Processing using SGX. Several recent systems surfaced to give secure data processing in the SGX terrain. We give a literal perspective on the problem starting with TrustedDB and Cipherbase, the systems that were first to use secure tackle and initiated the field of secure tackle-grounded data processing. We also bandy recent systems, similar as Opaque Enclave, DBS, Health DB, M2R, and VC3, that are erected to influence SGX. We give an overview of types of queries these systems support, and bandy their advantages and limitations, and whether they give practical security in the face of important side-channel attacks. We also bandy an ongoing microarchitectural work aimed to address side-channel attacks in the coming generation of chips, and possible limitations of tackle approaches. Eventually, we give an overview of possible system-position defenses and algorithmic approaches, furnishing a secure terrain with SGX. Data Partitioning-grounded outsourcing being a secure ways do not gauge to large-sized datasets. For illustration, on TPC-H Line-Item table, executing a simple selection query took 1051 sec on 1 M rows using secret-sharing-grounded Jana and 89 sec using SGX based Opaque on 6 M rows. Partitioned computing involves segregating data into sensitive and non-sensitive categories, offering potential advantages by (i) eliminating costly cryptographic operations on non-sensitive data, and (ii) enabling query processing on non-sensitive data to utilize indicators. Similar indicators (that cannot be fluently supported alongside encryption-grounded mechanisms in a non-interactive setting) are a crucial medium for effective query processing in traditional database systems. Partitioned computing over sensitive/non-sensitive data has been considered, especially, in cold-blooded shadows (e.g., Hybr Ex Sedic Prometheus Tagged-Map Reduce SEMROD and where sensitive data stays at a private pall and only clear text non-sensitive data stays at

a public pall still, do not generalize to partitioned computing in the public pall setting (where sensitive data is stored cryptographically secure and non-sensitive data resides in clear text). Recent work generalizes the partitioned computing approach at the pall. We bandy a new security challenge due to contemporaneous prosecution of queries on the translated (sensitive) dataset and on the plaintext (non-sensitive) datasets, and also, show the need to new security description, called partitioned data security. We punctuate the significance of partitioned calculation at the pall by illustrating selection query prosecution.

GOAL OF THE TUTORIAL

Outgrowth, intended followership, and duration. This study provides a check on data security and sequestration by preamble reducing the state-of-the-art results from the security literature (especially, secret-sharing-, secure tackle-, and data partitioning-grounded ways) that are particularly applicable for databases. Experimenters, scholars, inventors, interpreters interested in data security and sequestration should be served. We covered content for different cults, as 10 freshman, 40 intermediate, 50 advanced. The duration was 3 h. We show that the being data outsourcing ways and systems are not enough if we aim for systems that operate concurrently.

1. Are effective and general enough (i.e., support significant corridor of SQL) to be practical.
2. Offer sustainable security from the stoner's perspective. Below, we give open questions in different directions. We may suppose about how SSS-grounded results can be developed for large-sized datasets, how they can support complex SQL queries, and how one can minimize the number of shadows while doing all operations at the pall. In the face of multitudinous side channel attacks, naive relinquishment of SGX does not give any practical protection. Some of the challenges will be answered by unborn tackle but some will bear algorithmic and compiler-position results to support effective memory and branch-unconscious calculations. To make partition computing further practical, we need to find ways to execute queries (e.g., join and nested queries) using partitioned computing. Another open problem is in changing how one can classify the data into sensitive and non-sensitive data. Another open problem could be in finding which types of cryptographic ways can be supported by partition computing.

CONCULSION

Data security plays a crucial role in safeguarding digital information from unauthorized access, alteration, or deletion. As the volume of sensitive data stored and transmitted online continues to grow, the significance of data security has escalated. The challenges and advantages associated with technology in terms of data security and privacy are closely intertwined. While technology introduces new threats and vulnerabilities, it also offers contemporary solutions. Through adopting effective strategies, staying abreast of emerging technologies, and implementing robust security protocols, we can ensure the resilient protection of data and privacy in our increasingly digitalized environment. As technology progresses, collaboration among individuals, organizations, and policymakers is essential to address challenges and capitalize on opportunities effectively.

REFERENCES

1. Hacigümüş H, Iyer B, Li C, Mehrotra S. Executing SQL over encrypted data in the database-service-provider model. In Proceedings of the 2002 ACM SIGMOD international conference on Management of data. 2002 Jun 3; 216–227.
2. Agrawal R, Kiernan J, Srikant R, Xu Y. Order preserving encryption for numeric data. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data. 2004 Jun 13; 563–574.
3. Bellare M, Boldyreva A, O'Neill A. Deterministic and efficiently searchable encryption. In Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27. Berlin Heidelberg: Springer; 2007; 535–552.
4. Goldwasser S, *et al*. Probabilistic encryption. J Comput Syst Sci. 1984; 28(2): 270–299.

5. Gentry C. A fully homomorphic encryption scheme. PhD thesis. California: Stanford University; 2009.
6. Song DX, *et al*. Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE Symposium on Security and Privacy (S&P 2000*). 2000; 44–55.
7. Curtmola R, *et al.* Searchable symmetric encryption: Improved definitions and efficient constructions. Journal of Computer Security (JCS). 2011; 19(5): 895–934.
8. Popa RA, *et al*. CryptDB: processing queries on an encrypted database. Commun ACM. 2012; 55(9): 103–111.
9. Tu S, *et al*. Processing analytical queries over encrypted data. Proceedings of the VLDB Endowment (PVLDB). 2013; 6(5): 289–300.
10. Bajaj S, *et al*. Correctdb: SQL engine with practical query authentication. Proceedings of the VLDB Endowment (PVLDB). 2013; 6(7): 529–540.
11. Saini M, Susan S. Deep transfer with minority data augmentation for imbalanced breast cancer dataset. Appl Soft Comput. 2020 Dec 1; 97: 106759.
12. Tayal S. Smart City Ranking Based on Big Data. Doctoral dissertation. Delhi: Delhi Technological University 2020.
13. Qadir AM, Varol N. A review paper on cryptography. In 2019 IEEE 7th international symposium on digital forensics and security (ISDFS). 2019 Jun 10; 1–6.
14. Chachapara K, Bhadlawala S. Secure sharing with cryptography in cloud computing. In 2013 IEEE Nirma University International Conference on Engineering (NUiCONE). 2013 Nov 28; 1–3.
15. Orman H. Creating an Open Community of Cryptographers. In Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman. ACM; 2022 Aug 24; 185–212.
16. Gennaro R, Karger P, Matyas S, Peyravian M, Roginsky A, Safford D, Willett M, Zunic N. Two-phase cryptographic key recovery system. Comput Secur. 1997 Jan 1; 16(6): 481–506.
17. Yoshizawa T, Singelée D, Muehlberg JT, Delbruel S, Taherkordi A, Hughes D, Preneel B. A survey of security and privacy issues in v2x communication systems. ACM Comput Surv. 2023 Jan 13; 55(9): 1–36.
18. Egorov M, Wilkison M. ZeroDB white paper. arXiv:1602.07168 [cs.CR]. 2016. CoRR, vol. abs/1602.07168, 2016.
19. Tetali SD, Lesani M, Majumdar R, Millstein T. MrCrypt: Static analysis for secure cloud computations. InProceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications 2013 Oct 29 (pp. 271-286).
20. AWS. Amazon Aurora. [Online]. available at: https://aws.amazon.com/rds/aurora/.
21. MariaDB. [Online]. Available at: https://mariadb.com/.
22. Bajaj S, *et al*. TrustedDB: A trusted hardware-based database with privacy and data confidentiality. IEEE *Trans Knowl Data Eng* (TKDE). 2014 Mar; 26(3): 752–765.
23. Arasu A, *et al*. Orthogonal security with cipherbase. In 6th Biennial Conference on Innovative Data Systems Research (CIDR). 2013.
24. Sion R. Secure data outsourcing. In 33rd International Conference on Very Large Data Bases (VLDB). 2007; 1431–1432.
25. Arasu A, *et al*. Querying encrypted data. In *2013 IEEE 29th International Conference on Data Engineering (*ICDE). 2013; 1262–1263.
26. Arasu A, *et al*. Querying encrypted data. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. 2014; 1259–1261.
27. Costan V, *et al.* Intel SGX explained. IACR ePrint Archive. 2016.
28. Agrawal D, *et al.* Secure and privacy-preserving database services in the cloud. In *2013 IEEE 29th International Conference on Data Engineering (*ICDE). 2013; 1268–1271.
29. Sahin C, *et al*. Data security and privacy for outsourced data in the cloud. In *2018 IEEE 34th International Conference on Data Engineering (*ICDE). 2018; 1731–1734.
30. SHAttered. [Online]. Shattered.io. 2017. Available from: https://shattered.io/
31. TinyURL. (2024). URL Shortener, Branded Short Links & Analytics. [Online]. Tinyurl. Available from: https://tinyurl.com/app

32. Shamir A. How to share a secret. Commun ACM. 1979; 22(11): 612–613.
33. Beimel A. Secret-sharing schemes: A survey. In International Conference on Coding and Cryptology (IWCC). 2011; 11–46.
34. Gilboa N, Ishai Y. Distributed point functions and their applications. In EUROCRYPT. 2014; 640–658.
35. Boyle E, Gilboa N, Ishai Y. Function secret sharing,. In EUROCRYPT. 2015; 337–367.
36. Dolev S, *et al.* Accumulating automata and cascaded equations automata for communication-less information theoretically secure multiparty computation. Theor Comput Sci (TCS). 2019; 795: 81–99.
37. Dolev S, *et al*. Privacy-preserving secret shared computations using MapReduce. IEEE *Trans Depend Secur Comput* (TDSC). 2021; 18(4): 1645–1666.
38. Gupta P, *et al.* Obscure: Information-theoretic oblivious and verifiable aggregation queries. Proceedings of the VLDB (PVLDB). 2019; 12(9): 1030–1043.
39. Bater J, *et al*. SMCQL: secure query processing for private data networks. Proceedings of the VLDB (PVLDB). 2017; 10(6): 673–684.
40. Volgushev N, *et al.* Conclave: secure multi-party computation on big data. In EuroSys. 2019; 3:1–3:18.
41. Wang F, *et al*. Splinter: Practical private queries on public data. In NSDI. 2017; 299–313.
42. Stealth Pulsar. [Online]. Available at: http://www.stealthsoftwareinc.com/.
43. Archer DW, *et al.* From keys to databases - real-world applications of secure multi-party computation. IACR Cryptology ePrint. 2018.
44. Emekci F, *et al.* Privacy preserving query processing using third parties. In Proceedings of the 22nd International Conference on Data Engineering (ICDE). 2006; 2
45. Emekci F, *et al*. Dividing secrets to secure data outsourcing. Inf Sci. 2014; 263: 198–210.
46. Xiang T, *et al*. Processing secure, verifiable and efficient SQL over outsourced database. Inf Sci. 2016; 348: 163–178.
47. Zheng W, *et al.* Opaque: An oblivious and encrypted distributed analytics platform. In NSDI. 2017; 283–298.
48. Priebe C, *et al*. Enclavedb: A secure database using SGX. In IEEE *Symposium on Security and Privacy (*SP). 2018; 264–278.
49. Vinayagamurthy D, *et al*. StealthDB: a scalable encrypted database with full SQL query support. PoPETs. 2019; 370–388.
50. Dinh TTA, *et al.* M2R: enabling stronger privacy in mapreduce computation. In USENIX. 2015; 447–462.
51. Schuster F, *et al.* VC3: trustworthy data analytics in the cloud using SGX. In IEEE *Symposium on Security and Privacy* (SP). 2015; 38–54.
52. Brasser F, *et al*. Software grand exposure: SGX cache attacks are practical. In WOOT Proceedings of the 11th USENIX Conference on Offensive Technologies. 2017; 1–11.
53. Gotzfried J, *et al*. Cache attacks on Intel SGX. In EuroSec, Proceedings of the 10th European Workshop on Systems Security. 2017; 2(6p).
54. Moghimi A, *et al*. Cachezoom: How SGX amplifies the power of cache attacks. In Cryptographic Hardware and Embedded Systems (CHES). 2017; 69–90.
55. Lee S, *et al.* Inferring fine-grained control flow inside SGX enclaves with branch shadowing. In USENIX Security. 2017; 557–574.
56. Hahnel¨ M, *et al.* High-resolution side channels for untrusted operating systems. In USENIX ATC. 2017; 299–312.
57. Schwarz M, *et al*. Malware guard extension: Using sgx to conceal cache attacks. In Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA). 2017; 3–24.
58. Chen G, *et al*. SgxPectre attacks: Stealing intel secrets from SGX enclaves via speculative execution. arXiv preprint. 2018.
59. Costan V, *et al*. Sanctum: Minimal hardware extensions for strong software isolation. In USENIX. 2016; 857–874.

