Practical Security and Privacy for Database Systems

Xi He
University of Waterloo
xi.he@uwaterloo.ca

Jennie Rogers
Northwestern University
jennie@northwestern.edu

Johes Bater
Duke University
jsb108@cs.duke.edu

Ashwin Machanavajjhala
Duke University
ashwin@cs.duke.edu

Chenghong Wang
Duke University
chenghong.wang552@duke.edu

Xiao Wang
Northwestern University
wangxiao@northwestern.edu

ABSTRACT

Computing technology has enabled massive digital traces of our personal lives to be collected and stored. These datasets play an important role in numerous real-life applications and research analysis, such as contact tracing for COVID 19, but they contain sensitive information about individuals. When managing these datasets, privacy is usually addressed as an afterthought, engineered on top of a database system optimized for performance and usability. This has led to a plethora of unexpected privacy attacks in the news. Specialized privacy-preserving solutions usually require a group of privacy experts and they are not directly transferable to other domains. There is an urgent need for a general trustworthy database system that offers end-to-end security and privacy guarantees. In this tutorial, we will first describe the security and privacy requirements for database systems in different settings and cover the state-of-the-art tools that achieve these requirements. We will also show challenges in integrating these techniques together and demonstrate the design principles and optimization opportunities for these security and privacy-aware database systems. This is designed to be a three hour tutorial.

CCS CONCEPTS

• **Security and privacy** → **Data anonymization and sanitization**; **Management and querying of encrypted data**; • **Information systems** → *Federated databases.*

KEYWORDS

Privacy; Security; Differential privacy; Secure computation; Trusted execution environment

ACM Reference Format:
Xi He, Jennie Rogers, Johes Bater, Ashwin Machanavajjhala, Chenghong Wang, and Xiao Wang. 2021. Practical Security and Privacy for Database Systems. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD ’21), June 20–25, 2021, Virtual Event, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3448016.3457544

1 TUTORIAL OVERVIEW

Databases are everywhere and their role is increasing in almost every facet of life. Moreover, an enormous amount of sensitive personal data is kept in the cloud or accessed by untrusted third parties, e.g., smart home devices, location data from smartphones. The current approach to protecting sensitive data in these systems has not kept pace with this new reality. New regulations, such as GDPR and proposed laws in the US (CITE), are demonstrating the need to address this challenge in large-scale data management systems, especially those in the cloud. At the same time, in the wake of a parade of scandals like the Facebook-Cambridge Analytica leak in 2018, users are seeking ever-greater assurances that their data will only be used in ways they authorized.

Next generation database systems will need to treat security and privacy as first class citizens in their design. This paradigm shift impacts virtually every aspect of a DBMS from how we access data to how we prepare query results for the user. It will have a profound impact on many core database challenges such as query optimization, storage management, and database operator algorithms.

Right now security and privacy is mostly treated as an add-on in production database systems and they offer simple mechanisms, like access control lists, to protect their data. The security and privacy community has developed a myriad of techniques [22, 37, 50, 77], but they do not address the end-to-end workflow of a DBMS query execution in different environment settings. The deployments of these techniques in a particular environment requires multiple PhD-level specialists and cannot be easily transferred to other settings as they are usually hard-coded. There has been limited work on building end-to-end systems with security and privacy guarantees from the database community [11, 42, 45, 55, 78], but each has its own focus of assurance. However, composing these assurances is a non-monotonic cost model for query optimization. Naive integration of these techniques may even lead to new privacy attacks [40].

In this tutorial, we will present common architectures in this emerging landscape and identify when and where each will need to incorporate security and privacy techniques to satisfy this emerging demand for more rigorous protection of sensitive data in relational-style DBMSs. We will also highlight the challenges of the trade-offs associated with competing solutions in the large-scale database systems and the challenges in composing security and privacy techniques to provide end-to-end guarantees in trustworthy DBMSs. Thus, the tutorial is targeted to general DB researchers and practitioners who would like to learn about privacy and security state-of-the-arts, as well as security and privacy experts who are looking for new research problems in database settings. Our tutorial will cover

SIGMOD ’21, June 20–25, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/3448016.3457544

the theoretical foundations of the key security and privacy building blocks (secure computation, differential privacy, and trusted execution environments) and case studies that integrate these techniques for trustworthy DBMSs.

2 TUTORIAL OUTLINE

Our tutorial will consist of 3 modules (3 hours). The modular organization will allow attendees to choose which parts of the tutorial they might be most interested in. The first module covers 'defining security and privacy requirements' for database systems. This focuses on the background, problem setting and definition. The second module on 'building blocks and toolboxes' presents the state-of-the-art in security and privacy techniques. These first two modules are intended to give general database researchers an overview of the security and privacy landscape in database systems. The second module also provides essential toolboxes for researchers to integrate security and privacy into their own systems. The third module on 'system integration and optimization' provides case studies to show the opportunities and challenges in building practical trustworthy database systems. This module also provides an overview of cutting edge research that may be of interest even to experts.

2.1 Module I: Defining S&P Requirements

In this module, we will first introduce three common *reference architectures* to describe the relationships between database systems and their users and between autonomous databases in a federation. In Figure 1, we see our three reference architectures: (i) a client-server model [26, 47], (ii) an untrusted cloud service provider [2, 5], and (iii) a data federation [8, 74]. Here, an untrusted party denotes a player that intends to do whatever they can – or act maliciously – to gain unauthorized access to private data within a database. Interacting with an untrusted party requires the use of privacy-preserving technologies to guarantee the privacy, security, or integrity of a database's storage and querying facility. For the data federation setting, we also introduce a player that is semi-honest or malicious, shown with a broken padlock. The malicious player acts as before, but we will also introduce techniques for semi-honest players, or ones that will follow a set of protocols faithfully to participate in query processing, but who will try to learn everything they can about private data by observing a query's execution. Semi-honest techniques offer higher performance than full malicious guarantees. These settings are not meant to be exhaustive. They provide a canvas for us to examine the security and privacy challenges associated with these systems.

We will show the importance of ensuring security, privacy, and integrity guarantees in these three settings using real world privacy leakage examples and security attacks on systems [43, 44, 60]. Then we will define the security, privacy, and integrity requirements at every stage of the systems in the reference architecture — before, during, and after the query execution. We will also provide a high-level overview of the state-of-the-art efforts and techniques in achieving each requirement (Table 1). Note that each technique has its own performance, privacy, and utility trade-off. Here, privacy denotes the strength of their guarantees for data protection, performance captures how efficiently the system evaluates queries and utility describes the accuracy of query results and expressiveness of its querying facility. This leads to new decision spaces for trustworthy DBMSs. We will highlight the new optimization problems and challenges ahead.

In this tutorial, we will focus on protecting the security and privacy of data throughout the query lifecycle. We consider how to protect the privacy of a query's private inputs using differential privacy [45, 55, 68]. After that, we will briefly touch on the related challenge of running a secret query over public data using private information retrieval [17, 61]. Lastly, we will examine techniques used to protect query records during query evaluation with software (secure computation) and hardware (trusted execution environments). We will examine how these techniques protect the data from side-channel leakage during query evaluation and from being read directly by an adversary on the computing machine. We will not cover private information retrieval and techniques for integrity in this tutorial in-depth, but we will highlight how they interact with other techniques at the end of the tutorial.

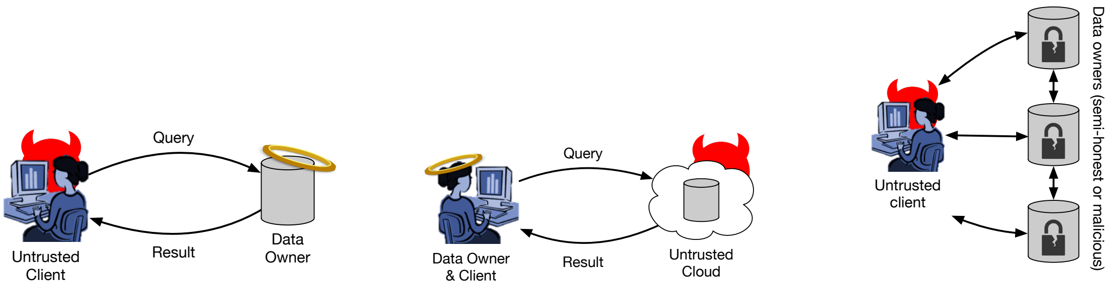
2.2 Module II: Techniques and Toolboxes

2.2.1 *Secure Computation.*

Secure computation allows a set of mutually distrustful parties to jointly compute a function on their inputs without revealing anything beyond the output of the function. Since the invention of secure computation [82], this technology has witnessed significant growth in its practicality. Numerous start-ups based on various secure computation technologies have been founded to use related cryptographic techniques to protect financial information [12], for anonymous reporting of sexual misconduct [64], private auctions [13], and more.

Secure computation protects private data during query evaluation in two ways. First, it protects the confidentiality of the data by encrypting it and the outputs derived from it throughout the query's runtime. Second, the query's evaluation is oblivious, meaning that its instruction traces are independent of its private inputs. In other words, they leak no information about the underlying data via early termination. This property result in a high performance penalty for queries running within secure computation. In practice, their runtime is typically multiple orders of magnitude slower than running the same query insecurely.

Secure computation can be used in the cloud and data federation settings for query evaluation over private data. In the cloud, data owners use secure computation to query their private records using an untrusted service provider [2, 79]. In a data federation, oblivious query processing was researched in [8, 9, 72, 74]. Almost all secure computation protocols consist of following steps: 1) represent the computation as a circuit; 2) execute a secure subprotocol that securely encrypt the input data for evaluation in the circuit; 3) following the topological order of the circuit, evaluate all gates therein. Usually, the evaluation of each gate incurs some computational and communication cost, which becomes significant when the function is complex. Large-scale computation and analysis usually require billions of gates, leaving a huge space for further optimization. More recently, customized MPC protocol using for database operations also started to gain attention where further improvement in speed was obtained. For example, [48] discussed performing join-and-compute more efficiently supporting default values and [57]



(a) Client-server

(b) Cloud service provider

(c) Data federation

Figure 1: Reference architectures for database systems.

Guarantee	Client-server	Cloud service provider	Data federation
Privacy of: Input data	Differential privacy [45, 55]	N/A	Differential privacy [9, 67]
Queries	N/A	Private information retrieval [61]	Private function evaluation [75]
Query evaluation	N/A	Secure computation [8, 74], trusted execution environments [7, 86]	
Integrity of:			
Storage	Authenticated data structures [25]		Blockchain [3, 29]
Query evaluation	Zero-knowledge proofs	Verifiable computation[85], secure [31] computation trusted execution environments [63]	

Table 1: Overview of security and privacy techniques for architectures in Figure 1 with citations of exemplar systems.

supports generic join operations more efficiently using techniques from private set intersection.

Zero-knowledge proofs (ZKP) can be viewed as a special type of secure computation, where only one party (i.e., prover) has the input, and the other party (i.e., verifier) obtains one bit of output indicating if a certain public predicate is true on the prover’s input [32, 33]. ZKP can have a lot of interesting use with databases [84]. For example, the data owner can first publish a digest of the database, which does not reveal any information about its contents but binds the database’s contents. When the data owner receives a query, they will return the result to the client with a proof of its correctness that the client verifies by combining it with the initial digest. This ensure integrity of the query while revealing nothing else about the database except the query result.

This tutorial will focus on covering the application side of secure computation, including basic security definition, and a computation abstraction that incorporates most secure computation protocols. We will covert some basic techniques on how to use secure computation efficiently as well as some recent works in this direction. We will also discuss recent advances in private information retrieval (PIR) [18, 46] as well as their scalable variations [14, 20]. Finally, we will talk about practical considerations, include existing secure computation tools as well as how to incorporate various other techniques in this tutorial.

2.2.2 Differential Privacy. Unlike secure computation that maintains the confidentiality of the input dataset during query execution, differential privacy (DP) [27, 28] offers a guarantee on whether sensitive values in the dataset can be inferred or “reconstructed" from the output of a query. This privacy guarantee is considered the gold standard for ensuring privacy of the input data in most data sharing scenarios and has been adopted by several organizations, including the US Census Bureau [53], and tech companies like Google, Apple, Microsoft and Uber. This guarantee is often achieved via randomization, such as injecting carefully controlled levels of noise into a query’s results. A private dataset begins with a privacy budget defining how much information about the data may be revealed in noisy query results. Each query receives some quantity of the privacy budget. The composition of multiple differentially private algorithms still ensures differential privacy. This is useful for proving the privacy guarantees of complex queries or workloads with many queries over a single dataset [83].

This tutorial will focus on the definition of DP, the basic DP mechanisms for answering a single SQL-like query (e.g. the Laplace mechanism, and the sensitivity analysis of a query plan), and composition properties of DP [28]. We will also survey existing DP frameworks and toolboxes for DP algorithm developments (e.g. ektelo [83], Google DP, IBM diffprivlib), and existing DP systems that support end-to-end private data analysis (e.g. PINQ [55], Flex [42], PrivateSQL [45], Airavat [66]). Last, we will introduce computational relaxations of standard DP, known as computational DP [56], for the cloud setting and the data federation setting, and show the adaptations of the basic DP mechanisms for these settings [9, 58].

2.2.3 Trusted Execution Environment. Trusted Execution Environment (TEE) is a tamper-resistant isolated execution environment [69]. It guarantees the authenticity of all executed code, the integrity of run-time states (such as CPU registers and memory), and confidentiality for their data data and run-time states. In addition, TEEs offer remote attestation, wherein a user queries a trusted third party to verify that an enclave he or she receives is authentic, or has not been tampered with. In contrast to secure computation, TEEs uphold their security guarantees using hardware-based solutions. In the past few years, a number of TEEs have been proposed, including Intel SGX [21] and Apple's "Secure Enclave co-processor" [4]. Most of these solutions are aimed at protecting the security of program execution. Recently, there has been a trend to use TEEs to safeguard database systems, especially to provide secure query processing on outsourced databases. Notable works under this literature include StealthDB [34], Opaque [86], ObliDB [30], Obladi [23], and more.

The privacy guarantees offered by TEEs and secure computation differ in one subtle, but important way: although they protect their input tuples from prying eyes by encrypting their contents for the duration of the query, their execution is not oblivious. This means that branching, loop iteration counts, and other program behavior are observable by the adversary. This leakage can enable an adversary to deduce unauthorized information about a secure program's input data [35, 73, 76]. A TEE-based DBMS can address leaking memory access patterns by doing its I/Os using oblvious memory primitives such as ZeroTrace [70].

In this tutorial, we will review existing TEE-based methods that safeguard the query processing as well as some end-to-end secure databases that use TEEs throughout the data lifecycle. We will also describe the key techniques with which these systems deploy TEEs, and compare them with software-based approaches. Finally, we will discuss the available trusted hardware-centric designs for building secure databases, their limitations and related open research questions.

2.3 Module III: System Integration and Optimization

Each security and privacy technique has its own performance, privacy, and utility trade-off. The integration of these techniques for a database system leads to new challenges and opportunities in the decision space. In this module, we will present case studies to demonstrate the practical challenges in building a trustworthy database system with one or more than one security and privacy guarantees for each reference architecture. Note that integration of various tools is a challenging task. On the one hand, system insight is needed to ensure high performance and scalability, on the other hand, to ensure provable security and privacy, it is important to design the database systems in a principled way. For example, negligence towards composability can cause attacks [36, 59] to systems like CryptDB [62].

For the client-server model, we will illustrate how to handle complex privacy policies for the input data that involves multiple relations and join queries using PrivateSQL [45]. In this setting, the main trade-off is between privacy of the input data and the utility of the query answers. As the true query processing time on the underlying databases also leaks information to the data analyst [38], to avoid this side-channel attack, PrivateSQL first generates differentially private synopses offline. This allows unlimited number of queries answered online over these synopses without further leaking any information about the input data.

For the cloud service provider setting, we will show how to support oblivious query processing for general query workloads using TEE-based DBMS solutions, Opaque [86] and ObliDB [30]. Both systems utilize Intel SGX hardware enclaves for the guarantees of computation integrity and query processing obliviousness to the untrusted cloud service provider. The performance overhead for these guarantees are usually very expensive. To tackle this challenge, fine-grained oblivious operators are proposed together with new optimization rules or cost model. Note that in this cloud service provider setting, when the data owner and the data analyst are different parties, differential privacy can also be used for the protection of the input data [1, 24, 67].

For data federations, we will consider case studies including SMCQL [8], Shrinkwrap [9], and SAQE [10]. These systems aim to offer both input data privacy and oblivious query evaluation. These systems consider the end-to-end privacy measure known as computational differential privacy that composes these privacy guarantees and demonstrates a three-way trade-off between performance, privacy and utility, unlike the case studies shown in the first two architectures. In addition, SAQE further expands the trade-off space using approximate query processing technique from the database community. As more security and privacy desiderata with integrity constraints are considered in a DBMS, composing them becomes non-trivial. We will highlight the challenges and open questions at the end of this module.

3 INTENDED AUDIENCE

The intended audience for this tutorial includes DB researchers and practitioners who want to learn more about how to solve S/P challenges in data management platforms, as well as members of the DB community who want to learn about the latest achievements in the field and how these technologies can be used in real-world deployments. This tutorial will assume some background in databases, data mining, and the basics of probability with knowledge equivalent of an introductory undergraduate or graduate class. This tutorial will not assume prior knowledge of cryptography or differential privacy.

4 INTENDED LENGTH

The tutorial consists of 3 modules (3 hours). The first module will focus primarily on the problem definition and overview the security and privacy techniques for trustworthy DBMSs (45 mins). The second module will focus on three building blocks (secure computation, differential privacy, and trusted execution environment), and each building block will take 30 mins (90 mins in total). The last module will present several case studies to illustrate the challenges and open questions in building such trustworthy systems (45 mins).

5 RELATED WORK

This tutorial describes both security and privacy concerns for a variety of database settings. We expand upon previous tutorials that focus purely on security or privacy. For differential privacy, previous tutorials have covered the basics and open challenges [52, 81], how to incorporate privacy guarantees into specialized settings like machine learning [16] and information networks [39, 51], as well as how to design appropriate mechanisms for data release [19]. Regarding secure computation, we identified tutorials that cover the basics of secure computation [80], their proofs [49], and specific implementations [6, 15, 65, 71]. In contrast to the referenced work, this tutorial describes how to integrate both privacy and security into database systems. We show the trade-offs and synergies that practitioners must navigate to successfully incorporate differential privacy and secure computation into their systems. Tutorials on blockchains for database systems [54] and private information retrieval [41] focus on a single integrity criteria. Our tutorial will highlight their importance in different settings and focus on the security and privacy guarantees of database systems.

6 PRESENTERS

Ashwin Machanavajjhala is an associate professor in the Department of Computer Science, Duke University and an associate director at the Information Initiative@Duke (iiD). Previously, he was a Senior Research Scientist in the Knowledge Management group at Yahoo! Research. His primary research interests lie in algorithms for ensuring privacy in statistical databases and augmented reality applications. He is a recipient of the National Science Foundation Faculty Early CAREER award in 2013, and the 2008 ACM SIGMOD Jim Gray Dissertation Award Honorable Mention. Ashwin graduated with a Ph.D. from the Department of Computer Science, Cornell University and a B.Tech in Computer Science and Engineering from the Indian Institute of Technology, Madras.

Jennie Rogers is an assistant professor of Computer Science at Northwestern University. She investigates pragmatic privacy-preserving data analytics, federating databases over multiple data models, and new approaches with which individuals can explore and understand their data. She received the NSF CAREER Award in 2019. She earned her Ph.D. at Brown University and completed a post-doc at MIT CSAIL.

Xiao Wang is an assistant professor of Computer Science at Northwestern University. He was a postdoc researcher at MIT and Boston University and obtained his Ph.D. at the University of Maryland. His research interests include computer security, privacy, and cryptography. He has recently been working on practical multi-party computation, zero-knowledge proof, oblivious RAM, and post-quantum cryptography. He is in a team submitting to NIST post-quantum cryptography standardization, currently in round 3. He has received an ACM CCS Best Paper Award in 2017.

Xi He is an assistant professor in the Cheriton School of Computer Science at the University of Waterloo. Her research interests span the areas of privacy and security for big-data management and analysis. She obtained her Ph.D. at Duke University. She also received a double degree in Applied Mathematics and Computer Science from the University of Singapore. She has given tutorials on privacy at VLDB 2016 and SIGMOD 2017. She received a best demo award on differential privacy at VLDB 2016 and was awarded a 2017 Google Ph.D. Fellowship in Privacy and Security.

Johes Bater is a postdoctoral research associate in computer science at Duke University. He completed his Ph.D. from Northwestern University in 2020 under the direction of Jennie Rogers. His primary research interests lie in the intersection of security, privacy, and performance for database systems, with a focus on building fast, accurate database systems that support privacy-preserving analytics with provable security guarantees.

Chenghong Wang is a third year Ph.D. student in computer science at Duke University under the supervision of Dr. Ashwin Machanavajjhala and Dr. Kartik Nayak. His primary research interests lie in the area of differential privacy, applied cryptography, secure computing and database security.

7 ACKNOWLEDGEMENTS

This work was supported by National Science Foundation under the grant #1846447, #2016240, #2016393, #2029853, by DARPA and SPAWAR under contract N66001-15-C-4067, and by NSERC through a Discovery Grant.

REFERENCES

[1] A. Agarwal, M. Herlihy, S. Kamara, and T. Moataz. Encrypted databases for differential privacy. *Proceedings on Privacy Enhancing Technologies*, 2019(3):170 – 190, 01 Jul. 2019.

[2] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: A distributed architecture for secure database services. *CIDR*, 2005.

[3] L. Allen, P. Antonopoulos, A. Arasu, J. Gehrke, J. Hammer, J. Hunter, R. Kaushik, D. Kossmann, J. Lee, R. Ramamurthy, et al. Veritas: shared verifiable databases and tables in the cloud. In *9th Biennial Conference on Innovative Data Systems Research (CIDR)*, 2019.

[4] Apple. *Apple Secure Enclave.* https://support.apple.com/guide/security/secure-enclave-overview-sec59b0b31ff/web.

[5] A. Arasu, S. Blanas, K. Eguro, R. Kaushik, D. Kossmann, R. Ramamurthy, and R. Venkatesan. Orthogonal Security with Cipherbase. In *CIDR*. Citeseer, 2013.

[6] A. Arasu, K. Eguro, R. Kaushik, and R. Ramamurthy. Querying encrypted data (tutorial). In *29th International Conference on Data Engineering (ICDE)*, April 2013. Tutorial presentation.

[7] A. Arasu and R. Kaushik. Oblivious query processing. *ICDT*, 2014.

[8] J. Bater, G. Elliott, C. Eggen, S. Goel, A. Kho, and J. Rogers. SMCQL: Secure Querying for Federated Databases. *Proceedings of the VLDB Endowment*, 10, 2017.

[9] J. Bater, X. He, W. Ehrich, A. Machanavajjhala, and J. Rogers. Shrinkwrap: Differentially-Private Query Processing in Private Data Federations. *Proceedings of the VLDB Endowment*, 12(3):307–320, 2019.

[10] J. Bater, Y. Park, X. He, X. Wang, and J. Rogers. Saqe: Practical privacy-preserving approximate query processing for data federations. *Proc. VLDB Endow.*, 2020.

[11] M. Benedikt, J. Leblay, and E. Tsamoura. Querying with access patterns and integrity constraints. *Proc. VLDB Endow.*, 8(6), Feb. 2015.

[12] D. Bogdanov, L. Kamm, B. Kubo, R. Rebane, V. Sokk, and R. Talviste. Students and Taxes: A Privacy-Preserving Social Study Using Secure Computation. In *Privacy Enhancing Technologies Symposium (PETS)*, 2016.

[13] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. I. Schwartzbach, and T. Toft. Secure multiparty computation goes live. In R. Dingledine and P. Golle, editors, *FC 2009*, volume 5628 of *LNCS*, Accra Beach, Barbados, Feb. 23–26, 2009. Springer, Heidelberg, Germany.

[14] E. Boyle, N. Gilboa, and Y. Ishai. Function secret sharing. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 337–367, Sofia, Bulgaria, Apr. 26–30, 2015. Springer, Heidelberg, Germany.

[15] R. Canetti. Universally composable security: A tutorial, 2016. *Talk at Boston University, A Modular Approach to Cloud Security.*

[16] K. Chaudhuri and A. Sarwate. Differential privacy for signal processing and machine learning. WIFS'14, 2014.

[17] B. Chor, N. Gilboa, and M. Naor. *Private information retrieval by keywords.* Citeseer, 1997.

[18] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *36th FOCS*, pages 41–50, Milwaukee, Wisconsin, Oct. 23–25, 1995. IEEE Computer Society Press.

[19] G. Cormode. Building blocks of privacy: Differentially private mechanisms, 2013. *Invited tutorial talk at Privacy Preserving Data Publication and Analysis (PrivDB) workshop.*

[20] H. Corrigan-Gibbs and D. Kogan. Private information retrieval with sublinear online time. In V. Rijmen and Y. Ishai, editors, *EUROCRYPT 2020, Part I*, LNCS, pages 44–75. Springer, Heidelberg, Germany, May 2020.

[21] V. Costan and S. Devadas. Intel sgx explained. *IACR Cryptol. ePrint Arch.*, 2016(86):1–118, 2016.

[22] E. Crockett, C. Peikert, and C. Sharp. Alchemy: A language and compiler for homomorphic encryption made easy. In *CCS*, 2018.

[23] N. Crooks, M. Burke, E. Cecchetti, S. Harel, R. Agarwal, and L. Alvisi. Obladi: Oblivious serializable transactions in the cloud. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 727–743, 2018.

[24] G. G. Dagher, B. C. M. Fung, N. Mohammed, and J. Clark. Secdm: privacy-preserving data outsourcing framework with differential privacy. *Knowl. Inf.*

Syst., 62(5):1923–1960, 2020.

[25] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: amazon's highly available key-value store. *ACM SIGOPS operating systems review*, 41(6):205–220, 2007.

[26] C. Dwork. Differential privacy. *International Colloquium on Automata, Languages and Programming*, pages 1–12, 2006.

[27] C. Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ICALP'06, pages 1–12, Berlin, Heidelberg, 2006. Springer-Verlag.

[28] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[29] M. El-Hindi, M. Heyden, C. Binnig, R. Ramamurthy, A. Arasu, and D. Kossmann. Blockchaindb-towards a shared database on blockchains. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1905–1908. ACM, 2019.

[30] S. Eskandarian and M. Zaharia. Oblidb: Oblivious query processing using hardware enclaves. *arXiv preprint arXiv:1710.00458*, 2017.

[31] D. Froelicher, J. R. Troncoso-Pastoriza, J. S. Sousa, and J.-P. Hubaux. Drynx: Decentralized, secure, verifiable system for statistical queries and machine learning on distributed datasets. *IEEE Transactions on Information Forensics and Security*, 15:3035–3050, 2020.

[32] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

[33] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press.

[34] A. Gribov, D. Vinayagamurthy, and S. Gorbunov. Stealthdb: a scalable encrypted database with full sql query support. *arXiv preprint arXiv:1711.02279*, 2017.

[35] P. Grubbs, M.-S. Lacharité, B. Minaud, and K. G. Paterson. Learning to Reconstruct: Statistical Learning Theory and Encrypted Database Attacks. In *Learning to Reconstruct: Statistical Learning Theory and Encrypted Database Attacks*, page 0. IEEE.

[36] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart. Leakage-abuse attacks against order-revealing encryption. pages 655–672, 2017.

[37] D. Gupta, B. Mood, J. Feigenbaum, K. Butler, and P. Traynor. Using Intel Software Guard Extensions for Efficient Two-Party Secure Function Evaluation. *4th Workshop on Encrypted Computing and Applied Homomorphic Cryptography - WAHC'16*, (February), 2016.

[38] A. Haeberlen, B. C. Pierce, and A. Narayan. Differential privacy under fire. In *20th USENIX Security Symposium, San Francisco, CA, USA, August 8-12, 2011, Proceedings.* USENIX Association, 2011.

[39] M. Hay, K. Liu, G. Miklau, J. Pei, and E. Terzi. Privacy-aware data management in information networks. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, page 1201–1204, New York, NY, USA, 2011. Association for Computing Machinery.

[40] X. He, A. Machanavajjhala, C. Flynn, and D. Srivastava. Composing differential privacy and secure computation: A case study on scaling private record linkage. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1389–1406, New York, NY, USA, 2017. Association for Computing Machinery.

[41] R. Henry. Tutorial: Private information retrieval. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 2611–2612, New York, NY, USA, 2017. Association for Computing Machinery.

[42] N. Johnson, J. P. Near, and D. Song. Towards practical differential privacy for sql queries. *Proc. VLDB Endow.*, 11(5):526–539, Jan. 2018.

[43] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill. Generic attacks on secure outsourced databases. In *CCS*, pages 1329–1340. ACM, 2016.

[44] E. M. Kornaropoulos, C. Papamanthou, and R. Tamassia. The state of the uniform: attacks on encrypted databases beyond the uniform query distribution. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1223–1240. IEEE, 2020.

[45] I. Kotsogiannis, Y. Tau, X. He, M. Fanaeepour, A. Machanavajjhala, M. Hay, and G. Miklau. PrivateSQL: A Differentially Private SQL Engine. *Proceedings of the VLDB Endowment*, 12(12), 2019.

[46] E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th FOCS*, pages 364–373, Miami Beach, Florida, Oct. 19–22, 1997. IEEE Computer Society Press.

[47] K. Lefevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, pages 49–60. ACM, 2005.

[48] T. Lepoint, S. Patel, M. Raykova, K. Seth, and N. Trieu. Private join and compute from pir with default. Cryptology ePrint Archive, Report 2020/1011, 2020. https://eprint.iacr.org/2020/1011.

[49] Y. Lindell. How to simulate it: A tutorial on the simulation proof technique, 2018. *Tutorials on the Foundations of Cryptography.*

[50] C. Liu, X. S. Wang, K. Nayak, Y. Huang, and E. Shi. ObliVM : A Programming Framework for Secure Computation. *Oakland*, pages 359–376, 2015.

[51] K. Liu, G. Miklau, J. Pei, and E. Terzi. Privacy-aware data mining in information networks. *KDD 2010 Tutorial*, 2010.

[52] A. Machanavajjhala, X. He, and M. Hay. Differential privacy in the wild: A tutorial on current practices and open challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, page 1727–1730, New York, NY, USA, 2017. Association for Computing Machinery.

[53] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, page 277–286, USA, 2008. IEEE Computer Society.

[54] S. Maiyya, V. Zakhary, M. J. Amiri, D. Agrawal, and A. El Abbadi. Database and distributed computing foundations of blockchains. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 2036–2041, New York, NY, USA, 2019. Association for Computing Machinery.

[55] F. D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09. ACM, 2009.

[56] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. Computational differential privacy. In *CRYPTO*, pages 126–142. Springer, 2009.

[57] P. Mohassel, P. Rindal, and M. Rosulek. Fast database joins and psi for secret shared data. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1271–1287, New York, NY, USA, 2020. Association for Computing Machinery.

[58] A. Narayan and A. Haeberlen. DJoin: differentially private join queries over distributed databases. *Proceedings of the 10th USENIX Symposium ...*, page 14, 2012.

[59] M. Naveed, S. Kamara, and C. V. Wright. Inference attacks on property-preserving encrypted databases. In I. Ray, N. Li, and C. Kruegel, editors, *ACM CCS 2015*, pages 644–655, Denver, CO, USA, Oct. 12–16, 2015. ACM Press.

[60] M. Naveed, C. V. Wright, S. Kamara, and C. V. Wright. Inference Attacks on Property-Preserving Encrypted Databases. In *CCS*, pages 644–655. ACM, 2015.

[61] F. Olumofin and I. Goldberg. Privacy-preserving queries over relational databases. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 75–92. Springer, 2010.

[62] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan. Cryptdb: Protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, page 85–100, New York, NY, USA, 2011. Association for Computing Machinery.

[63] C. Priebe, K. Vaswani, and M. Costa. EnclaveDB: A Secure Database using SGX. In *EnclaveDB: A Secure Database using SGX*, page 0. IEEE, 2018.

[64] A. Rajan, L. Qin, D. W. Archer, D. Boneh, T. Lepoint, and M. Varia. Callisto: A cryptographic approach to detecting serial perpetrators of sexual misconduct. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, page 49. ACM, 2018.

[65] M. Rosulek. A brief history of practical garbled circuit optimizations, 2015. *Talk at Simons Secure Computation Workshop.*

[66] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: Security and privacy for mapreduce. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, page 20, USA, 2010. USENIX Association.

[67] A. Roy Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha. Crypte: Crypto-assisted differential privacy on untrusted servers. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, New York, NY, USA, 2020. Association for Computing Machinery.

[68] A. Roy Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha. Cryptϵ: Crypto-assisted differential privacy on untrusted servers. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 603–619, 2020.

[69] M. Sabt, M. Achemlal, and A. Bouabdallah. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. IEEE, 2015.

[70] S. Sasy, S. Gorbunov, and C. Fletcher. ZeroTrace: Oblivious memory primitives from Intel SGX. *IACR Cryptology ‘Archive Report*, 549:2017, 2017.

[71] S. Sharma, A. Burtsev, and S. Mehrotra. Advances in cryptography and secure hardware for data outsourcing. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1798–1801, 2020.

[72] M. Suresh, Z. She, W. Wallace, A. Lahlou, and J. Rogers. Kloakdb: A platform for analyzing sensitive data with k-anonymous query processing. *CoRR*, abs/1904.00411, 2019.

[73] J. Van Bulck, N. Weichbrodt, R. Kapitza, F. Piessens, and R. Strackx. Telling your secrets without page faults: Stealthy page table-based attacks on enclaved execution. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1041–1056, 2017.

[74] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros. Conclave: Secure multi-party computation on big data. In *EuroSys*, 2019.

[75] F. Wang, C. Yun, S. Goldwasser, V. Vaikuntanathan, and M. Zaharia. Splinter: Practical Private Queries on Public Data. In *NSDI*, pages 299–313, 2017.

[76] W. Wang, G. Chen, X. Pan, Y. Zhang, X. Wang, V. Bindschaedler, H. Tang, and C. A. Gunter. Leaky Cauldron on the Dark Land: Understanding Memory Side-Channel

Hazards in SGX. *CCS*, pages 2421–2434, 2017.

[77] X. Wang, S. Ranellucci, and J. Katz. Authenticated garbling and efficient maliciously secure two-party computation. In *CCS*, 2017.

[78] Z. Wei, U. Leck, and S. Link. Entity integrity, referential integrity, and query optimization with embedded uniqueness constraints. In *ICDE*, 2019.

[79] W. K. Wong, B. Kao, D. W. L. Cheung, R. Li, and S. M. Yiu. Secure query processing with data interoperability in a cloud database environment. In *SIGMOD*, pages 1395–1406. ACM, 2014.

[80] B. P. Y. Lindell. Secure computation and efficiency, 2011. *Invited talk at Bar-Ilan Winter School*.

[81] Y. Yang, Z. Zhang, G. Miklau, M. Winslett, and X. Xiao. Differential privacy in data publication and analysis. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, page 601–606, New York, NY, USA, 2012. Association for Computing Machinery.

[82] A. C.-C. Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167, Toronto, Ontario, Canada, Oct. 27–29, 1986. IEEE Computer Society Press.

[83] D. Zhang, R. McKenna, I. Kotsogiannis, M. Hay, A. Machanavajjhala, and G. Miklau. EKTELO: A framework for defining differentially-private computations. In G. Das, C. M. Jermaine, and P. A. Bernstein, editors, *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 115–130. ACM, 2018.

[84] Y. Zhang, D. Genkin, J. Katz, D. Papadopoulos, and C. Papamanthou. vSQL: Verifying arbitrary SQL queries over dynamic outsourced databases. Cryptology ePrint Archive, Report 2017/1145, 2017. https://eprint.iacr.org/2017/1145.

[85] Y. Zhang, J. Katz, and C. Papamanthou. IntegriDB : Verifiable SQL for Outsourced Databases. *ACM CCS*, 2015.

[86] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, and I. Stoica. Opaque: An oblivious and encrypted distributed analytics platform. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 283–298, 2017.