Distributed systems and trusted execution environments: Trade-offs and challenges

Rafael Pires[1*]
Pascal Felber[2] *(advisor)*, **Marcelo Pasin**[3] *(co-advisor)*

[1] EPFL – Swiss Federal Institute of Technology in Lausanne, Switzerland

[2]UniNE – University of Neuchâtel, Switzerland

[3]HES-SO HE-Arc – University of Applied Sciences and Arts of Western Switzerland
Neuchâtel, Switzerland

{first.last}@epfl.ch, @unine.ch, @he-arc.ch

Abstract. *This extended abstract summarises my PhD thesis[1], which explores design strategies for distributed systems that leverage trusted execution environments (TEEs). We aim at achieving better security and privacy guarantees while maintaining or improving performance in comparison to existing equivalent approaches. To that end, we propose a few original systems that take advantage of TEEs. On top of prototypes built with Intel software guard extensions (SGX) and deployed on real hardware, we evaluate their limitations and discuss the outcomes of such an emergent technology.*

1. Introduction

Security and privacy concerns in computer systems have grown in importance with the ubiquity of connected sensing devices. Additionally, cloud computing boosts such distress as private data are stored and processed in multi-tenant infrastructure providers. In recent years, TEEs have caught the attention of scientific and industry communities as they became largely available in user- and server-class machines.

Integrity and confidentiality of applications are typically enforced by means of logical isolation. Virtual address spaces and privileged instructions, for instance, are traditional hardware mechanisms used by operating systems (OSes) to prevent unauthorized processes from getting access to potentially sensitive pieces of memory or operations. Such mechanisms protect user applications from one another (memory management), multiple OSes from one another (virtualisation), and the system software from user applications (privileged instructions). However, they neither provide isolation to user applications against the system software nor are resilient to physical attacks.

Being vulnerable to the mentioned threats is particularly critical in multi-tenant setups, where the provider controls at least part of the system software. In such environments, cryptography is widely used to protect data in transit or at rest. When data are processed though, ciphertexts must be deciphered before loaded into memory. During processing, data confidentiality is hence threatened by privileged users and physical attackers. To solve issues like these, TEEs were proposed.

---

[*]This work was developed while the author was affiliated to the University of Neuchâtel.

[1]Full text available through the DOI 10.35662/unine-thesis-2812

TEEs provide security guarantees based on cryptographic constructs built in hardware, therefore offering stronger protection against remote or physical attacks when compared to their software counterparts. Intel SGX, in particular, implements powerful mechanisms that shield sensitive data even from privileged users with full control of system software.

Designing secure distributed systems is remarkably challenging, since they involve many coordinated processes running in geographically-distant nodes, therefore having numerous points of attack. Since TEEs provide an opportunity to tackle these challenges, we explore some of them by using Intel SGX as cornerstone. We do so by designing and experimentally evaluating several elementary systems ranging from communication and processing middleware to a peer-to-peer privacy-preserving solution.

2. Communication and processing

We start with support systems that naturally fit cloud deployment scenarios, namely content-based routing (CBR), batching and stream processing frameworks. Our communication middleware called secure content-based routing (SCBR) consists in the first system that demonstrates the practical benefits of SGX for privacy-preserving CBR. From a privacy perspective, we aim at protecting CBR's most critical stage: matching subscriptions against publications. To achieve that, we perform the compute-intensive matching operations in the trusted environment, within protected *enclaves*. Simply put, clients send their subscriptions to routers which later compare them against publications provisioned by data providers and forward this content to the matching subscribers (Figure 1).

Unlike software-only cryptographic approaches such as homomorphic encryption or asymmetric scalar-product preserving encryption (ASPE) [Choi et al. 2010], we are able to use plaintext data and containment relations among subscriptions [Carzaniga et al. 2001]. This technique explores relationships between subscription predicates, which entails a reduced memory footprint of the subscription index and improved matching speeds. As a consequence, we reach performance gains of one order of magnitude in comparison to ASPE while offering analogous guarantees (see curves corresponding to Native ASPE and Enclaved SCBR in Figure 2a).

The processing platforms, in turn, receive encrypted data and code to be executed within the trusted environment. For doing so, we ported a Lua interpreter engine to run inside secure enclaves and leveraged it as execution unit that operates on code and data provisioned

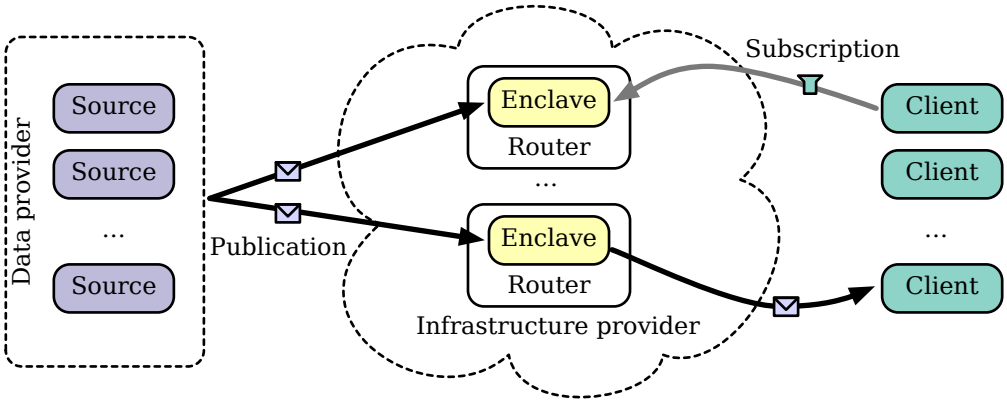


Figure 1. SCBR model

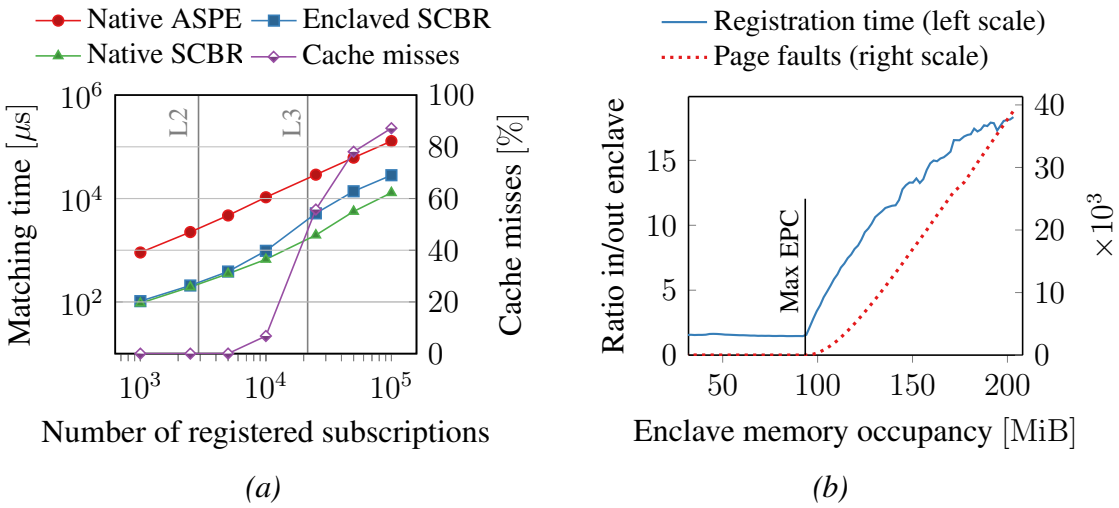


Figure 2. SGX caching and paging effects under SCBR

on the fly. On top of this, we propose the Lightweight MapReduce, a processing framework based on a programming model extensively used for parallel data processing in distributed environments. From the usability perspective, a user can just write MapReduce scripts to be run within enclaves. The framework, in turn, handles data encryption and dissemination. Besides, we observe the performance influence of going beyond the last level cache (LLC) in enclave executions. In this regard, Figure 2a shows that effect when going beyond the L3 cache limit when contrasting the curves that correspond to Enclaved and Native SCBR.

MapReduce jobs operate on data batches, *i.e.*, possibly large and finite data chunks that are entirely available upfront. For supporting the treatment of timely unbound events, we propose SecureStreams: a reactive middleware built on top of Lua libraries. Its architecture relies on Lua virtual machine (VM) pairs on each node, *i.e.*, one running inside enclaves and another outside. This way, only sensitive data processing is relayed to trusted environments, while message queuing and the pipeline management is kept outside. Then, we analyse performance losses when compared to unsafe executions in terms of throughput and scalability.

We also analyse SGX overheads when surpassing its memory limitations, specially with respect to the enclave page cache (EPC), which consists of a reserved memory that is automatically encrypted by the processor. Figure 2b shows up to 18× performance loss when using roughly twice as much the EPC limit for storing subscriptions with SCBR. This is mostly due to the management of data structures that ensure data integrity and the memory swapping between protected and unprotected regions, since the EPC is oversubscribed.

With all these systems, we could observe the overheads due to SGX, which are mostly influenced by memory usage. These findings indicate the architectural challenges that one should face when desiging scalable platforms that take advantage of this technology. In spite of this, we also show that horizontal scalability is a viable mitigation to overcome its limitations.

3. Group communication and data sharing

Looking into the adjustments that one should make to leverage secure enclaves in distributed communication and processing systems, we observed considerable performance implications under memory-intensive scenarios. Because of this, simply porting memory-eager artifacts to run within SGX enclaves may not be a good idea. Nevertheless, the benefits of TEEs outreach such niche. Confidentiality and isolation properties are also desired in less memory-eager applications.

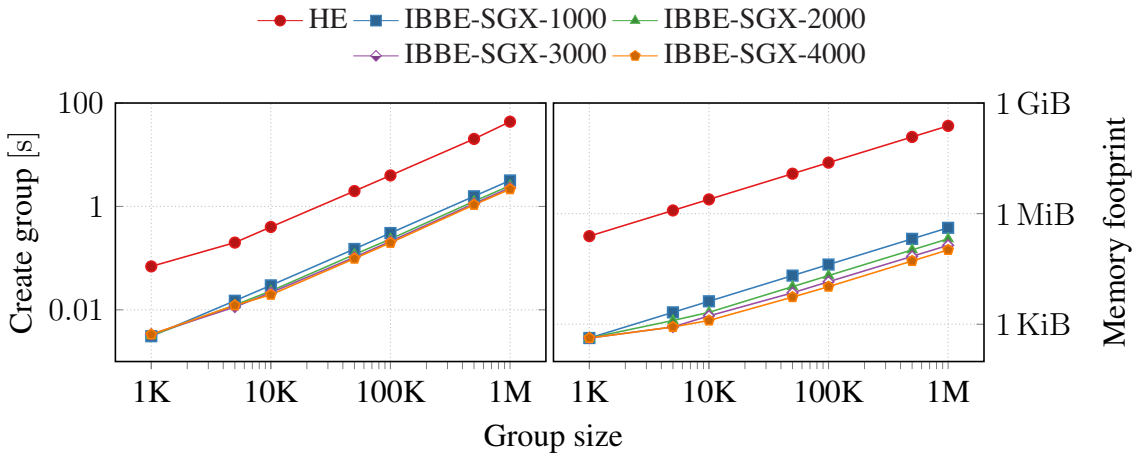


Figure 3. IBBE-SGX performance improvements with respect to HE considering several partition sizes (from 1000 to 4000)

In this direction, we turn our attention to designing or adapting cryptographic schemes by taking advantage of TEEs for protecting very sensitive data: cryptographic keys. We first present IBBE-SGX, a new cryptographic access control extension for collaborative editing of shared data. Thanks to TEEs, we are able to cut part of the computational complexity of an identity-based broadcast encryption (IBBE) scheme [Delerablée 2007]. Shielding a master key inside the trusted environment allows us to spare considerable computation time by avoiding the usage of an IBBE public-key during encryption. Because of this, we improve performance by orders of magnitude in comparison to hybrid encryption (HE) (*i.e.*, using asymmetric cryptography to encrypt a symmetric shared key), both in terms of membership changes (Figure 3 right) and produced metadata (Figure 3 left), consequently also profiting in storage and network usage.

For decryption, however, we assume end users might use portable devices and therefore they may not have SGX hardware. To mitigate the overheads due to the quadratic complexity of such operation, we propose a partitioning scheme, so that the complexity becomes bounded to the number of users in one partition rather than in the whole group. Table 1 shows the comparison between complexities among our proposal and IBBE.

To be able to communicate using IBBE, a group member must know the identities of all other members in the same group, which represents a potential privacy threat. In face of

Table 1. IBBE-SGX: Operations complexities comparison.

Operation	IBBE	IBBE-SGX	IBBE-SGX with partitioning
System setup	$O(N)$	$O(N)$	$O(n)$
Extract user key	$O(1)$	$O(1)$	$O(1)$
Create group key	$O(N^2)$	$O(N)$	$O(mn)$
Add user to group		$O(1)$	$O(1)$
Remove user from group		$O(1)$	$O(m)$
Decrypt group key	$O(N^2)$	$O(N^2)$	$O(n^2)$

Cardinalities: N: global number of users. n: members in one partition. m: number of partitions.

this, we present A-Sky, which grants anonymity among group members. Instead of relying on costly asymmetric cryptography like pretty good privacy (PGP), secure enclaves allow A-Sky to create key envelopes using efficient symmetric operations, hence achieving faster execution times and shorter ciphertexts. In addition, we only require the usage of a TEE proxy for writing to the shared storage and leave the dominant data consumption operations directly in charge of rightful readers. We propose an end-to-end system based on micro-services which communicate through a representational state transfer (REST) interface. We then evaluate A-Sky in terms of performance and scalability.

Our cryptographic protocols for group data sharing show how TEEs can be used to reduce the computational complexity of legacy methods. As a bonus, our proposals allow large savings on metadata volume and processing time of cryptographic operations, all with equivalent security guarantees. This highlights some of the advantages of using TEEs in security-sensitive applications, despite their limitations in terms of memory consumption.

4. Privacy enforcement

Rather than proposing server-side designs, we focus next on privacy-preserving solutions from the perspective of users. After all, they cannot modify some existing systems like web-search engines. Moreover, the providers of these services may keep individual profiles containing sensitive private information about them. We aim at achieving indistinguishability and unlinkability properties by employing techniques like sensitivity analysis, query obfuscation and leveraging relay nodes.

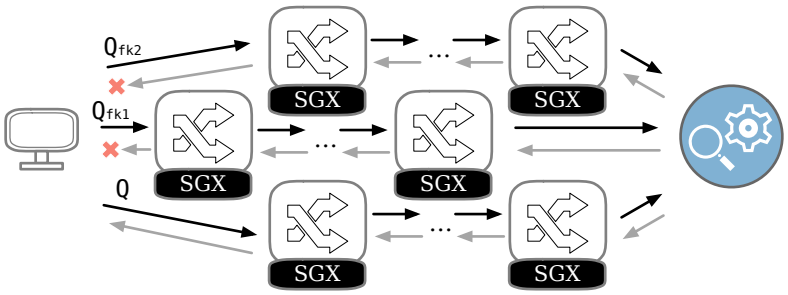
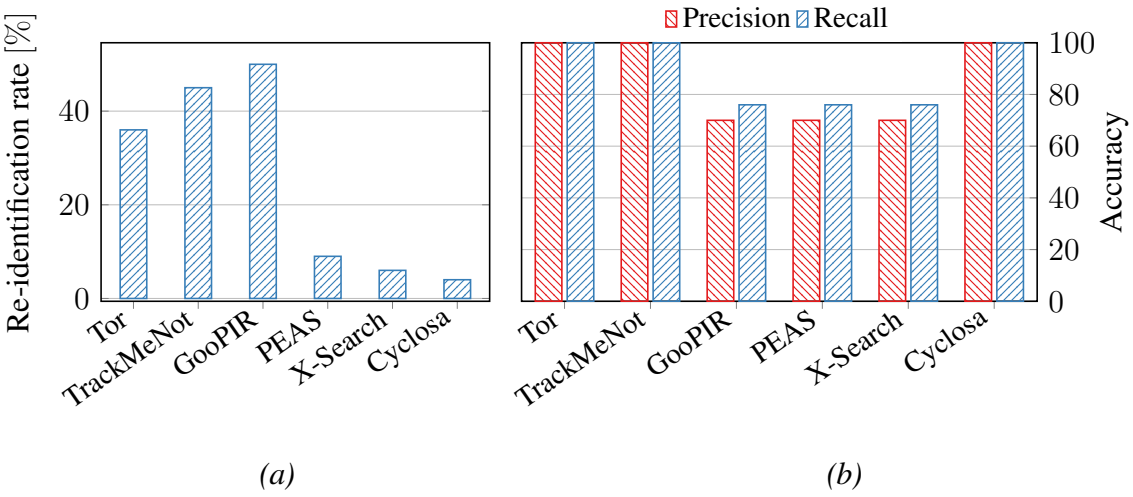


Figure 4. Cyclosa



(a) *(b)*

Figure 5. Cyclosa performance

X-Search leverages TEEs for providing a privacy-preserving solution for Web search. In order to prevent service providers from keeping accurate user profiles and therefore obstruct privacy breaches, we propose a SGX proxy between users and search engines. From the service provider's perspective, queries originate from another source, thus becoming more difficult to link them back to their issuing users (unlinkability). Since the proxy operates in the trusted environment, we can safely store past user queries and use them to obfuscate requests, so that the search engine cannot distinguish real from fake queries (indistinguishability). These strategies combined offer stronger privacy guarantees and outperform previous approaches in latency and throughput.

Having a centralised proxy, however, can be ineffective in terms of scalability. Moreover, the X-Search proxy can be easily neutralised by search providers due to the excessive amount of requests it potentially makes. To tackle that, we propose Cyclosa, where we spread the load across a peer-to-peer (P2P) network of SGX relay nodes (Figure 4). Each one may issue their own queries through the decentralised network and also forward requests to the search engine on behalf of others, always having enclaves as intermediaries.

Obfuscation is done through different paths, thus facilitating the delivery of results by simply discarding those that handle fake queries and therefore achieving perfect accuracy. Although this approach generates more traffic, the burden is scattered across the participating nodes. To lessen the extra load, we propose a sensitivity analysis scheme that reduces the amount of fake queries for requests that do not contain sensitive terms.

By decentralising requests, Cyclosa solves the issue of possibly being blacklisted by search engines while meeting scalability and accuracy requirements. We compare Cyclosa and X-Search to solutions that provide unlinkability (Tor [Dingledine et al. 2004]), indistinguishability (TrackMeNot [Howe and Nissenbaum 2009] and GooPIR [Domingo-Ferrer et al. 2009]) and a combination of both (PEAS [Petit et al. 2015]). The outcomes of our experiments indicate that Cyclosa is the most robust system in comparison to existing solutions with regard to user re-identification rates (Figure 5a) and accuracy of results (Figure 5b).

5. Related work

As the first SGX machines were released in the last quarter of 2015, all its possibilities is yet to be defined. Despite that, much research work has been done. Some supporting systems like SCONE [Arnautov et al. 2016] or Graphene-SGX [Tsai et al. 2017] offer a runtime that ease porting legacy applications to SGX.

Closer to ours, VC3 [Schuster et al. 2015] and SecureKeeper [Brenner et al. 2016] proposed distributed systems on top of TEEs. Some vulnerabilities such as Foreshadow [Bulck et al. 2018] were also published. Despite them, designs that count on TEEs do not lose their relevance due to occasional breaches. In principle, our proposals could use different (even future) hardware implementations that retain equivalent or comparable features to SGX.

6. Conclusion

Designing secure distributed systems is complex. Their dispersed nature multiplies the number of attack vectors. On top of that, further threats arise when deploying such systems on shared infrastructures. Countermeasures to risks that stem from providers, their personnel or co-located tenants are either infeasible due to computational complexity or fail at protecting running code

and data from powerful adversaries like the OS. The design of secure systems can greatly benefit from TEEs, and this work explores such domain.

We start with middleware for distributed communication and processing. These use cases highlight the challenges that arise from the very limited amount of memory available within SGX enclaves before incurring in prohibitive overheads. In spite of that, we leverage SGX for designing efficient cryptographic protocols that are able to take advantage of the TEE isolation in order to perform operations with lower computational complexity while preserving security assurances. Finally, we propose privacy-preserving Web search systems along with the trade-off between privacy protection and the production of extra network load.

In a nutshell, this thesis proposes new mechanisms that take advantage of TEEs for distributed architectures. We show through an empirical approach on top of Intel SGX what are the compromises of distinct designs applied to distributed systems. We believe that TEEs came to stay and further research will certainly help to advance this exciting area. Hopefully, the contributions of this thesis will facilitate such purpose.

7. Publications

This work has generated several thousands of lines of code and text, apart from presentations in conferences, workshops and seminars. All papers have a pre-print version publicly available in arXiv and most of the source code is open. Publications are listed below in chronological order.

1. Rafael Pires, Marcelo Pasin, Pascal Felber, and Christof Fetzer. **Secure content-based routing using intel software guard extensions** in Proc. of the 17th Int. Middleware Conf., Middleware '16 (*Qualis A2*). Trento, Italy: ACM, Dec. 2016, pp. 10:1–10:10. DOI: 10.1145/2988336.2988346. arXiv: 1701.04612. Source code: github.com/rafaelppires/scbr.

2. Florian Kelbert, Franz Gregor, Rafael Pires, Stefan Köpsell, Marcelo Pasin, Aurélien Havet, Valerio Schiavoni, Pascal Felber, Christof Fetzer, and Peter Pietzuch. **SecureCloud: Secure big data processing in untrusted clouds** in Design, Automation Test in Europe Conf. Exhibition (DATE - *Qualis A1*). Lausanne, Switzerland, Mar. 2017, pp. 282–285. DOI: 10.23919/DATE.2017.7926999. arXiv: 1805.01783.

3. Rafael Pires, Daniel Gavril, Pascal Felber, Emanuel Onica, and Marcelo Pasin. **A lightweight MapReduce framework for secure processing with SGX** in 17th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing (CCGRID - *Qualis A1*). Int. Workshop on Assured Cloud Computing and QoS aware Big Data (WACC'17). Madrid, Spain, May 2017, pp. 1100–1107. DOI: 10.1109/CCGRID.2017.129. arXiv: 1705.05684. Source code: github.com/rafaelppires/sgx_lightweight_mapreruce.

4. Aurélien Havet, Rafael Pires, Pascal Felber, Marcelo Pasin, Romain Rouvoy, and Valerio Schiavoni **SecureStreams: A reactive middleware framework for secure data stream processing** in Proc. of the 11th ACM Int. Conf. on Distributed and Event-based Systems, DEBS '17 (*Qualis A2*). Barcelona, Spain: ACM, Jun. 2017, pp. 124–133. DOI: 10.1145/3093742.3093927. arXiv: 1805.01752. Source code: github.com/vschiavoni/SecureStreams-DEBS17.

5. Sonia Ben Mokhtar, Antoine Boutet, Pascal Felber, Marcelo Pasin, Rafael Pires, and Valerio Schiavoni **X-search: Revisiting private web search using intel SGX** in Proc. of the 18th Middleware Conf., Middleware '17 (*Qualis A2*). Las Vegas, USA: ACM, Dec. 2017, pp. 198–208. DOI: 10.1145/3135974.3135987. arXiv: 1805.01742. Source code: github.com/rafaelppires/x_search.

6. Stefan Contiu, Rafael Pires, Sébastien Vaucher, Marcelo Pasin, Pascal Felber, and Laurent Réveillère. **IBBE-SGX: Cryptographic group access control using trusted execution environments** in 48th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN - *Qualis A1*). Luxembourg, Jun. 2018, pp. 207–218. DOI: 10.1109/DSN.2018.00032. arXiv: 1805.01563. Source code: github.com/rafaelppires/trusted-sharing.

7. Sébastien Vaucher, Rafael Pires, Pascal Felber, Marcelo Pasin, Valerio Schiavoni, and Christof Fetzer. **SGX-aware container orchestration for heterogeneous clusters** in IEEE 38th Int. Conf. on Distributed Computing

Systems (ICDCS - *Qualis A1*). Vienna, Austria, Jul. 2018, pp. 730– 741. DOI: 10.1109/ICDCS.2018.00076. arXiv: 1805.05847. Source code: github.com/sebva/sgx-orchestrator.

8. Rafael Pires, David Goltzsche, Sonia Ben Mokhtar, Sara Bouchenak, Antoine Boutet, Pascal Felber, Rüdiger Kapitza, Marcelo Pasin, and Valerio Schiavoni. **Cyclosa: Decentralizing private web search through SGX-based browser extensions** in IEEE 38th Int. Conf. on Distributed Computing Systems (ICDCS - *Qualis A1*). Vienna, Austria, Jul. 2018, pp. 467–477. DOI: 10.1109/ICDCS.2018.00053. arXiv: 1805.01548.

9. Christian Göttel, Rafael Pires, Isabelly Rocha, Sébastien Vaucher, Pascal Felber, Marcelo Pasin, and Valerio Schiavoni. **Security, performance and energy trade-off of hardware-assisted memory protection mechanisms** in IEEE 37th Int. Symp. on Reliable Distributed Systems (SRDS - *Qualis B1*). Salvador, Brazil, Oct. 2018, pp. 133-142. DOI: 10.1109/SRDS.2018.00024. arXiv: 1903.04203.

10. Andrei Mogage, Rafael Pires, Crăciun Vlad, Emanuel Onica, and Pascal Felber. **Supply chain malware targets SGX: Take care of what you sign** in IEEE 38th Int. Symp. on Reliable Distributed Systems (SRDS - *Qualis B1*). Lyon, France, Oct. 2019, pp. 52-60. DOI: 10.1109/SRDS47363.2019.00016 arXiv: 1907.05096.

11. Stefan Contiu, Sébastien Vaucher, Rafael Pires, Marcelo Pasin, Pascal Felber, and Laurent Réveillère. **Anonymous and confidential file sharing over untrusted clouds** in IEEE 38th Int. Symp. on Reliable Distributed Systems (SRDS - *Qualis B1*). Lyon, France, Oct. 2019, pp. 21-31. DOI: 10.1109/SRDS47363.2019.00013 arXiv: 1907.06466. Source code: github.com/rafaelppires/anonym-sharing.

References

Arnautov, S., Trach, B., Gregor, F., Knauth, T., Martin, A., Priebe, C., Lind, J., Muthukumaran, D., O'Keeffe, D., Stillwell, M. L., Goltzsche, D., Eyers, D., Kapitza, R., Pietzuch, P., and Fetzer, C. (2016). SCONE : Secure linux containers with intel SGX. In *12th USENIX Symp. on OSes Design and Implementation (OSDI)*.

Brenner, S., Wulf, C., Goltzsche, D., Weichbrodt, N., Lorenz, M., Fetzer, C., Pietzuch, P., and Kapitza, R. (2016). SecureKeeper: Confidential ZooKeeper using intel SGX. In *Proc. of the 17th Int. Middleware Conf.*, New York, NY, USA.

Bulck, J. V., Minkin, M., Weisse, O., Genkin, D., Kasikci, B., Piessens, F., Silberstein, M., Wenisch, T. F., Yarom, Y., and Strackx, R. (2018). Foreshadow: Extracting the keys to the intel SGX kingdom with transient out-of-order execution. In *27th USENIX Security Symp.*

Carzaniga, A., Rosenblum, D. S., and Wolf, A. L. (2001). Design and evaluation of a wide-area event notification service. *ACM TOCS*, 19(3).

Choi, S., Ghinita, G., and Bertino, E. (2010). A privacy-enhancing content-based publish/subscribe system using scalar product preserving transformations. In *21st Int. Conf. on database and expert systems applic. (DEXA)*, Bilbao, Spain.

Delerablée, C. (2007). Identity-based broadcast encryption with constant size ciphertexts and private keys. In *Advances in Cryptology – ASIACRYPT 2007*.

Dingledine, R., Mathewson, N., and Syverson, P. F. (2004). Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symp.*, San Diego, USA.

Domingo-Ferrer, J., Solanas, A., and Castellà-Roca, J. (2009). h(k)-private information retrieval from privacy-uncooperative queryable databases. *Online Inf. Review*, 33(4).

Howe, D. C. and Nissenbaum, H. (2009). TrackMeNot: Resisting surveillance in web search. *Lessons from the id. trail: Anonymity, priv., and id. in a networked society*, 23.

Petit, A., Cerqueus, T., Mokhtar, S. B., Brunie, L., and Kosch, H. (2015). PEAS: Private, efficient and accurate web search. In *2015 IEEE Trustcom/BigDataSE/ISPA*.

Schuster, F., Costa, M., Fournet, C., Gkantsidis, C., Peinado, M., Mainar-Ruiz, G., and Russinovich, M. (2015). VC3: Trustworthy data analytics in the cloud using SGX. In *IEEE Symp. on Security and Privacy*.

Tsai, C., Porter, D. E., and Vij, M. (2017). Graphene-SGX: A practical library OS for unmodified applications on SGX. In *USENIX ATC*, Santa Clara, CA.