Time to Rethink: Trust Brokerage Using Trusted Execution Environments

Patrick Koeberl[1], Vinay Phegade[2], Anand Rajan[2], Thomas Schneider[3], Steffen Schulz[1(✉)], and Maria Zhdanova[4]

[1] Intel Labs, Darmstadt, Germany
{patrick.koeberl,steffen.schulz}@intel.com
[2] Intel Labs, Portland, ON, USA
{vinay.phegade,anand.rajan}@intel.com
[3] TU Darmstadt, Darmstadt, Germany
thomas.schneider@ec-spride.de
[4] Fraunhofer SIT, Darmstadt, Germany
maria.zhdanova@sit.fraunhofer.de

Abstract. Mining and analysis of digital data has the potential to provide improved quality of life and offer even life-saving insights. However, loss of privacy or secret information would be detrimental to these goals and inhibit widespread application. Traditional data protection measures tend to result in the formation of data silos, severely limiting the scope and yield of "Big Data". Technology such as privacy-preserving multi-party computation (MPC) and data de-identification can break these silos enabling privacy-preserving computation. However, currently available de-identification schemes tend to suffer from privacy/utility trade-offs, and MPC has found deployment only in niche applications.

As the assurance and availability of hardware-based Trusted Execution Environments (TEEs) is increasing, we propose an alternative direction of using TEEs as "neutral" environments for efficient yet secure multi-party computation. To this end, we survey the current state of the art, propose a generic initial solution architecture and identify remaining challenges.

1 Introduction

Large amounts of data are created and accumulated all around us. This trend is increasing and data is commonly named the digital fuel of the 21st century. In fact, analyzing such "big data" has huge expected business value. Already today, many applications in all areas of life benefit from such big data analysis ranging

Thomas Schneider—This work has been co-funded by the European Union (EU FP7/2007-2013) grant agreement n. 609611 (PRACTICE), by the DFG project E3 within the CRC 1119 CROSSING, by the BMBF within EC SPRIDE, and by the Hessian LOEWE excellence initiative within CASED.
Maria Zhdanova—This work has been co-funded by the EU project PRIPARE ID 610613.

M. Conti et al. (Eds.): TRUST 2015, LNCS 9229, pp. 181–190, 2015.
DOI: 10.1007/978-3-319-22846-4_11

from "people you might know" in social networks, over rating and reputation systems on eBay, to product recommendations on Amazon. However, privacy and security of data is increasingly critical to these applications as consumers become aware of the risks associated with aggregating digital identities, payment information and personal profiles in the cloud. Similarly, companies are hesitant to make their data assets available for external analysis due to the risk of losing control or violating their clients' privacy. As a result, vast amounts of data remain locked in data silos, unavailable to use for business and research.

Breaking the Data Silos. A solution to privacy-preserving multiparty computation must assure that data owners retain control of the data during transfer, storage and processing. In addition to data confidentiality, it must be assured that privacy is maintained even when results of different computations are combined or correlated with public information. Privacy-preserving filtering schemes must be applied to prevent such attacks, and the system must allow the data owners to flexibly negotiate and enforce such policies. Finally, data owners must obtain assurances that the requested policies are enforced and able to revoke access on violation. However, even if a solution meets all these security and privacy requirements, which have been the focus of much prior work, it must also meet some key ecosystem requirements in order to qualify as a practical solution:

1. **Solution Cost:** This includes cost to design the solution, and total cost to run and maintain the solution during deployment. The cost will increase if the system is complex to design, maintain or requires frequent re-design for different usages. Enabling existing developer skill sets, leveraging tools and automation are important factors for broad ecosystem acceptance.
2. **Data Utility:** In order to assure the privacy and confidentiality of the computation, implementations may demand sacrifices on the extent and accuracy of data sets. An ideal implementation should not limit the available privacy/utility trade-offs.
3. **Performance & Scalability:** Computational performance is important for the overall cloud analytics scenario to be economically viable. For interactive applications, users require acceptable application responsiveness. Additionally, as data size grows from terabytes to petabytes, the computation should be able to scale to distributed storage and computation networks.

Our Contributions. This position paper proposes a TEE-based "trust brokerage" approach that enables multiple parties to compute under previously agreed security assurances. We review current approaches based on Secure Multi-Party Computation (MPC) and Data De-Identification (DDI) in Sect. 2, pointing to the significant recent advances in Trusted Execution Environments (TEEs). We then describe a generic TEE-based solution architecture in Sect. 3 and compare it with previous approaches. As expected, a TEE-based solution is more efficient than MPC and enables better data utility and flexibility than DDI. We conclude with a call to action in Sect. 4, detailing the major research challenges that must still be resolved to achieve a secure and scalable solution.

2 Research Developments and State of the Art

In the following we review the state of the art in secure multi-party computation (Sect. 2.1), data de-identification (Sect. 2.2), and trusted execution environments (Sect. 2.3).

2.1 Secure Multi-party Computation

Secure Multi-party Computation (MPC) was invented in the late 1980s [12,27]. It allows two or multiple parties to jointly evaluate a function on their joint inputs without revealing anything but the result of the computation.

During the first twenty years after its invention, MPC was perceived as a feasibility result of theoretical interest. However, the situation changed in 2004 where the Fairplay project presented the first MPC implementation [16]. Since then, MPC has received renewed interest and many tools have been

The first real-world deployment of MPC was the Danish sugar beets auction in 2009 [3]. Since then, some small companies have developed the first MPC products, e.g., Cybernetica's Sharemind[1] to analyze confidential data, The Alexandra Institute's Partisia[2] for auctions and exchanges, or Dyadic Security[3] to protect against server breaches. However, some major roadblocks remain for large-scale deployment of MPC, mainly due to its low performance and high design costs that do not yet meet RoI expectations.

The drawback of MPC is that protocols often represent the computation function as a Boolean circuit, which results in of billions of gates for realistic applications. The runtime and communication of today's most efficient MPC protocols is linear in the size of this circuit.

MPC is designed for a world without trusted third parties, which would otherwise be able to retrieve all necessary inputs, compute the function and return the output to participants. In fact, MPC literature often argues that such trusted third parties do not exist or point to the poor computation and I/O capabilities of smartcards. However, some researchers also suggested to improve MPC performance by extending it with trusted hardware, e.g., [7,13].

2.2 Data De-Identification (DDI)

Data De-identification (DDI) is a procedure where personally identifying information is removed from a data set to ensure that distinct data items cannot be linked to individuals. It is different from anonymization and pseudonymization in that some identifying information may be retained in the data or remain accessible to trusted parties. Depending on the context, various types of information may be considered personally identifying, for example, it was shown that 87 % of US citizens can be uniquely identified by the combination of ZIP code, date of birth, and sex [25]. Unlike direct identifiers which are easily masked or deleted,

[1] https://sharemind.cyber.ee.
[2] http://alexandra.dk/uk/expertise/products/partisia.
[3] https://www.dyadicsec.com.

such quasi-identifiers tend to be valuable for analytics and are often retained, leaving a risk of re-identification.

Methods for mitigating the risk of re-identification were presented as early as 1974 as part of Statistical Disclosure Control (SDC) [4]. SDC methods include (1) non-perturbative techniques that re-encode attributes without modifying initial values, (2) perturbative techniques which add noise, randomize or aggregate attributes, and (3) generation of synthetic data such that the initial relationships and characteristics are preserved. However, depending on the computation to be performed, such modifications can destroy the integrity of data, reduce their quality and change overall statistics, rendering them useless for analytics. Moreover, SDC does not consider the problem of combining results of multiple carefully crafted queries to extract private information [25].

The first formalized privacy protection model which considers the challenge of outsourcing data for processing by remote parties was introduced as k-anonymity in 2002 [25]. Generally, k-anonymity can be achieved, e.g., by generalizing or suppressing identifying attributes until the data item does not differ anymore from the other $k-1$ items. However, neither k-anonymity nor its various variations achieve reasonable data utility while ensuring complete privacy [8].

Besides respondents privacy, two additional dimensions are distinguished when outsourcing sensitive data for external analytics today: the privacy of data owners and that of data users [5]. Owner privacy aims to protect both the data and associated knowledge, for example, data mining rules. This is the focus of privacy-preserving data mining which comprises a variety of methods from statistics and database theory to cryptography to allow sharing of data for analysis and publish the results without jeopardizing owner privacy [26]. Among these, differential privacy emerged to achieve strong privacy guarantees for statistical databases which does not depend on the background knowledge of an adversary (data linkage attacks) or her ability to perform series of random queries (database reconstruction attacks) [9]. Differential privacy considers the setting where a trusted database curator processes statistical queries of the users, using a randomized mechanism to ensure that the published results do not reveal if any single data item is part of the computation or not [10]. Several implementations have been proposed recently [10], e.g., Fuzz[4] and GUPT[5], however, the balance between data utility and privacy assurance remains a problem.

User privacy deals with hiding user access patterns to a remote data source, such as which particular item the user wanted to retrieve with the query. Private Information Retrieval (PIR) solves this problem with reasonable efficiency[6], though it does not scale well with the number of accessible records [5].

2.3 Advancements in Trusted Execution

Trusted Execution Environments (TEEs) allow the execution of software in such a way that the main operating system and other "untrusted" software outside the

[4] http://privacy.cis.upenn.edu/software.html.
[5] https://github.com/prashmohan/GUPT.
[6] http://percy.sourceforge.net/.

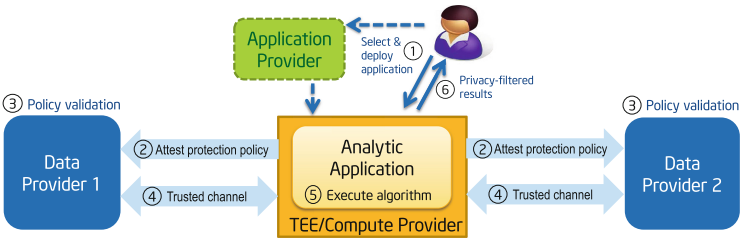

Fig. 1. Generic architecture of a Trust Brokerage solution.

TEE and its Trusted Computing Base (TCB) can neither violate the integrity of the performed computation nor the secrecy of processed data [1,11].

Considering this conservative definition, variations of TEE technology have been available and in fact widely deployed for many years in the form of secure co-processors, remote management interfaces and smartcards. However, these implementations are typically presented as fixed-function devices without an option for user programming.

With the rise of modern Trusted Computing technology, the benefits of controlled environments for verifiable (attested) code execution became more and more apparent. A significant body of research investigated the design and implementation of execution environments which would remain unaffected by security bugs in the "untrusted" or "non-secure" world and whose correct deployment, execution and outputs could be cryptographically confirmed by the underlying security infrastructure [17,21,24]). A variety of compelling usages have been examined under this model including secure online banking, credential storage, digital rights management and trusted virtual domains [2,18,24].

Today, we may be at the brink of a revolution in computer security as TEE technology becomes a wide-spread feature of computing platforms. Products such as ARM TrustZone and Texas Instruments M-Shield, which partition hardware in a simple secure/non-secure world view, are starting to embrace the concept of user-defined "trusted apps" which may be owned by different stakeholders and managed through a platform-independent API [11].

The Intel® Software Guard Extensions (SGX) [14,19] represent the next major step in this development. Intel® SGX enables users to run a large number of independent TEEs with only minor performance overhead [22]. It provides strong protection against software attacks including compromised hypervisors and platform firmware, and also defeats some common hardware attacks [1]. Application developers are offered a single TEE API across large segments of the computing spectrum, enabling modern software development and deployment practices such as "trusted app markets".

Following the trends in recent TEE research and attestation [6,15,20], we expect this development to continue down to low-end, resource-constrained devices, enabling an expansion of the trusted computing and TEE continuum across the whole IoT spectrum.

Table 1. Comparison of alternate approaches

	Secure multiparty computation	Data De-Identification	TEE solution
Solution Cost	design per application & data	design per data	design per application
Scalability	multi-party	multi-party	multi-party & interconnected
Data Utility	filters applied after computation	computation can be obstructed by filters	filters applied after computation
Performance	low	good	good
Assurance	high	good	good
Maturity	deployed in niches	widely deployed	new approach

3 TEE-based Trust Brokerage and Computation

We propose to leverage modern TEEs and attestation for trust brokerage and computing. As illustrated in Fig. 1, an analytics application running in a TEE would perform computation on data sourced from multiple, mutually untrusting data providers. The analytics application can be sourced from external software providers and executes in a neutral environment with strong protection against hardware and software attacks, while attestation and trusted channels enable negotiation and commit to security and privacy policies.

In more detail, we envision the following generic flow: (1) The user selects an analytics application, possibly from external application providers, and submits it for processing into the TEE of the compute provider. The application is bundled with one or more supported security and privacy policy options to be selected by the user. (2) The analytics application contacts the data providers, attesting to the application identity, and security and privacy configuration which were loaded into the TEE. (3) Before sourcing data to the analytics application, the data providers verify the information provided in the attestation report against the security and privacy policy associated with the requested data set. (4) If the request complies with the respective policies, the data is provided to the analytics application using a trusted channel. The data or channel may be subject to additional privacy protection filters depending on data protection policy. (5) The analytics application leverages TEE assurances to enforce the security and privacy policies while computing the result. Data providers may employ additional monitoring of requested data in order to validate the enforced protection policy. (6) On completion, the analytics application may apply additional privacy-filters as determined by the data protection policy before returning the results to the user.

Comparison of Approaches. Table 1 compares our TEE-based solution and previously pursued MPC and DDI approaches with regard to the requirements

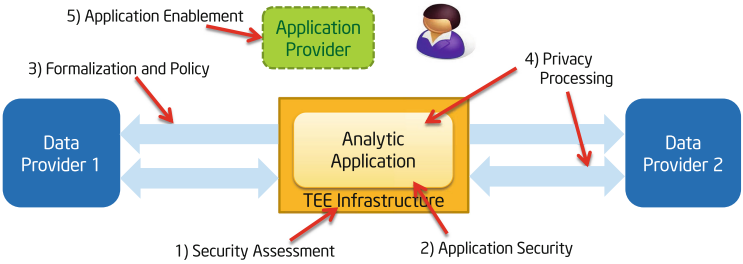

Fig. 2. Major challenges in TEE-based Trust Brokerage.

outlined in Sect. 1. As can be seen, MPC achieves high assurance by using well-established cryptographic techniques. Privacy filters can be applied after computation to provide privacy and confidentiality while maintaining high data utility. Unfortunately, MPC solutions incur a high performance penalty and require re-design for each application and data set. On the other hand, DDI solutions achieve good performance, however, data utility is lowered as the computation results may be distorted from privacy filters and designs cannot be generalized across data sets. In comparison, our TEE-based solution exhibits good performance as well as data utility. A lower solution cost is expected as applications and privacy filters can be ported once and re-used as enforced by TEE policy. Depending on the implementation, TEEs can deliver good assurance, however, additional research is required to analyze the security of TEEs in this scenario and mitigate possible attacks (see Sect. 4).

4 Research Challenges in TEE-based Trust Brokerage

While TEEs offer a scalable and efficient approach to trust brokerage, Fig. 2 points to a number of challenges which need to be addressed before a comprehensive solution can be achieved.

(1) Security Assessment. A number of TEE solutions have been proposed, and it can be expected that more products with different security properties will push into this market to cover the complete range of the computing spectrum. To maximize the use of this technology, it is necessary to investigate and assess their security properties and determine shortcomings based on the various possible usages, such as the resistance to side-channel attacks, malicious cloud providers and the level of isolation between multiple TEEs on the same platform.

(2) Application Security. While TEEs enable a major reduction of the trusted computing base, it must be expected that TEE applications will have their share of security vulnerabilities. Attackers will continue trying to trick users into installing trojan horses and making wrong decisions. Hence we must revisit known problems and solutions w.r.t. TEE applications, adopt modern defenses such as control-flow integrity and investigate the role that new deployment and attestation protocols might play.

(3) Formalization and Policy. An advantage of MPC and DDI is their ability to provide formally provable assurances. This helps reasoning about provided assurances and can be a basis for determining and negotiating abstract security policies. We require a similar formalization of TEE security properties to negotiate, assess and enforce a similar level of assurance.

(4) Privacy Processing. The assured execution provided by TEEs allows to enforce privacy filtering *after* the computation has been performed, making it less application dependent and also potentially enabling better privacy / data utility trade-offs. Processing larger data chunks or streams may also enable more efficient PIR schemes.

(5) Application Enablement. While prior work was focused on local TEE usages such as secure credential storage (see Sect. 2.3), we envision multi-party compute and collaboration services to enable a new secure cloud experience. Some applications may only require a generic TEE compatibility layer, while others will uniquely benefit from the deployment and policy negotiation technology supported by TEEs. Examples in this direction are the Contractual Anonymity System [23] and Verifiable Confidential Cloud Computing [22].

5 Conclusion

In this paper we propose TEE-based trust brokerage as a practical alternative to privacy-preserving multi-party computation. Recent advances in TEEs move us closer to a generic solution architecture that compares favorably with previous approaches in terms of efficiency, data utility and flexibility. A number of research challenges remain to be solved in order to meet security and scalability requirements; we detail these and suggest a path towards a comprehensive solution.

References

1. Asokan, N., Ekberg, J.E., Kostiainen, K., Rajan, A., Rozas, C., Sadeghi, A.R., Schulz, S., Wachsmann, C.: Mobile trusted computing. Proceedings of the IEEE **102**(8), 1189–1206 (2014)
2. Berger, S., Cáceres, R., Pendarakis, D.E., Sailer, R., Valdez, E., Perez, R., Schildhauer, W., Srinivasan, D.: TVDc: Managing security in the trusted virtual datacenter. Operating Syst. Rev. **42**(1), 40–47 (2008)
3. Bogetoft, P., Christensen, D.L., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., Nielsen, J.D., Nielsen, J.B., Nielsen, K., Pagter, J., Schwartzbach, M., Toft, T.: Secure multiparty computation goes live. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 325–343. Springer, Heidelberg (2009)
4. Dalenius, T.: The invasion of privacy problem and statistics production. an overview. Statistik Tidskrift **12**, 213–225 (1974)
5. Danezis, G., Domingo-Ferrer, J., Hansen, M., Hoepman, J.H., Métayer, D.L., Tirtea, R., Schiffner, S.: Privacy and data protection by design - from policy to engineering. Technical report, ENISA (2015)

6. Defrawy, K.E., Francillon, A., Perito, D., Tsudik, G.: SMART: Secure and minimal architecture for (establishing a dynamic) root of trust. In: Network and Distributed System Security Symposium (NDSS 2012). The Internet Society (2012)
7. Demmler, D., Schneider, T., Zohner, M.: Ad-hoc secure two-party computation on mobile devices using hardware tokens. In: USENIX Security Symposium, pp. 893–908. USENIX (2014)
8. Domingo-Ferrer, J., Torra, V.: A critique of k-anonymity and some of its enhancements. In: Conference on Availability, Reliability and Security (ARES 2008) (2008)
9. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
10. Dwork, C.: A firm foundation for private data analysis. Commun. ACM **54**(1), 86–95 (2011)
11. Global Platform: TEE system architecture v1.0 (2011). http://www.globalplatform.org/specificationsdevice.asp
12. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Symposium on Theory of Computing (STOC 1987), pp. 218–229. ACM (1987)
13. Hazay, C., Lindell, Y.: Constructions of truly practical secure protocols using standard smartcards. In: ACM CCS 2008, pp. 491–500. ACM (2008)
14. Hoekstra, M., Lal, R., Pappachan, P., Phegade, V., Del Cuvillo, J.: Using innovative instructions to create trustworthy software solutions. In: Hardware and Architectural Support for Security and Privacy (HASP). ACM (2013)
15. Koeberl, P., Schulz, S., Sadeghi, A.R., Varadharajan, V.: Trustlite: A security architecture for tiny embedded devices. In: European Conference on Computer Systems (EuroSys). ACM (2014)
16. Malkhi, D., Nisan, N., Pinkas, B., Sella, Y.: Fairplay – a secure two-party computation system. In: USENIX Security Symposium, pp. 287–302. USENIX (2004)
17. McCune, J.M., Li, Y., Qu, N., Zhou, Z., Datta, A., Gligor, V., Perrig, A.: TrustVisor: Efficient TCB reduction and attestation. In: Security and Privacy (S&P), pp. 143–158. IEEE (2010)
18. McCune, J.M., Parno, B.J., Perrig, A., Reiter, M.K., Isozaki, H.: Flicker: An execution infrastructure for TCB minimization. In: European Conference on Computer Systems (EuroSys), pp. 315–328. ACM (2008)
19. McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C.V., Shafi, H., Shanbhogue, V., Savagaonkar, U.R.: Innovative instructions and software model for isolated execution. In: Hardware and Architectural Support for Security and Privacy (HASP). ACM (2013)
20. Noorman, J., Agten, P., Daniels, W., Strackx, R., Van Herrewege, A., Huygens, C., Preneel, B., Verbauwhede, I., Piessens, F.: Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base. In; USENIX Security Symposium. USENIX (2013)
21. Pfitzmann, B., Riordan, J., Stüble, C., Waidner, M., Weber, A.: The PERSEUS system architecture. Technical report, RZ 3335 (#93381), IBM Research (2001)
22. Schuster, F., Costa, M., Fournet, C., Gkantsidis, C., Peinado, M., Mainar-Ruiz, G., Russinovich, M.: VC3: Trustworthy data analytics in the cloud using SGX. In: IEEE Security and Privacy (S&P 2015). IEEE (2015)
23. Schwartz, E.J., Brumley, D., McCune, J.M.: A contractual anonymity system. In: Network and Distributed System Security (NDSS). The Internet Society (2010)
24. Singaravelu, L., Pu, C., Haertig, H., Helmuth, C.: Reducing TCB complexity for security-sensitive applications: three case studies. In: European Conference on Computer Systems (EuroSys). ACM SIGOPS (2006)

25. Sweeney, L.: k-anonymity: A model for protecting privacy. Int. J. Uncertainty, Fuzziness Knowl.-Based Syst. **10**(05), 557–570 (2002)
26. Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P., Saygin, Y., Theodoridis, Y.: State-of-the-art in privacy preserving data mining. SIGMOD Rec. **33**(1), 50–57 (2004)
27. Yao, A.C.: How to generate and exchange secrets. In; Foundations of Computer Science (FOCS 1986). pp. 162–167. IEEE (1986)