Facilitating plausible deniability for cloud providers regarding tenants' activities using trusted execution

Dan O'Keeffe
Royal Holloway, University of London
daniel.okeeffe@rhul.ac.uk

Asma Vranaki
University of Bristol
asma.vranakis@bristol.ac.uk

Thomas Pasquier
University of Bristol
thomas.pasquier@bristol.ac.uk

David Eyers
University of Otago
dme@cs.otago.ac.nz

***Abstract*—A cloud provider that can technically determine tenants' operations may be compelled to disclose such activities by law enforcement agencies (LEAs). The situation gets even more complex when multiple LEAs across different jurisdictions are involved, e.g., because of the distributed locations of cloud servers and data storage. Yet cloud providers typically do not need or want to know about their tenants' activities, other than measuring how such activities incur expenses for using cloud resources.**

Thus mechanisms should be developed for cloud providers to have sufficient *plausible deniability* with regards to the processing being carried out by tenants on their platform, in jurisdictions that permit cloud providers to avoid liabilities in this way. Symmetrically, such mechanisms could protect tenants from legal over-reach, for example, when the country in which the cloud provider is incorporated could force disclosure of the processing carried out by cloud tenants.

But to what extent can cloud providers acquire plausible deniability? Current discussions regarding risk have focused on data confidentiality and integrity. We argue that processing operations can equally reveal sensitive information—such as trade secrets and business processes—and that for some classes of application both *data protection* and *algorithm protection* are necessary.

In this paper, we examine the legal and technical motivations for achieving plausible deniability in cloud interactions. We demonstrate the likely performance overhead of using containers secured with technologies such as Intel SGX. Further, we examine the current limitations of our proposed plausible deniability mechanisms, and outline a potential approach for enabling lawful access to enclaves subject to appropriate judicial oversight.

***Index Terms*—Enclave, Deniability, Legal, Cloud**

I. Introduction

Over the last decade cloud computing has become an integral part of most company IT provisioning. Nonetheless, in many cases usage of cloud computing is affected by concerns about the security and privacy of data and processing [1], [2], [3].

Cloud tenants must trust that cloud providers will carry out processing and storage of their data more effectively than tenants can themselves. In terms of technology, the major commercial cloud providers have access to vast economies of scale, and thus are highly likely to be able to provide services efficiently. There have been surprisingly few major security incidents affecting large cloud providers, most probably because they devote significant attention to security matters—the reputation of their brand is critical. However, the actual legal guarantees [4], [5] cloud providers give tenants may not be fully satisfactory.

Indeed, new laws such as the Clarifying Lawful Overseas Use of Data Act (Cloud Act [6]) in the US empower LEAs to demand access from entities subject to US jurisdiction to data stored in foreign jurisdictions. Cloud tenants can therefore have reasonable concerns their information (including sensitive data and business processes) could be disclosed to third-parties like LEAs. Similarly the China National Intelligence Law [7] requires companies and citizens to collaborate with the government.

A solution for these cloud tenants is to host data and run computing on cloud providers such that those providers have no useful visibility of the data or transformations of it. We will demonstrate how commodity trusted hardware within modern CPUs can solve this problem.

In this paper, however, we also highlight that the *cloud provider* should actually share their tenants' desire not to be able to see or interpret their tenants' data or processing, for a combination of legal and operational reasons (e.g., to reduce the costs associated with handling legal requests from LEAs). The aim is to support *plausible deniability* for the cloud provider in terms of the processing and storage carried out by tenants. We explore the likely overhead of a readily-available techni-

978-1-7281-1099-8/20/$31.00 ©2020 Crown
DOI 10.1109/IC2E48712.2020.00013

cal mechanism. Nonetheless, we highlight that plausible deniability is likely to be only partly achieved. We discuss what exposure cloud providers might continue to have, but also propose that the remaining exposure is likely to be manageable in the face of legal requests.

The rest of this paper is structured as follows: in the next section (§ II) we discuss the background knowledge underpinning our vision. § III explores the legal motivation in more detail. § IV describes our assumptions, particularly our supported security threat model. In § V we present our architecture and detail the components within it. We discuss key challenges in realising our vision in § VI. Finally § VII indicates future research that we plan to undertake, and concludes the paper.

II. Background

Secure cloud computing. There exist a variety of potential mechanisms for protecting data and computation on remote cloud infrastructure. Techniques such as database encryption [8] are straightforward to implement, but either do not protect data in memory or do not support general purpose applications [9]. More advanced homomorphic encryption techniques [10] allow for computation over encrypted data, but fully homomorphic encryption is prohibitively expensive. In this paper, we focus instead on hardware-enforced *trusted execution environments*, which can protect code and data of general purpose applications on remote cloud computers with reasonable overhead.

Early trusted execution proposals primarily leveraged widely available trusted platform module (TPM) chips [11], [12], [13], but either suffered from poor performance or did not provide full confidentiality. We focus instead on a recently available trusted execution technology for commodity CPUs (Intel SGX), which provides strong security guarantees for general-purpose applications with low overhead.

Intel SGX Enclaves. In 2015, Intel released the Software Guard Extensions (SGX) for their CPUs, which support the creation of trusted execution environments called *enclaves* [14]. Critically, an enclave shields application code and data from higher privileged software, including the OS, hypervisor and BIOS, by transparently encrypting and integrity protecting enclave memory pages at runtime. Enclaves also protect against attackers with physical access, assuming the CPU package is not breached. Enclaves therefore offer a promising solution for protecting the confidentiality and integrity of applications running on remote cloud infrastructure from incompetent or malicious employees of cloud providers.

Enclave memory protection. Enclave code and data reside in a region of protected physical memory called the enclave page cache (EPC). Only application code executing inside the enclave is permitted to access the EPC. Enclave code can access the memory outside the enclave. An on-chip memory encryption engine encrypts and decrypts EPC cache lines written to and fetched from memory. As enclave code is always executed in user mode, any interaction with the OS through system calls, e.g., for network or disk I/O, must execute outside of the enclave.

Enclave lifecycle. Developers can create enclave libraries (e.g., using Intel's SGX SDK [15]), that are loaded into an enclave and executed by a CPU with SGX support. Enclaves are created by untrusted code using the `ECREATE` instruction, which initialises an SGX enclave control structure (SECS) in the EPC. The `EADD` instruction adds pages to the enclave. When all enclave pages are loaded, the `EINIT` instruction creates a cryptographic measurement of the enclave contents. After enclave initialisation, an unprivileged application can execute enclave code through the `EENTER` instruction, which switches the CPU to enclave mode and jumps to a predefined enclave offset. Conversely, the `EEXIT` instruction causes a thread to leave the enclave. A developer defines the interface between the enclave code and other, untrusted application code: a call into the enclave is referred to as an *enclave entry call* (*ecall*); *outside calls* (*ocalls*) allow enclave functions to call untrusted functions outside.

Remote attestation. A remote party can verify the integrity of an enclave [16]. Based on the measurement during enclave initialisation, a dedicated quoting enclave signs the measurement using a secret CPU key. Intel provides an auxiliary attestation service to verify the validity of the signed measurements. Enclaves allow data to be written to persistent storage securely—a process known as sealing. Sealed data can be bound to a signing authority, which allows enclaves to persist state across reboots. Any enclave signed by the same authority can subsequently unseal it.

III. Legal Motivation

The legal landscape surrounding cloud computing is evolving quickly and is becoming increasingly complicated with multiple overlapping national, international and supranational regimes governing data privacy, cybercrime and LEA access to foreign data.[1] After years of outdated and inadequate data privacy laws[2], which were unfit for the modern digital age, many jurisdictions are now introducing new laws, such as the General Data Protection Regulation, which protect personal data rights. At the same time, jurisdictions like the US have also introduced new laws, such as the Cloud Act[3], to

[1] see Code de Procédure Pénale [C. Pr. Pén.] [Criminal Procedure Code] art. 57-1 (Fr.); Restatement (Third) of the Foreign Relations Law of the United States § 442; Article 48, GDPR; 9 Council of Europe Convention on Cybercrime 23 Nov. 2001, S. Treaty Doc. No. 11, 108th Cong., 1st Sess. (2003), 2296 U.N.T.S. 167 (Cybercrime Convention);

[2] e.g., Data Protection Directive, UK Data Protection Act 1998

[3] Clarifying Lawful Overseas Use of Data (CLOUD) Act, H.R. 1625, 115th Cong. div. V (2018) (enacted) (to be codified in scattered sections of 18 U.S.C.).

improve their access to data stored in other jurisdictions for purposes such as law enforcement.

In this section we will focus on Part 1 of the Cloud Act, its implications for cloud providers when it comes to granting access to cloud data to LEAs, and the potential legal ramifications of enclave technology for cloud providers. It is beyond the scope of this paper to explore the relationship between the Cloud Act, the GDPR and the Cybercrime Convention. We also do not consider Part 2 of the Cloud Act which enables the US to enter in executive data access agreements with foreign countries.

Cloud Act overview. Part 1 of the Cloud Act specifies that service providers, such as cloud providers, are required to disclose all data in their possession, custody, or control, pursuant to lawful process, regardless of its location[4] as long a number of conditions are met including:

- the US has persona jurisdiction over the target entity;
- the entity is an electronic communication service or remote computing service provider which falls within the ambit of the Cloud Act;
- the target entity has possession, custody or control over the data being sought;
- LEAs have to follow the legal process to obtain access to all data including establishing 'probable cause' for certain content.

Once these requirements are met, LEAs gain access to the data (e.g., communication contents, stored data and account information) in question by serving a warrant or subpoena on the cloud provider in question.

The target entity can challenge a warrant by arguing that one or more of the statutory conditions are not met. For example, a cloud provider can put forward a cogent case that it does not possess or have control or custody of stored encrypted data because the data or business processes are not accessible (see § V for how this can be technically enacted). This means that access to the data can only be provided by the cloud tenant who possesses the corresponding secret key.

The meaning of the phrase 'possession, custody or control' has been extensively litigated in other contexts in the US. Although it is straightforward to determine if the target entity has possession or custody of the data by evaluating whether it has physical possession of the data, the issue of control from a legal perspective is more complex. Typically in the US, courts apply one of two tests to determine if the cloud provider has 'control' of the data [17]. The first test is the 'practical ability' test which requires the courts to evaluate factors such as common ownership, common directorship, exchange of data in normal course of business and the financial relationship between the US and foreign company. If the courts find that an entity can demand or have access to certain data in the normal course of business, the presumption is that the data is within the control of the entity. The 'legal right' test is the second, stricter and less common test which the courts could apply to determine control.[5] Here, control is determined by looking at the legal entitlement of the entity over the data held by the foreign company.

The requirement of control, possession or custody can limit LEAs' access to cloud data. Likewise, LEAs have to follow the legal process to obtain access to the data under the Cloud Act. For example, LEAs have to meet certain standards of proof to obtain the customer information of a remote computing service or electronic communication service. These standards will depend on the type of information sought. LEAs also need to obtain a search warrant from a judge to access the content of electronic communications which have been in storage for less than 180 days. LEAs can only obtain a search warrant if they establish 'probable cause' or a 'fair probability' that evidence of crime can be found in the location on the basis of available evidence.[6] Access to this type of data cannot be secured unless this standard is met. Even for data stored longer than 180 days, LEAs can only access such data through a subpoena or court order which, while not requiring 'probable cause', do require LEAs to show connection between the information sought and a lawful investigation.

Legal implications of enclaves. Consequently, returning to the scenario where data or business processes are not intelligible to the cloud provider because an enclave-like technology is used, the cloud provider has an arguable case to challenge a warrant on a number of grounds. It could argue that although it has physical custody or possession of the data on its servers, it actually does not have 'control' over the data because it does not meet the requirements of the 'practical ability' test. For example, although the data is physically stored on its server, it cannot demand or have access to the stored data in an intelligible way in the normal course of business because of its contractual and technical arrangements with the cloud tenant.

At worst, a warrant or subpoena should target both the cloud provider and cloud tenant so that the LEAs can access the data or business processes in question in an intelligible manner. In case the courts rely on the stricter legal entitlement test to determine control, depending on the circumstances the cloud provider could argue that this test also fails because, for example, it is not legally entitled to access the data in intelligible form as per its contract with the cloud tenant. Our position is that

[4]H.R. 1625, 115th Cong. div. V, § 103(a) (2018) (enacted) (to be codified at 18 U.S.C. § 2713).

[5]United States v. Int'l Union of Petroleum & Indus. Workers, AFL-CIO, 870 F.2d 1450, 1452 (9th Cir. 1989).

[6]e.g., United States v. Perkins, 850 F. 3d 1109, 1119 (9th Cir. 2017).

the technical mechanisms provided by trusted hardware, such as Intel's SGX enclaves, can render pointless search warrants from LEAs that seek to have the cloud provider retrieve the code and/or data of their tenants.

One possible reaction that LEAs might have is to seek a ban or control on the use of technologies such as SGX enclaves, because of the potentially relevant information that enclaves render inaccessible (a similar debate resurfaces regularly regarding encryption [18], [19]). We are not aware of any push toward such bans, not least since such regulatory power would be well beyond any given LEA. Given that no single jurisdiction controls CPU designs, or the key organisations within the CPU industry, it seems unlikely that such restrictions would be effective.

Another option for LEAs would be to seek to have access to a backdoor mechanism into the SGX enclaves. However this again has the problem that Intel's operations span multiple jurisdictions, and are well beyond the scope of any given LEA. Because the security of CPUs is so fundamental to all computing it is unlikely that regulation to modify its operation will be easily effected. Furthermore, from the perspective of the cloud provider a backdoor mechanism most likely would not require any cooperation or additional effort, thereby offloading associated legal concerns and costs to Intel.

IV. Assumptions & Threat Model

This vision paper is based on the following assumptions:

- We assume that data processed by an enclave service is kept encrypted (or is non-sensitive) when at rest, and only decrypted within the enclave.
- We assume that enclave-like technologies can be made secure. Many vulnerabilities and side channels exist in the current Intel SGX implementation [20], [21], [22], however, such issues are related to decisions in hardware design and implementation, and do not disqualify such technology overall.
- As discussed in previous sections, we assume the cloud provider has an incentive to guarantee that they cannot access tenant data or business logic. The cloud provider business model relies on trust, and attempts to break the proposed service model are likely to damage the cloud provider's reputation beyond repair. This would inevitably result in a reduction of its customer base. Consequently, we consider cloud providers to be willing participants.
- An explicit non-goal are threats from determined state actors that bypass legal due process. We aim to isolate the activities of cloud tenants and providers, such that one cannot be targeted through the other. We do not want tenants to avoid justice, but for the judicial process to target tenants directly and to avoid information disclosure through judicial pressure on a third party (i.e., the cloud-provider).

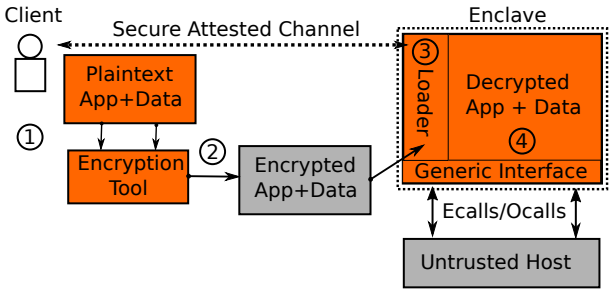

Fig. 1. Plausible deniability architecture.

V. Architecture & Proof of Concept

Having outlined our legal motivation for plausible deniability, in this section we introduce an architecture and workflow for achieving it (Figure 1). The architecture consists of both client-side and cloud components, where the client-side is assumed to be under tenant control.

The tenant initiates deployment at the client side ①, where it gathers the application code and data and then encrypts it using an encryption tool and a secret key ②. The tenant then uploads the encrypted code and data to the cloud, where a new enclave is created containing an unencrypted loader and the encrypted application code and data ③. The loader creates an attestation report using the Intel SGX attestation infrastructure, which the tenant uses to remotely attest the state of the enclave. If satisfied with the attestation, the tenant and loader establish a secure channel over which the decryption key for the application code and data is provisioned ④.

A. Key aspects of architecture

From the perspective of plausible deniability, there are several key aspects of the architecture.

Dynamic encrypted code loading: By default, code (and data) copied into the enclave during creation is visible to the untrusted host. In scenarios where a tenant wishes to hide its code from the cloud provider (e.g., to protect trade secrets), the tenant must encrypt its sensitive code and use a dynamic loader to decrypt it at runtime inside the enclave [23], [24]. The source code of the loader must be available to the tenant to allow it to verify the loader's behaviour and also to check the contents of the enclave during remote attestation. After decrypting the application code inside the enclave, the loader restricts page privileges to harden the enclave code against software vulnerabilities where possible (e.g., when JIT compilation is not required). However, dynamic encrypted code loading may increase the time taken to create a new enclave, which may be important for applications that rely on fast spawning of

new enclaves (e.g., serverless frameworks). We evaluate the overhead of encrypted code loading in § V-C.

Generic Enclave Interface: Our architecture is purposefully agnostic to the interface exposed by the enclave to the untrusted host. If the enclave includes the complete application [25], [26], [27], the interface will typically correspond closely to the OS system call interface (e.g., POSIX), or some subset of it. Alternatively, for applications concerned with reducing the trusted computing based (TCB) inside the enclave, a more specialised interface may be preferable [28]. We note however that for tenants wishing to hide the nature of their application through dynamic encrypted code loading, a more specialised interface may leak information through offline inspection of the declared ecalls and ocalls.

Convenient Application Construction: To minimise the reengineering effort required to run legacy applications within SGX enclaves, the architecture does not impose any restrictions on the toolchain used, except that it must be possible to encrypt and decrypt the output of the toolchain for dynamic loading.

Secure communication: To hide the data coming into and out of the application from the cloud provider, the tenant should terminate any encrypted communication channels inside the enclave (e.g., TLS endpoints [29], although other secure communication protocols may also be used).

B. Enclave Overheads

Shifting software into SGX enclaves will have a performance penalty, both in terms of the time required to load and start software, the time to read and write data contained within encrypted pages in main memory, and in terms of the cost of interacting with the underlying operating system.

Previous work has attempted to address some of these enclave performance challenges. For example, data partitioning schemes [28] allow some non-sensitive data to be stored unencrypted in main memory (still accessible to enclaves). SCONE demonstrated how the cost of operating system interactions can be reduced by passing data through main memory buffers to avoid hardware threads performing calls in and out of the enclave [25]. Thus from a tenant's perspective, SGX enclaves can achieve pratical performance for a variety of different application workloads.

From a cloud provider's perspective, the limited size of enclave memory on current SGX hardware remains a deployment barrier, since it must be virtualized across all tenants on a physical machine [30]. SGXv2's support for dynamic memory management [31] may help to mitigate this issue, as might future increases in EPC size, but this is still an area of active research.

C. Overhead from encrypted code loading

Given other aspects of enclave execution performance in cloud settings have been studied in previous work [25], [30], in this section we analyse additional performance overheads specific to plausible deniability for cloud providers. Concretely, we evaluate the potential overhead of dynamic encrypted code loading, and in particular its impact on enclave startup time.

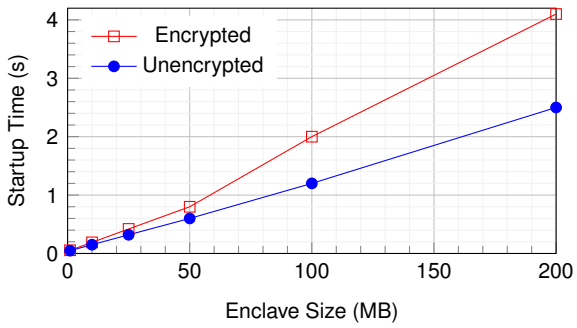

Fig. 2. Overhead from using encrypted enclave code

Experiment Setup: All experiments are conducted on an SGX-capable machine with 4 Intel(R) Xeon(R) CPU E3-1240 v5 3.50GHz cores (8 hyperthreaded) and 16GB RAM. Our test applications are implemented using the Intel SGX Linux SDK version 2.5, executing on Ubuntu Linux 16.04. We use the encryption tool and protected code loader provided by the SDK to encrypt test applications, and measure the time taken to create enclaves with decrypted applications of increasing size.

Results: Figure 2 shows the time taken to create an enclave as we increase the application size with both plaintext and dynamic encrypted code loading. For smaller enclave sizes, the overhead of decryption is minimal in comparison to the fixed costs of creating the enclave. For an enclave of 1MB, the overhead is 12%. However, as enclave size increases to 200MB, the overhead in terms of startup delay increases to 64%.

While the above results indicate a substantial performance impact for large enclaves, there are several potential mitigations. Firstly, for many applications enclave creation happens infrequently and is not on the critical path. Secondly, it should be possible to extend our loader to lazily decrypt code and/or data just before it is needed, avoiding the worst-case overhead in many scenarios.

VI. Challenges Faced in Achieving Plausible Deniability

A. Technical Challenges

Even with the proposed vision relying on Intel SGX to hide tenants' code and data from cloud providers, there are a number of remaining challenges for the community to tackle in terms of achieving practical plausible deniability.

Side-channel attacks: As noted above (see § IV), current SGX implementations have vulnerabilities due to side-channels [20], [21], [22]. Mitigating these attacks may involve significant performance impacts, on

the basis of mitigations seen so far for problems with speculative execution.

Were some side-channel vulnerabilities to remain unmitigated, it is not clear whether LEAs would request remote or even physical access in order to mount such side-channel attacks against the targets of warrants. Likewise, it would need to be tested whether LEAs would need to demonstrate that they could mount a side-channel attack in order to obtain a warrant in the first place. The same questions would also apply to fault injection attacks [32].

Behavioural fingerprinting: Dynamically loading encrypted enclave code helps protect the business logic of a tenant's application. However, without further effort, the externally observable behaviour of an application (e.g., memory access patterns, network interaction and other system calls) will often allow for identification of the *class* of application executing within the enclave, e.g., web server-like, VPN-like, etc. This could give probable cause for further legal requests. We assume tenants wishing to further mask the behaviour of their application will implement application-specific hardening techniques (e.g., by imitating a different class of application or using anonymous communication mechanisms such as TOR [33] or Vuvuzela [34] to hide network interactions).

The challenge is to assess whether it is worth a cloud provider trying to avoid warrants based on their computational infrastructure, if warrants can be served on them regarding records of their network activity.

B. Socio-ethical Challenges

One of the argument against the development of the proposed solution is the fact that this could help malicious individuals to deploy applications that escape control of law enforcement agencies and other governmental bodies. We believe those fears to be as misguided as those targetted towards encryption technology (e.g., see recent noise over DNS over HTTPS). In the rest of this section, we discuss why it may be legitimate to force LEAs to interact directly with tenants rather than access being granted through the underlying cloud provider.

First of all, it is important to keep in mind that one person's "benign government" can be someone else's "malicious state actor". As discussed in § III, a cloud provider may be forced by law to disclose information to state actors. This may be not be in the interest of the cloud provider nor the cloud tenant. We do argue that the majority of cloud providers would prefer to be considered as providing a sort of "utility service", where the tenant should be the one legally responsible and targetted by a disclosure request. The current situation puts cloud providers and cloud tenants in a clearly uncomfortable situation, especially as the international regulation is evolving relatively quickly. For example, the current US government and emerging legislation are causing major concerns in the European Union which in turn is creating push back from US actors (so called European "digital sovereignty" [35]). In this context, the proposed approach provides a technical solution that alleviates some of those concerns (please see technical caveats discussed in § VI-A).

Second, the request for data access by LEAs can be opaque and not open to public scrutiny. When an LEA wants access to some data it generally must go through a judge who creates a sealed order to be presented to the cloud provider. The cloud provider may challenge the sealed order or comply. The sealed order may be unsealed and revealed to the public after a certain amount of time as defined during its issue. However, a report [36] has shown that no systematic mechanism exists to unseal orders. Which means that a tenant may never discover that it has been targetted by an LEA. Further, as the order is sealed and may remain so, it is impossible to ensure that the cloud provider and the LEA are working within the remit of the court order. This ties back to fear, for example, in the European Union that American "giants" may be to willing to cooperate with US LEAs, potentially subjecting European citizens and companies to surveillance.

VII. Conclusion

There is a pragmatic basis for a cloud provider to seek to achieve plausible deniability about the specific code and data of their tenants, and thus avoid LEAs' warrants for access to the cloud provider's systems. This avoids risks to the provider of needing to resource answering LEAs' requests, and the risk of damage to their image, globally.

Trusted hardware such as Intel SGX supports a cloud software architecture we have presented and demonstrated, which can facilitate tenants' code and data remaining encrypted, from the perspective of the cloud provider. Further, based on our experiments on an initial prototype system, the startup overhead of encrypting code as well as data ranges from 16% to 64% depending on the size of the code and data to load into the enclave. We are confident that further engineering effort would significantly improve this performance.

We discuss legal perspectives regarding recent cloud regulation and note that while warrants prepared for LEAs will be rare, it is likely that courts will accept the futility of trying to access purely encrypted data, should the cloud provider use the architecture proposed in this paper. Courts will likely target tenants directly, thus simplifying the cloud provider's role.

Finally, we discuss some research challenges that remain to be solved, in order to significantly strengthen the nature of plausible deniability that cloud providers can achieve, in practice.

REFERENCES

[1] C. Esposito, A. Castiglione, B. Martini, and K.-K. R. Choo, "Cloud manufacturing: security, privacy, and forensic concerns," *IEEE Cloud Computing*, vol. 3, no. 4, pp. 16–22, 2016.

[2] I. Arpaci, K. Kilicer, and S. Bardakci, "Effects of security and privacy concerns on educational use of cloud services," *Computers in Human Behavior*, vol. 45, pp. 93–98, 2015.

[3] Z. Tari, X. Yi, U. S. Premarathne, P. Bertok, and I. Khalil, "Security and privacy in cloud computing: vision, trends, and challenges," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 30–38, 2015.

[4] S. Bradshaw, C. Millard, and I. Walden, "Innovation and Intellectual Property Rights," in *Cloud Computing Law*, C. Millard, Ed. Oxford University Press, 2013.

[5] W. K. Hon, C. Millard, and I. Walden, "Negotiated Contracts for Cloud Services," in *Cloud Computing Law*, C. Millard, Ed. Oxford University Press, 2013.

[6] A. A. S. Winston Maxwell, Mark W. Brennan, "Demystifying the U.S. CLOUD Act: Assessing the law's compatibility with international norms and the GDPR," https://www.hoganlovells.com/~/media/hogan-lovells/pdf/2019/2019_01_15_whitepaper_demystifying_the_us_cloud_act.pdf, accessed: 26/05/2019.

[7] National People's Congress, "National Intelligence Law of the People's Republic of China," https://www.chinalawtranslate.com/en/%E4%B8%AD%E5%8D%8E%E4%BA%BA%E6%B0%91%E5%85%B1%E5%92%8C%E5%9B%BD%E5%9B%BD%E5%AE%B6%E6%83%85%E6%8A%A5%E6%B3%95/, 2017, unnofficial translation into English.

[8] R. A. Popa, C. 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*. ACM, 2011, pp. 85–100.

[9] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 644–655.

[10] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 79, 2018.

[11] S. Bajikar, "Trusted platform module (TPM) based security on notebook PCs-white paper," *Mobile Platforms Group Intel Corporation*, pp. 1–20, 2002.

[12] R. Perez, R. Sailer, L. van Doorn *et al.*, "vTPM: virtualizing the trusted platform module," in *USENIX Security Symposium*, 2006, pp. 305–320.

[13] C. Chen, H. Raj, S. Saroiu, and A. Wolman, "cTPM: A cloud TPM for cross-device trusted applications," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI'14)*, 2014.

[14] V. Costan and S. Devadas, "Intel SGX explained." *IACR Cryptology ePrint Archive*, vol. 2016, no. 086, pp. 1–118, 2016.

[15] Intel, "Intel Software Guard Extensions SDK," https://software.intel.com/en-us/sgx/sdk, Accessed June 2019.

[16] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for CPU based attestation and sealing," in *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy*, vol. 13. ACM New York, NY, USA, 2013.

[17] J. D. Jordan, "Out of control federal subpoenas: When does a nonparty subsidiary have control of documents possessed by a foreign parent," *Baylor L. Rev.*, vol. 68, p. 189, 2016.

[18] R. L. Rivest, "The case against regulating encryption technology," *Scientific American*, vol. 279, no. 4, pp. 116–117, 1998.

[20] 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," in *Conference on Computer and Communications Security (CCS'17)*. ACM, 2017, pp. 2421–2434.

[19] C. Everett, "Should encryption software be banned?" *Network Security*, vol. 2016, no. 8, pp. 14–17, 2016.

[21] J. V. Bulck, M. Minkin, O. Weisse, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, T. F. Wenisch, Y. Yarom, and R. Strackx, "Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018, p. 991–1008.

[22] M. Schwarz, M. Lipp, D. Moghimi, J. V. Bulck, J. Stecklina, T. Prescher, and D. Gruss, "Zombieload: Cross-privilege-boundary data sampling," *CoRR*, vol. abs/1905.05726, 2019. [Online]. Available: http://arxiv.org/abs/1905.05726

[23] J. Seo, B. Lee, S. M. Kim, M. Shih, I. Shin, D. Han, and T. Kim, "SGX-Shield: Enabling address space layout randomization for SGX programs," in *24th Annual Network and Distributed System Security Symposium, (NDSS)*, 2017.

[24] T. Hunt, Z. Zhu, Y. Xu, S. Peter, and E. Witchel, "Ryoan: A distributed sandbox for untrusted computation on secret data," *ACM Trans. Comput. Syst.*, vol. 35, no. 4, pp. 13:1–13:32, Dec. 2018.

[25] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'keeffe, M. L. Stillwell *et al.*, "SCONE: Secure Linux containers with Intel SGX," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 689–703.

[26] A. Baumann, M. Peinado, and G. Hunt, "Shielding applications from an untrusted cloud with Haven," *ACM Transactions on Computer Systems (TOCS)*, vol. 33, no. 3, p. 8, 2015.

[27] C.-C. Tsai, D. E. Porter, and M. Vij, "Graphene-SGX: A practical library OS for unmodified applications on SGX," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, 2017, pp. 645–658.

[28] J. Lind, C. Priebe, D. Muthukumaran, D. O'Keeffe, P.-L. Aublin, F. Kelbert, T. Reiher, D. Goltzsche, D. Eyers, R. Kapitza *et al.*, "Glamdring: Automatic application partitioning for Intel SGX," in *2017 USENIX Annual Technical Conference (USENIX ATC '17)*, 2017, pp. 285–298.

[29] P.-L. Aublin, F. Kelbert, D. O'Keeffe, D. Muthukumaran, C. Priebe, J. Lind, R. Krahn, C. Fetzer, D. Eyers, and P. Pietzuch, "TaLoS: Secure and transparent TLS termination inside SGX enclaves," *Imperial College London, Tech. Rep*, vol. 5, 2017.

[30] T. Dinh Ngoc, B. Bui, S. Bitchebe, A. Tchana, V. Schiavoni, P. Felber, and D. Hagimont, "Everything you should know about Intel SGX performance on virtualized systems," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 1, Mar. 2019.

[31] B. C. Xing, M. Shanahan, and R. Leslie-Hurd, "Intel® Software Guard Extensions (Intel® SGX) software support for dynamic memory allocation inside an enclave," in *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*, ser. HASP 2016. New York, NY, USA: Association for Computing Machinery, 2016.

[32] A. Tang, S. Sethumadhavan, and S. Stolfo, "CLKSCREW: Exposing the perils of security-oblivious energy management," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 1057–1074.

[33] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[34] J. Van Den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, "Vuvuzela: Scalable private messaging resistant to traffic analysis," in *Proceedings of the 25th Symposium on Operating Systems Principles*. ACM, 2015, pp. 137–152.

[35] C. Stupp, "European Cloud Project Draws Backlash From U.S. Tech Giants," in *Wall Street Journal*, 2019.

[36] C. Parsons and A. Molnar, "Government surveillance accountability: The failures of contemporary canadian interception reports," *Canadian Journal of Law and Technology*, vol. 16, no. 1, 2018.