Challenges of Using Trusted Computing for Collaborative Data Processing

Paul Georg Wagner[1(✉)], Pascal Birnstill[2], and Jürgen Beyerer[1,2]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
paul.wagner@kit.edu
[2] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany

Abstract. In recent years many business processes have become more interconnected than ever before. Driven by the advance of the Internet of Things, companies rely on complex data processing chains that span over many collaborating corporations and across different countries. As a result of this development, automated data acquisition and collaborative data usage is now a foundation of many innovative and successful business models. However, despite having a clear interest in sharing valuable data with other stakeholders, data owners simultaneously need to protect their assets against illegitimate use. In order to accommodate this requirement, existing data sharing solutions contain usage control systems capable of enforcing policies on data even after they have been shared. The integrity of these policy enforcement components is often monitored by a trusted platform module (TPM) on the data receiver's side. In this work we evaluate the adequacy of TPM-based remote attestation for protecting shared data on foreign systems. In order to do so we develop an attacker model that includes privileged system users and expose attack vectors on TPM-protected data sharing applications. We show that TPMs do not provide sufficient protection against malicious administrators from competing stakeholders. Finally, we describe the advantages of using Intel's Software Guard Extensions (SGX) to protect shared data in hostile environments and propose an enhanced system architecture that includes both SGX enclaves as well as a classical TPM.

Keywords: Trusted computing · Trusted platform modules · Software guard extensions · Usage control · Policy enforcement · Data sharing

1 Introduction

Ever since data have become invaluable assets in many modern business processes, preventing malicious attackers from accessing critical information is a major challenge. In the past, data protection efforts mostly consisted of securing internal information processing infrastructure, like corporate computer systems

S. Mauw and M. Conti (Eds.): STM 2019, LNCS 11738, pp. 107–123, 2019.
https://doi.org/10.1007/978-3-030-31511-5_7

and databases. Today, with complex data processing chains spanning over multiple collaborating corporations and across different countries, this isolated view on data security is not sufficient anymore. Especially many industrial use cases, such as the joint operation of production equipment and support of complex service agreements, require the flexible exchange of information between different stakeholders. Hence one of the most urgent security requirements in many modern applications is to monitor and control the usage of sensitive information, even after it has been transmitted to other stakeholders.

In general, data sharing solutions connect the databases of participating corporations and provide mechanisms to securely exchange data across corporate boundaries. While many of these systems are still heterogeneous in nature, there are ongoing attempts to consolidate common standards and governance models into a single trusted business ecosystem [9]. This results in a virtual data space that is responsible for controlling and securing the data sharing process across multiple participating corporations. Furthermore, data owners are often allowed to specify restrictions on how data receivers may use the disclosed information. In most cases this is achieved by distributing usage control policies alongside the original data. These usage rules are evaluated and applied by policy enforcement components that run on the data receiver's systems. Since the data receiver is motivated to bypass the imposed usage restrictions, it is necessary to remotely verify the integrity of these usage control components before transmitting sensitive information. Usually data owners rely on trusted platform modules (TPMs) [13] to establish a trusted software stack on the remote side. By executing the TPM-backed remote attestation protocol and thereby verifying the software stack of the target system, access to shared data can be limited to trustworthy (i.e. unmodified and sufficiently protected) data receivers.

In this work we evaluate the adequacy of TPM-based remote attestation for protecting shared data on foreign systems. We do this by assessing the current architecture of a particular data sharing solution, the Industrial Data Space, in this regard. Section 2 briefly describes how the Industrial Data Space architecture applies usage control components and TPMs in order to ensure data sovereignty across multiple corporations. In Sect. 3 we then develop an attacker model that includes privileged system users such as administrators and outline the problems that arise when TPMs are used to protect data on foreign systems. Finally, in Sect. 4 we propose an enhanced system architecture that uses the capabilities of Intel's Software Guard Extensions (SGX) alongside a TPM to protect cross-domain data flows even against malicious administrators. The identified security problems as well as the proposed architectural improvements apply for the specific case of the Industrial Data Space, as well as for any generic data sharing system that relies on TPMs to establish a trusted computing base.

2 Data Sovereignty in Collaborative Data Spaces

The *Industrial Data Space* [9] is a virtual data space intended to automate data sharing for smart business ecosystems while simultaneously preserving data

sovereignty among its participants. In order to connect to the data space, participating corporations operate access points (called *connectors*) in their own respective IT infrastructure. Data space connectors can operate both as data provider and consumer, simultaneously sending and receiving information. They query remote data providers (i.e. other connectors) and are responsible for managing and conducting the subsequent data exchange. After processing the received information, the results are shared with neighboring connectors and serve as input for the next data processing step. That way complex data processing chains can be established across multiple collaborating corporations.

The data protection capability of the Industrial Data Space is based on a comprehensive usage control infrastructure that can monitor and govern shared data on foreign connectors. Unlike classical access control, usage control models focus on managing the future usage of data [10]. With usage control technology it is possible to restrict the processing and distribution of sensitive information even after it has been disclosed to other stakeholders. In the Industrial Data Space, data providers can define usage rules for their assets by specifying appropriate usage control policies. Whenever an outgoing data flow occurs, these policies are deployed on the receiving connector before transmitting any sensitive information. On the remote system a policy decision point (PDP) evaluates the received usage control policies and a policy enforcement point (PEP) enforces the specified rules on the shared data. Usually the PDP is included in the connector, while the PEPs are part of the data processing applications. Whenever sensitive information from another system is used by an application, its PEP generates an event that describes the specific data usage, sends it to the PDP and enforces the resulting decision. That way the usage control components on the data receiver's system ensure compliance with the usage restrictions specified by the data owner. Furthermore, the usage control components share data flow information across communicating connectors and hence constitute a distributed usage control infrastructure [7]. By specifying appropriate usage control policies, data providers can enforce complex usage strategies on their data, such as temporary or locally restricted access, even after they have been shared.

Any implementation of such a distributed usage control system has to make several assumptions. Most notably, the usage control components must not be maliciously manipulated or deactivated during their lifetime by either an internal or external attacker. Since it is the operating system's responsibility to protect the usage control components from any outside influence, we have to assume that it is implemented correctly and does not contain security-critical bugs. Furthermore, operating a distributed usage control system requires a mechanism to remotely verify the integrity of the remote protection components. In particular, the integrity of the remote connector, the data processing applications and the foreign usage control components has to be verified prior to a data flow. Additionally, transmitted data have to be encrypted in a way that only a trustworthy connector (and by extension trustworthy applications) can read them. Only if these requirements are fulfilled, the data provider can be sure that his usage control policies will be enforced correctly by the remote connector.

The connectors use trusted platform modules (TPMs) to establish a trusted computing base and meet these requirements. A TPM is a dedicated hardware chip that extends a computer with basic security related features [13]. It uses volatile platform configuration registers (PCRs) to measure the current hardware and software configuration as an unforgeable hash. This allows the data provider to seal confidential information to a certain TPM state. Furthermore, the data provider can use a remote attestation protocol to verify that the target system is in a trustworthy state before transmitting sensitive information. When a data provider is requested to share assets, his connector initiates the execution of the remote attestation protocol by transmitting a randomly drawn nonce to the remote side. The requesting connector then uses his TPM to generate a quote that contains this nonce and the current PCR values of his system. The quote serves as proof of the current system state and has been signed by the TPM with an attestation identity key (AIK). The AIK is an asymmetric cryptographic key pair that has been created by the TPM during a prior enrollment phase. While the public part of the AIK is known to all involved parties (usually it is certified by a CA), the private key never leaves the TPM. The signed quote is then transmitted back to the data provider, who verifies both nonce and signature, before confirming that the included PCR values belong to the expected, unmodified connector system. If these checks are successful, the data provider is convinced that the remote connector is in a trustworthy state, since only the TPM of a correctly configured system could have generated such a signed quote. After the remote attestation protocol executed successfully, the data provider issues usage control policies to the attested data consumer and finally initiates the requested data flow. To prevent eavesdropping, the provider encrypts the transmitted data with the public part of an ephemeral key pair that has been generated by the trusted application during the remote attestation process. The trusted application authenticates the public key by including it in the signed quote as well. Figure 1 shows how a connector executes the remote attestation protocol and deploys necessary usage control policies before allowing data access. A trusted third party (TTP) is responsible for providing the known "good" PCR values that are compared to the values in the quote. Also, we assume the existence of a CA that certifies the public keys of all involved parties.

The remote attestation protocol enables data space connectors to establish trust in systems that are operated by competing corporations. In combination with a distributed usage control model, the Industrial Data Space architecture allows corporations to safeguard their assets across data processing chains that leave their own IT infrastructure. In the following section we base our analysis on the presented reference architecture. However, the identified problems are applicable to any generic data sharing system that uses TPM-based remote attestation to protect transmitted information on possibly hostile systems.

3 Attack Vectors

The advantage of using TPMs to establish a trusted computing base lies in their low cost, widespread availability and uncomplicated application to many prob-

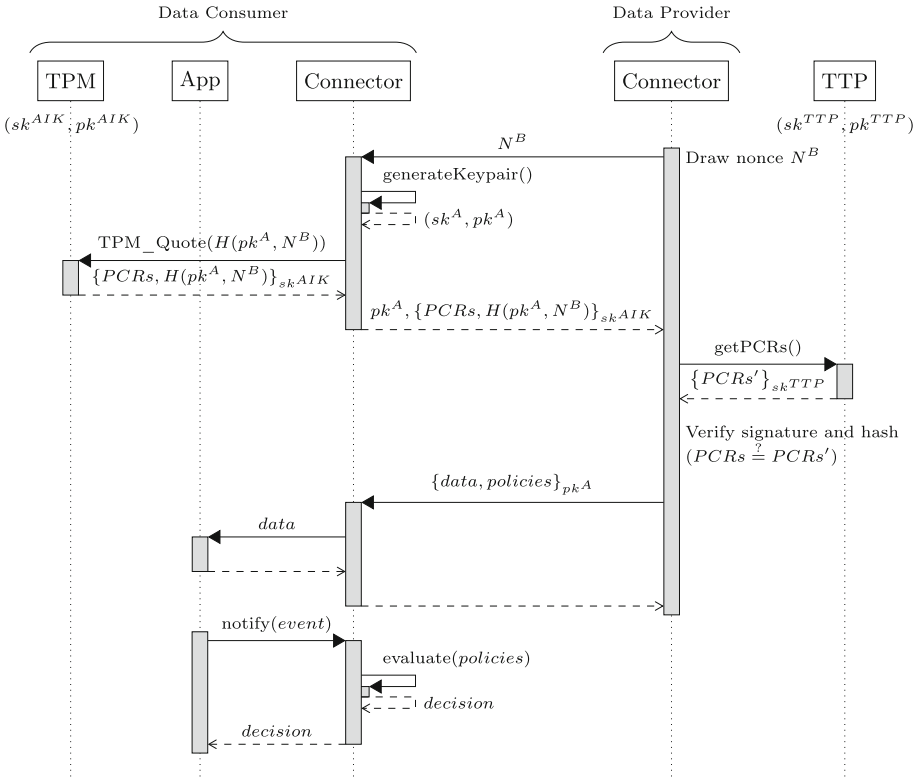

Fig. 1. Industrial data space connectors conduct remote attestation, data sharing and policy enforcement. N^B is a randomly drawn nonce. H is a hash function.

lems in the realm of trusted computing. However, there are drawbacks when using TPMs to protect sensitive information in foreign organizations. For example, distributed usage control systems can become insecure when the attacker model includes valid users of the attested system itself [14]. This is because malicious users have physical access to the TPM and can use it to decrypt previously intercepted data, as long as the PCR values do not change. Similar problems also occur in our use case. In order to point out the existing attack vectors regarding the use of TPMs, we first define a suitable security model by specifying the main attacker faced in the described scenario. Then we identify three different attack vectors on TPM-based data space architectures in general.

3.1 Security Model

The primary objective of a virtual data space is to secure the confidentiality of data that have been shared by a data provider. However, in our case the actual protection level that should be reached is specified by the deployed usage control policies. Depending on the implementation of the decision point, data

providers can specify with fine granularity what constitutes as legitimate data usage. Hence the main security goal is to protect the integrity of policies across connectors and enforce them remotely on data consumers.

The main attacker in our scenario is a malicious data consumer who wants to illegitimately use shared data without being subject to any usage restrictions imposed by the data owner. An example for this attacker is a corporation that participates in the data space with the intent to resell the received information outside the controlled data space. In order to do so, the attacker has to bypass his own connector's usage control enforcement components and extract the received information from the virtual data space. Since the malicious corporation operates the attacked connector in its own infrastructure, it can instruct the system administrator to tamper with the installed protection mechanisms. As a result, the strongest attacker faced in this scenario is an administrator who tries to bypass the TPM-based protection mechanisms that the data provider verifies before sharing his data. While companies must always trust their own administrators with regard to the managed systems, in this case the administrator acts as an attacker against the interests of other organizations.

3.2 Manipulating Connectors

An obvious way for attackers to extract information from the virtual data space is to manipulate the connectors that are running in their own infrastructure. For example, the attacker could disable the usage control components or disrupt the policy enforcement mechanisms. However, since the connector systems are measured by a TPM, these modifications will manifest themselves in changed PCR values. Hence the data provider is able to identify tampered systems by executing the remote attestation protocol. A successful attacker would have to either forge the PCR values on his connector (which he has physical access to), forge a quote signature, or exploit a vulnerability in the remote attestation protocol to convince the data provider, even though the PCR values are bad.

Due to the nature of the TPM as a hardware based trust anchor, these attacks are infeasible. The TPM is designed to only extend PCRs with new measurements, it is not possible to set them to a desired value [13]. Furthermore, the security of the TCG specification has been evaluated thoroughly [3,6,12]. Assuming that the attacker cannot break commonly used digital signature algorithms, forging a quote requires the attacker to obtain the private part of the attestation identity key. However, this key is generated and managed by the TPM, which does not reveal the private key to the outside world. Nonetheless it is important for the attesting party to include and verify randomly drawn nonces in the quote (c.f. Fig. 1). Otherwise the attacker can intercept a correct quote and replay it later, instead of forging a quote signature. In general, the TPM-based remote attestation secures the integrity of data space connectors and prevents attackers from gaining illegitimate data access by directly manipulating them.

3.3 Duplicating Attested Connections

Since the TPM protects the connector software against tampering, a successful attacker has to extract data without previous manipulation. As described before, in our scenario the attacker operates and controls the connector system. Hence the attacker can use that system to establish additional attested connections to data providers, unbeknownst to the still running and unmodified connector software. For this attack, the adversary first launches a legitimate connector system. Since the connector software is not manipulated, the subsequently conducted remote attestations will be successful. However, no useful information can be intercepted because any transmitted data are encrypted with an ephemeral key unknown to the attacker (c.f. Fig. 1). At this point the attacker launches a separate process that itself initiates the remote attestation protocol and establishes another connection with the data provider. This new instance of the attestation protocol will also succeed, since the PCRs of the targeted connector are still correct. However, this time the attacker controls the connection (i.e. chooses the ephemeral session key) and may receive sensitive information from the unsuspecting data provider, without being subject to usage control enforcement. This attack succeeds because in general data providers cannot distinguish establishing a connection with the legitimate software from communicating with an attacker-controlled process on an otherwise unmodified connector system. If the attacker simultaneously blocks the network traffic of the legitimate connector process, the data providers do not even notice any additional connection attempts. As a result, attackers with access to an unmodified connector system can bypass all protection mechanisms by impersonating a data consumer and requesting information from data providers.

A possible solution is to regard the connector system as untrustworthy as soon as any process other than the connector software initiates an attested connection. On a technical level this would require that creating a new attested connection to a remote connector invariably triggers a measurement and extends the PCRs. In that case the additional attestation fails and the additional connection would not be established. Even though a trusted operating system could accomplish this by monitoring the network interfaces, consequently there would be a very large set of constantly changing, yet valid PCR values to verify. From an architectural point of view, this attack is possible because the used attestation protocol can only identify the whole attested system as an endpoint for the communication, but not single processes or users on that system. In other words, by using the TPM-based remote attestation, data providers can only make sure that their data is transmitted to a remote system that is in a specific state (i.e. has a certain TPM with certain PCR values), but they have no means of verifying who receives the information on the attested system. This problem cannot entirely be avoided by relying solely on TPM technology.

3.4 In-memory Tampering

As presented earlier, the TPM is responsible for protecting the connector against outside manipulation. This mechanism works by making malicious modifications

of the connector implementation transparent to the data provider. When the connector system boots up, the TPM constructs a chain of trust that begins at the Core Root of Trust for Measurement (CRTM) and includes the BIOS as well as parts of the operating system. Ultimately, the connector's executable and configuration files are also measured and extended to the PCRs. During the subsequent remote attestation, the data provider precludes manipulations by verifying the PCR values. However, this procedure can only reveal connector modifications that occur before or during its launch. Once the connector is running, no more measurements are conducted and the PCRs do not change anymore. Since our attacker has administrative rights on the connector system, he can attach a debugger instance to the connector process and access its memory layout without changing the verifiable state of the system (i.e. the PCR values). Even simple tools from the GNU Compiler Collection like `gdb` and `dump` suffice for carrying out this attack. By directly accessing the connector's memory, an attacker can read out and manipulate confidential data that should be subject to usage restrictions enforced by the connector. The attacker can also tamper with the loaded code of both the connector and the data processing applications.

In order to address these types of attacks and allow the attestation of remote systems to be more flexible, measurements of executed applications can be automatically triggered during runtime. OS-based integrity measurement mechanisms, like the *Integrity Measurement Architecture* (IMA) on Linux, can appraise the integrity of data and executable files by comparing their hashes against prepared lists. Furthermore, the IMA can trigger TPM measurements while the system is running. For example, it is possible to measure the content of opened files as well as the memory image of every starting application and extend the PCRs accordingly. This allows to attest a system very precisely. In our case, a correctly configured IMA could detect the launch of a debugging tool and announce it to verifying parties by conducting an appropriate TPM measurement. However, this can lead to considerable side effects when operating connectors. For instance, when using the IMA in that manner, every starting application inevitably changes the PCR values. As a result there is a very large set of trustworthy PCR values, and validating them during remote attestation can be cumbersome. Furthermore, the IMA only measures the initial memory image of the loaded application. It is still possible to retrospectively modify memory regions of a running application without influencing the PCRs. As Sparks shows, this can be done by carefully manipulating the page tables of a running process [12]. D'Cunha proposes a countermeasure against this attack by continuously measuring the virtual address space of individual processes with each write access [2]. Nevertheless, continuously measuring a complete memory dump of a complex application is hardly feasible in practice. Apart from that, the data provider would also have to continuously keep probing the consumer's systems in order to re-verify the PCRs and detect any wrongdoing. In addition, this countermeasure only prevents an attacker from manipulating the connector's main memory without also influencing the PCRs. It is still possible for the attacker to simply read out sensitive information directly from memory without the origi-

nal data provider noticing. Afterwards the attacker can use and redistribute the stolen data without any usage control policies being enforced on them.

In summary, there are strong attack vectors on systems that use TPM-based remote attestation to protect data sharing applications. The main cause of the described problems is that on the data consumer's side the operating system is still responsible for protecting transmitted information. However, privileged users who act as an attacker can evade many OS-level protection mechanisms such as address space isolation. Usually administrators are not viewed as attackers in many scenarios, because in general they have to be fully trusted with regard to their employer's systems. But as soon as distributed use cases are considered, for example in the context of the Industrial Data Space, administrators have to be viewed as attackers who try to evade usage restrictions that are imposed by competing companies. Since TPMs cannot sufficiently protect against this type of attack, other technologies have to be considered as well.

4 SGX in Collaborative Data Spaces

Intel's Software Guard Extensions (SGX) consist of a set of processor instructions extending the x86 architecture, along with hardware security modules that are included in newer Intel CPUs. SGX can provide a trusted execution environment for security critical applications, even if privileged software such as the operating system or a hypervisor is malicious. This is achieved by executing code as a protected container called *enclave*, which cannot be accessed by other processes, administrators, or even by the operating system itself. The enclave is protected by trusted hardware and is isolated from the rest of the system (reverse sandboxing). SGX allows to encapsulate critical software, for example cryptographic libraries or key management services, in protected shells that will behave in expected ways. Architectural details of SGX and a thorough analysis of its security are provided in [1]. Since then several attacks on some parts of the comprehensive SGX architecture have been revealed, including side-channel attacks [4] and a vulnerability related to Spectre [8]. However, countermeasures against these attacks have also been proposed [5]. Overall SGX is still regarded as secure and is being used in an increasing number of projects.

Whenever an SGX enclave is launched, its code and initial data are cryptographically hashed. This hash is called the enclave's *measurement*. A remote third party can verify the state of a running enclave by requesting a signed quote that includes the enclave's measurement. The quote can be verified by contacting the Intel Attestation Service (IAS) and comparing the attested measurement to a desired value. This ensures that the loaded enclave code has not been manipulated before execution, and hence establishes trust in the remotely running enclave. Furthermore, this attestation mechanism establishes a secure channel between the verifying party and the enclave using a modified Sigma protocol that includes a Diffie-Hellman key exchange. If the quote verifies correctly, the remote party is convinced that he communicates with the right enclave (measurement is correct) and that only this enclave instance knows the established shared secret.

Both parties can then derive a symmetric secret from the Diffie-Hellman key and use it to encrypt their further communication. In [1] the remote attestation protocol is explained in greater detail. A similar protocol is also possible between two enclaves that reside on one SGX platform. This is called *local attestation*. It can be used to locally verify the integrity of another enclave and establish a secure channel between them.

Regarding our scenario, the main advantage of using SGX technology over TPMs is that the attestation protocol can establish a shared secret between the data provider and an isolated enclave, which cannot be influenced or observed even by malicious administrators. On the other hand, using SGX requires an expansion of the trust model. Since so far only the Intel Attestation Service can verify the quotes generated by enclaves, it has to be fully trusted. However, Intel recently announced the upcoming support of third-party remote attestation infrastructures [11]. In the remainder of this section we describe how to use SGX technology for securing usage control infrastructures in our data sharing scenario.

4.1 SGX-Based Data Space Connectors

In order to benefit from the advantages of SGX technology in a collaborative data space, security critical modules have to be encapsulated in enclaves. This includes the usage control components as well as any software that is acquiring and processing sensitive information. On the data consumer's side, a connector enclave conducts remote attestations with the data providers, before collecting the requested data and associated usage control policies. The policies are forwarded to a dedicated PDP enclave, which determines usage control decisions by evaluating them. As before, the data processing applications contain PEPs that generate events for any attempted data usage and subsequently enforce the PDP's decisions. However, now the applications are realized as SGX enclaves and are locally attested by the connector enclave before they receive any sensitive data. The necessary communication between the data consumer's enclaves and the data provider is shown in Fig. 2. Immediately after launch the connector enclave verifies the integrity of the PDP enclave by locally attesting to its measurement value (called MRENCLAVE). Only if this local attestation is successful, the connector enclave executes the remote attestation protocol and establishes a shared secret with the data provider. If the remote attestation has been completed successfully as well, the data provider transmits the requested data along with usage control policies to the connector enclave. The connector enclave then acts as a trusted intermediary and shares the received data with each eligible application according to the specified usage rules. More concretely, the connector enclave first locally attests the active data processing applications, thereby verifying that the applications are legitimate and contain PEPs that enforce usage control decisions. Since the connector enclave does not know beforehand which data processing applications will be requesting data, a trusted third party (TTP) provides the expected enclave measurements. A secure channel to the TTP can be established by adding the TTP's public key to the code of the connector enclave. The integrity of this key is implicitly verified by the

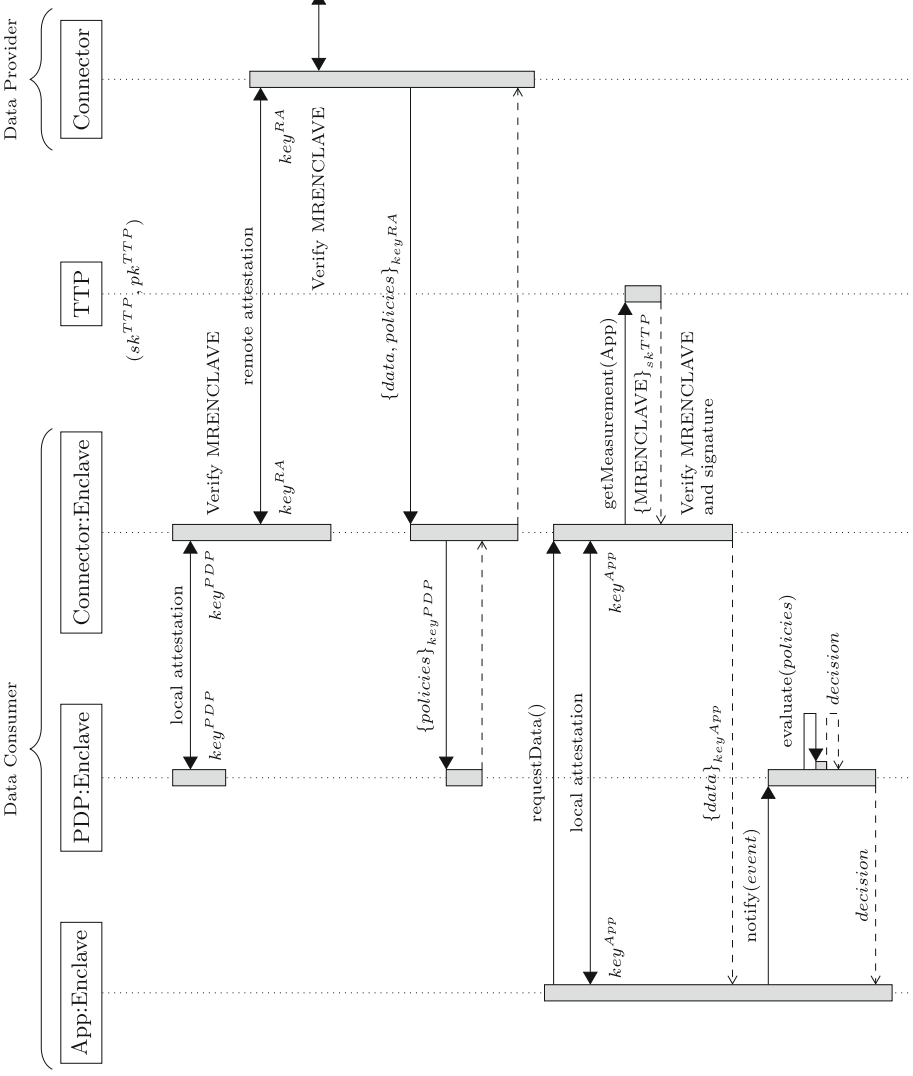

Fig. 2. A data space connector architecture based on SGX technology. Attestation messages are not shown.

data provider during the remote attestation. This process of locally attesting application enclaves avoids conducting separate remote attestations for every data processing application. If the local attestation of the application is successful, the connector releases the data to the application enclave. Afterwards the application's PEP contacts the PDP enclave for each attempted data access and enforces the resulting decision. For the sake of simplicity, Fig. 2 does not show the messages sent during the attestation procedures. In general, both the local and the remote attestation protocol establish a shared secret, which is then used to derive a symmetric session key for the attested connection. Further details of the attestation process are given in [1].

The security analysis of the SGX-backed connector solution is based on the same attacker model as before. We view the attacker as a malicious administrator of the data consuming connector system, who tries to intercept information and use it outside the usage restrictions imposed by the original data provider. Unlike with the previous TPM-based approach to data sovereignty, this solution shields shared data inside protected SGX enclaves from outside influence at all times. As a result, even malicious administrators cannot directly access or manipulate critical data and the in-memory tampering attack presented in the last section is not possible anymore. Another attack vector is the manipulation of enclave code before launching it on the data consumer's system. An attacker could try to tamper with the implementation of usage control components or modify the connector enclave in order to leak received data. However, tampering with enclave code is prevented by properly attesting the relevant enclaves before any data is released. Also, stealing sensitive data by duplicating attested connections is not possible anymore, because the attacker cannot successfully execute the remote attestation protocol with the data provider. Neither can the attacker pose as a data processing application and request data from the connector enclave, because the application is locally attested as well. By using SGX enclaves instead of TPMs to handle attestation, secure communication and data processing, the previously identified attack vectors have been resolved.

The proposed SGX-based architecture enables data providers to securely share information with collaborating data consumers. In addition to that, the architecture depicted in Fig. 2 also includes a usage control system that enforces usage restrictions on the shared data even after they have been transmitted. This makes it possible to realize flexible use cases where data from various sources have to be merged while simultaneously preserving data sovereignty. The architecture allows data providers to verify the trustworthiness of remote systems, and even malicious administrators cannot illegitimately access sensitive data. However, there are shortcomings with regard to the complexity of the data processing chains that can be constructed. Most importantly, the proposed architecture only supports internal usage control enforcement. This means that policy enforcement points can only be implemented as part of enclave applications. As a result the deployed usage rules can only control the usage of data that are processed by an application inside an enclave. Outside trusted enclaves the usage restrictions are no longer enforceable, which is why shared data must never leave the enclaves.

Since SGX-enabled processors only provide limited resources for enclaves, data processing applications cannot be very comprehensive. Typically only 128 MB encrypted memory (Enclave Page Cache, EPC) is available, which limits the size of applications that can run efficiently as enclaves. Furthermore, isolating the enclaves from the rest of the system – especially the operating system – has a considerable impact on the implementation of trusted applications. SGX enclaves have to link a specially modified system library, which re-implements numerous system operations that cannot be regularly executed by enclaves due to their independence from the operating system. This includes accessing memory and files, as well as inter process communication. Since applications running inside an enclave in general cannot depend on standard libraries, implementing complex data processing applications for the proposed architecture can be cumbersome or even impossible.

4.2 Joint TPM/SGX Architecture

Only by supporting the execution of data processing applications as normal, non-enclave processes, the disadvantages of an SGX-only solution can be overcome. However, data that are released outside enclaves into normal system processes still need to be protected. Due to the isolation from the rest of the system, usage control components that are realized as SGX enclaves cannot monitor normal system processes. In order to comprehensively enforce usage restrictions on non-enclave data processing applications, we need to support a powerful usage control system that can intercept data access on a kernel level (e.g. by hooking system calls). The integrity of such usage control components can be protected by including their code in a TPM-based chain of trust. However, as described previously, in our use case TPMs cannot sufficiently protect shared data against malicious administrators. In order to combine the flexibility of TPMs with the security of an SGX-based solution, a joint approach can be taken.

To achieve this we introduce a dedicated PEP that is responsible for intercepting data accesses and enforcing decisions across all data processing applications. In order to allow for complex data processing chains, this component does not run as an enclave and is instead implemented as a kernel module. As before, a PDP enclave receives usage control policies from the data provider and evaluates them for each intercepted event. The sequence of attestations that is necessary to securely share data using this architecture is illustrated in Fig. 3. During launch the TPM builds a chain of trust and measures both the PEP and all external data processing applications. Afterwards the connector enclave verifies the other enclaves (mainly the PDP, but also applications running as enclaves) by performing local attestations. Then the connector retrieves a TPM quote, queries the desired PCR values from the TTP and verifies all information. This step replaces the previously used TPM-based remote attestation. Instead of sending the quote to the data provider, the trusted connector enclave verifies the PCR values locally. Finally, the connector enclave performs the familiar remote attestation protocol with the data provider, thereby establishing a secure channel and announcing the integrity of the enclaves, the PEP and the outside

applications. After the data provider trusts the data consumer's system, he uses the established key to transmit data and usage control policies to the connector enclave. Since the external enforcement point has been attested, data may now leave the enclave into external data processing applications, if the policies allow it. The external PEP is then responsible for enforcing the usage restrictions even outside the enclaves.

The proposed system architecture combines the advantages of TPMs and SGX enclaves. While an SGX enclave is responsible for establishing the communication with data providers, the TPM safeguards the received data when they leave the enclaves for processing. The connector enclave is remotely verified by the data provider and receives the sensitive data along with their protection policies using the established secure channel. As described in the previous section, an attacker cannot intercept this communication or manipulate the transmitted information in memory, unlike when using only a TPM for attestation. Furthermore, attackers have no opportunity to impersonate a connector enclave and execute the remote attestation protocol with the data provider in order to steal sensitive data. Hence the previously described attack by duplicating attested connections is not possible with the joint architecture. On the other hand, by using an additional TPM to verify the integrity of the external system, data can be securely shared with processes outside the realm of SGX. Operating the enforcement point as a kernel module, which supervises the data usage of all running system processes, allows the execution of complex data processing applications as normal non-enclave processes. After receiving data and policies from the remote data provider, it is the responsibility of the connector enclave to issue the policy deployment and ensure that sensitive information is released outside the protected enclave only if it continues to be protected by the usage control system. For this, the connector enclave verifies the integrity of the external enforcement point by comparing the PCRs of the TPM to desired values. If an attacker tampers with the PEP in order to maliciously influence policy enforcement, the measurements will inevitably change and the PCR verification fails. In that case the connector enclave will not release any sensitive data to the outside world.

Despite the advantages of a combined approach, including a TPM brings back some of the problems that have been avoided in the SGX-only architecture. Most importantly, the TPM cannot prevent malicious administrators from accessing the unencrypted main memory of running data processing applications. However, this attack vector is only applicable for data that are in fact being processed by non-enclave applications. Given a policy scheme that is capable of describing data flows outside enclaves, the joint architecture allows original data owners to specify protection policies that prevent highly confidential information from leaving the trusted enclaves. In that case the data are safe from malicious administrators, but the complexity of supported data processing applications is limited. Furthermore, using a TPM always adds parts of the operating system to the chain of trust. This means that we have to assume the OS to be implemented correctly and free of security-critical bugs. Otherwise an attacker could

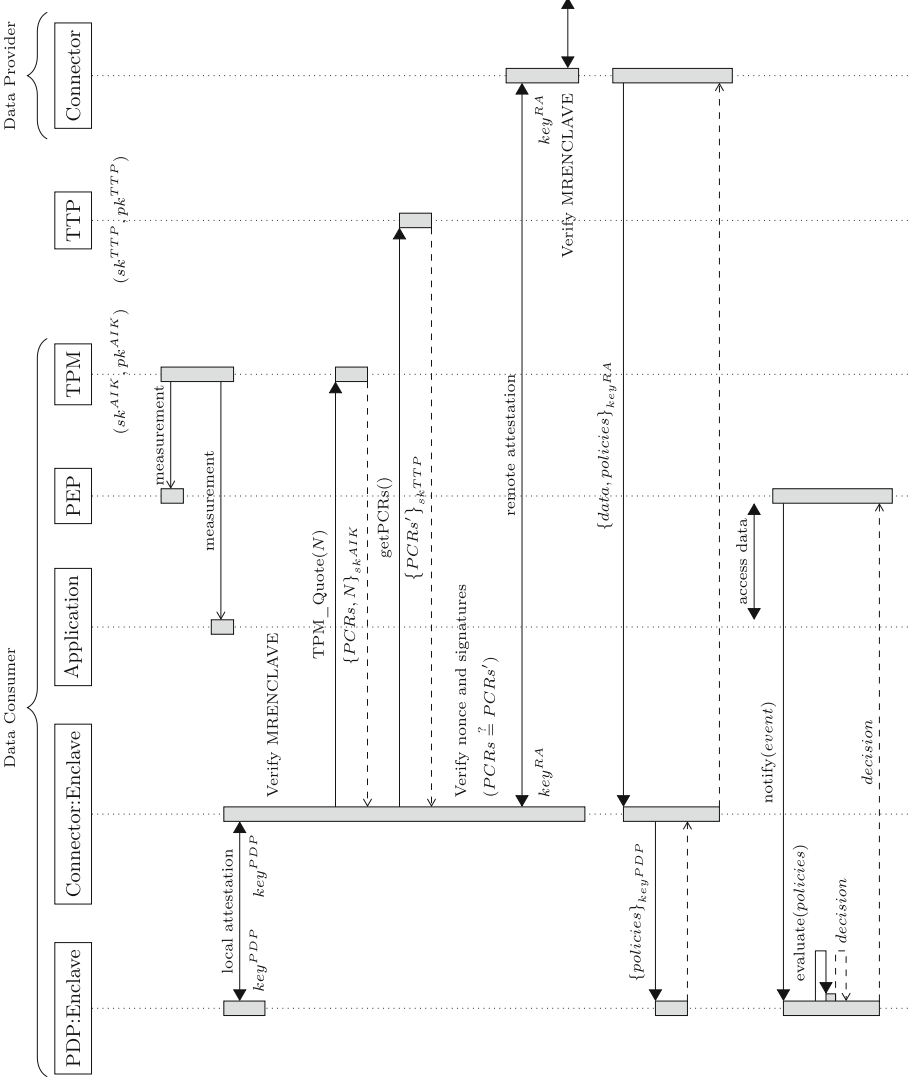

Fig. 3. A data space connector architecture based on TPMs and SGX technology. Attestation messages are not shown.

influence the enforcement point and bypass the protection mechanisms during runtime. While it is not necessary to trust the OS with the proposed SGX-only architecture, it is still a common requirement for many usage control systems. All in all, a system architecture that uses both SGX enclaves and a TPM resolves most of the attack vectors present with TPM-only solutions, while keeping the possibility of processing shared data in standard non-enclave applications.

5 Conclusion

In this work we evaluated the level of data sovereignty that can be reached by using TPMs to verify the integrity of data consumers. We have shown that especially in data sharing scenarios TPMs do not provide sufficient protection against malicious administrators from competing companies. Since TPM-based remote attestation can only identify the whole attested system as a trustworthy endpoint for the communication instead of a single process, sensitive data may be illegitimately intercepted. Furthermore, TPM-based attestation cannot adequately protect against in-memory tampering. In order to resolve these issues, we proposed an SGX-based connector architecture that enforces usage control policies even on malicious administrators. However, using SGX enclaves to process shared data considerably limits the scope of the data processing chains. Hence we proposed a joint connector architecture combining the advantages of both technologies. By including a TPM as well as SGX enclaves, this architecture supports powerful data processing applications while simultaneously preventing attacks that TPM-based systems suffer from. On the downside, using both technologies at once yields weaker security guarantees than the SGX-only solution.

Necessary future work includes the development of a policy scheme that makes it possible to distinguish data flows into enclaves from data flows into non-enclave applications. Due to the weaker security guarantees of data processing applications running outside enclaves, data providers need to be able to specify usage control policies that restrict the way their data may be processed. If multiple applications are involved in a data processing chain, this requirement needs to be enforceable across several enclaves as well. Furthermore, applying existing SGX development frameworks like SCONE[1] or Google's Asylo[2] to virtual data space architectures may ease the development of data processing applications in an SGX environment. However, the presented constraints of running data processing applications as SGX enclaves still remain problematic.

References

1. Costan, V., Devadas, S.: Intel SGX explained. IACR Cryptology Archive, p. 86 (2016)
2. D'Cunha, N.A.: Exploring the integration of memory management and trusted computing. Ph.D. thesis, Dartmouth College (2007)

[1] https://sconedocs.github.io/.
[2] https://asylo.dev/.

3. Delaune, S., Kremer, S., Ryan, M.D., Steel, G.: Formal analysis of protocols based on TPM state registers. In: 24th IEEE Computer Security Foundations Symposium (CSF 2011), pp. 66–80. IEEE (2011)
4. Götzfried, J., Eckert, M., Schinzel, S., Müller, T.: Cache attacks on Intel SGX. In: Proceedings of the 10th European Workshop on Systems Security, p. 2. ACM (2017)
5. Gruss, D., Lettner, J., Schuster, F., Ohrimenko, O., Haller, I., Costa, M.: Strong and efficient cache side-channel protection using hardware transactional memory. In: USENIX Security Symposium, pp. 217–233 (2017)
6. Gürgens, S., Rudolph, C., Scheuermann, D., Atts, M., Plaga, R.: Security evaluation of scenarios based on the TCG's TPM specification. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 438–453. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74835-9_29
7. Kelbert, F., Pretschner, A.: Data usage control enforcement in distributed systems. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy, pp. 71–82. ACM (2013)
8. Kocher, P., et al.: Spectre attacks: exploiting speculative execution. arXiv preprint arXiv:1801.01203 (2018)
9. Otto, B., Lohmann, S., Steinbuß, S., Teuscher, A.: IDS reference architecture model. Technical report, International Data Spaces Association (2018)
10. Park, J., Sandhu, R.: The ucon abc usage control model. ACM Trans. Inf. Syst. Secur. (TISSEC) **7**(1), 128–174 (2004)
11. Scarlata, V., Johnson, S., Beaney, J., Zmijewski, P.: Supporting third party attestation for Intel SGX with Intel data center attestation primitives (2018)
12. Sparks, E.R.: A security assessment of trusted platform modules computer science. Department of Computer Science, Dartmouth College, USA, Technical report, TR2007-597 (2007)
13. TCG: Architecture overview. Specification Revision 1 (2007)
14. Wagner, P.G., Birnstill, P., Beyerer, J.: Distributed usage control enforcement through trusted platform modules and SGX enclaves. In: Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies, pp. 85–91. ACM (2018)