Building Blocks for Identity Management and Protection for Smart Environments and Interactive Assistance Systems

Pascal Birnstill
Fraunhofer IOSB, Karlsruhe, Germany
pascal.birnstill@iosb.fraunhofer.de

Jürgen Beyerer
Fraunhofer IOSB, Karlsruhe, Germany
juergen.beyerer@iosb.fraunhofer.de

ABSTRACT

Interactive environments are more and more entering our daily life. Our homes are becoming increasingly *smart* and so do our working environments. Aiming to provide assistance that is not only suitable to the current situation, but as well for the involved individuals usually comes along with an increased scale of personal data being collected/requested and processed. While this may not be exceptionally critical as long as data does not leave one's smart home, circumstances change dramatically once smart home data is processed by cloud services, and, all the more, as soon as an interactive assistance system is operated by our employer who may have interest in exploiting the data beyond its original purpose, e. g. for secretly evaluating the work performance of his personnel. In this paper we discuss how a federated identity management could be augmented with distributed usage control and trusted computing technology so as to reliably arrange and enforce privacy-related requirements in externally operated interactive environments.

CCS CONCEPTS

• **Security and privacy → Information accountability and usage control**; **Privacy protections**; **Tamper-proof and tamper-resistant designs**;

KEYWORDS

Interactive Environments, Smart Assistance Systems, Identity Management, Distributed Usage Control, Trusted Reference Monitor, Trusted Platform Module, Remote Attestation

ACM Reference Format:
Pascal Birnstill and Jürgen Beyerer. 2018. Building Blocks for Identity Management and Protection for Smart Environments and Interactive Assistance Systems. In *PETRA '18: The 11th PErvasive Technologies Related to Assistive Environments Conference, June 26–29, 2018, Corfu, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3197768.3201563

1 INTRODUCTION

Interactive assistance systems are finding their way into daily life. Smart homes increase our comfort by controlling indoor climate, lighting or entertainment electronics according to our preferences. Future workplaces will interactively guide workers through complex manufacturing processes and assist surgeons during surgery. Aiming to be situation-aware and unobtrusive requires such assistance systems to monitor users, objects, and environmental conditions using a multitude of sensors. Personal data is collected and processed on a large scale, which raises privacy concerns and risks of abuse. This applies for workspaces, but also for smart home solutions, which often rely on cloud services for data analysis. Thus, interactive assistance systems have to be designed with caution, keeping in mind the paradigm of Privacy by Design (PbD).

Like interactive assistance systems, data protection is user-centric. It demands that users maintain sovereignty over their personal data, i. e. users must be able to determine which data is collected and used for which purposes by a digital assistance system. Assistance systems must also be transparent about whether and how long user data is stored, and about optional personal data, i. e. data that is not mandatory for the system to provide its service but may, for example, increase its awareness of situations.

Consider a system that guides a worker through a manufacturing process. The system has to capture the worker's activities in order to provide appropriate instructions for the detected workflow phase. This is usually achieved using video-based tracking of the worker's activities, the tools he uses and the objects on which he works, and matching these observations with model knowledge about the expected workflow, possible variations, and known complications. The assistance system may be able to increase its situation awareness and to provide even more suitable support, if it is allowed to track certain parameters indicating the worker's stress level. However, the worker is free to reject this extended data processing option or he may only agree to it for a short trial period.

Raw sensor data will typically not be stored (*live data*). However, live data is continuously analyzed to better adjust to users' characteristics over time so as to learn, e. g. that "user X is stressed once his heart rate exceeds 110 bpm". By this means live data becomes *profile data*, which may also include information such as "user X is experienced with manufacturing steps a, b, and e" or "user X is left-handed". Profile data may be directly provided by the user himself (e.g. "left-handed"), and the risk of abuse varies from non-existent to extremely high if it could be used for illegitimate performance monitoring or drawing conclusions regarding the user's state of health.

2 PREREQUISITES FOR DIGITAL SOVEREIGNTY

Given these examples it becomes evident that we are facing at least the following data protection challenges when it comes to interactive assistance systems:

PETRA '18, June 26–29, 2018, Corfu, Greece

ACM ISBN 978-1-4503-6390-7/18/06…$15.00
https://doi.org/10.1145/3197768.3201563

(1) Users must be enabled to declare and to manage their consents to live data collection by different assistance systems.
(2) Users must be enabled to manage identity attributes (profile data) shared with different assistance systems.
(3) Users must be enabled to arrange with assistance systems what must and what must not happen to their live data and profile data in terms of processing and storing of personal data.
(4) The enforcement of such usage constraints within externally controlled assistance systems must be reliable.
    - Assistance systems must be somehow validated or certified with respect to reliably enforcing privacy-related constraints and to not allow unintended export/leakage of personal data.
    - It must be ensured that operators/attackers cannot tamper with assistance systems (e. g. to circumvent the enforcement of privacy-related constraints) without being recognized.

A design of a privacy-aware assistance system must therefore make personal data collection and processing transparent, and provide an interface for defining usage arrangements according to users' preferences. In other words, assistance systems must be integrated within an infrastructure for maintaining and storing users' arrangements concerning live and profile data sharing (1, 2). This identity management infrastructure must also allow for managing policies concerning the usage of profile data and live data after access respectively collection has been allowed to an assistance system. Policies of this kind are called *usage control* policies, which are enforced by usage control mechanisms that have to be integrated within the assistance system, e. g. "Heart rate related data must be deleted after 7 days." (3). Usage control models require a trusted reference monitor (TRM) that continuously enforces policies on foreign data (4). Reliable enforcement further requires that the client, i. e. the assistance system, including the TRM has not been tampered with and is in a trustworthy condition. How a TRM can be properly protected in a hostile environment is the most challenging and unresolved question among the requirements listed above. In order to be able to verify a trustworthy system state, we have to demand that such systems are approved either by a certification process or by means of software verification methods so as to obtain a verifiable "fingerprint" of a trustworthy state. The task of verifying such a fingerprint is called *remote attestation*. Given a defined trustworthy state and a trusted third party attesting the according fingerprint, we can employ remote attestation protocols based on unforgeable hardware trust anchors such as Trusted Platform Modules (TPMs) to validate a system's integrity. Only after this verification we can deploy usage control policies on an assistance system we want to use, rely on their enforcement, and thus rely on the system's compliance to our data protection arrangements when we provide personal data or agree to its collection and processing by the system.

In this paper we outline an identity management and data protection enforcement infrastructure, which is based on User Managed Access (UMA), Distributed Usage Control (DUC), and Trusted Computing (TC) mechanisms. For user interaction with the identity management infrastructure, we rely on a mobile device, e.g., the user's smartphone. In the following, we describe the proposed technologies and how these could be combined to enable digital sovereignty while using interactive assistance systems.

3 IDENTITY MANAGEMENT: USER-MANAGED ACCESS

User-Managed Access (UMA) [3] is an access management protocol standard based on OAuth [2]. The purpose of the protocol specifications is to enable a resource owner to control the authorization of data sharing and other protected-resource access made between online services on the owner's behalf or with the owner's authorization by an autonomous requesting party. This means that UMA can be employed to deal with the consent management and access control issues described before (1). Conceptually, it uses an authorization server (AS) for managing resources, i. e. identity attributes used by certain services. The servers that actually provide the identity attributes act as so-called resource servers (RS) according to the terminology of UMA.

From the perspective of an assistance system requesting data or consent in data collection, UMA provides the following functionality. An assistance system can be authenticated using the OAuth-based OpenID Connect protocol. It can request data or consent to data collection from the UMA authorization server using an OpenID Connect [8] token for proving its identity. The AS will then look up the according profile. At this step, we have to distinguish at least the following cases:

(a) No profile exists
(b) A profile exists, but does not contain the resource requested by the assistance system
(c) A profile exists, but does not contain a policy matching the request
(d) A profile exists and does contain a policy matching the request

In case (a) the user of the assistance system is asked to create a profile for this newly approached system. He could do this immediately by accessing the identity management frontend via his smartphone. In order to facilitate this, the assistance system should submit a complete list of its consent and data processing requirements. The cases (b) and (c) indicate that the assistance system is advertising a new service, which is not covered by the profile so far. The user is asked to extend the profile for the given assistance system accordingly. Case (d) indicates that the UMA authorization token for accessing the requested resource is to be renewed. Otherwise the assistance system could have directly requested the resource from the UMA RS. If the user's policies in the according profile have changed since the assistance system has previously been authorized to use the given resource, new usage control policies may possibly have to be deployed at the assistance system, too.

4 DISTRIBUTED ENFORCEMENT OF PRIVACY-RELATED CONSTRAINTS

As explained before, usage control models have been introduced to extend the protection scope of data beyond the decision about whether to grant access to data or not. In other words, usage control

(UC) generalizes access control to the time after the initial access to data has been granted, which means that the usage of protected data is continuously monitored by a trusted reference monitor (TRM) given conditions specified in policies. Usage control models have been introduced by Park and Sandhu [5] as well as Pretschner and Hilty [6]. Requirements to be enforced include rights and duties, e. g. "data may not be forwarded", "data must be logged and deleted after 7 days", etc. Usage control policies are typically specified via events. Depending on a condition formulated in a policy, an event is *allowed*, *modified*, or *inhibited*. In distributed settings, e. g. forwarding a data item with an attached policy to another system, UC requirements can be enforced on the receiver's machine as well, which requires that usage control enforcement mechanisms are deployed at the receiving system. Reliable usage control enforcement in untrusted environments requires that UC mechanisms are based on a trusted reference monitor (TRM) as discussed in the following.

5 ESTABLISHING TRUST IN AN UNTRUSTED ENVIRONMENT

Since implementing a reference monitor in an untrusted environment is an important task in usage control systems, some previous research has been conducted. Implementing a reference monitor in a possibly hostile environment requires establishing a Trusted Computing Base (TCB) on the target system. This can be achieved by a Trusted Platform Module (TPM).

A TPM is a dedicated hardware chip that extends a computer with basic security related features [1]. The TPM holds several cryptographic keys that can be used to encrypt data, identify the computer system and attest to its current configuration. Furthermore the TPM contains volatile platform configuration registers (PCRs), which save a summary of the current hardware and software configuration as an unforgeable hash (called *measurement*). During the boot process, the TPM expects each boot stage to hash the software at the next stage and to extend the PCRs with this measurement. As a result, the PCR values reflect all the software measurements up to that point and hence attest to a certain set of software that is running on the system. Based on the PCR values, a TPM can encrypt data by *sealing* it to the current TPM state. Sealed data can only be decrypted by the same TPM and only in the same configuration state, i.e. the PCR values must still be the same as for the initial sealing operation. This ensures that sealed data can only be read in plain text if the system is in a trusted state. Furthermore, a third party can remotely verify that the target system is in a certain state by attesting to certain PCR values. This process is called *remote attestation*. TPMs are widely deployed, mainly because the TPM design relies on a dedicated tamper-resistant chip and does not require any hardware modifications. However, the TPM design provides only one isolation container, which covers all the software running on the computer, including the entire operating system and kernel modules. Since on most systems the operating system and kernel modules are regularly updated, acceptable measurement hashes (attesting to a trustworthy system state) also change frequently. This makes it tedious to apply TPM based software attestation in security systems. Nevertheless the application of a correspondent trusted software stack and the protocols that build on it has been extensively researched [1, 4].

Sandhu and Zhang [9] introduced the notion of a trusted reference monitor (TRM) inside the client-side operating system. The TRM is a reference monitor that operates in an untrusted environment, but is protected from external modification by a TPM. Implemented as a kernel module, the TRM is part of the measurement chain during the boot process. Hence its measurement is included in the PCRs, which prevents attackers from tampering with the reference monitor implementation. Before transmitting any data or policies, the data provider uses the remote attestation protocol to verify the PCR values of the client system. Only if the remote system is in a trustworthy state (i.e. the TRM is unmodified and running), information is transmitted.

Based on the work of Sandhu and Zhang, Sevinç et al. [10] developed a protocol that relies on a TPM to remotely verify the integrity of the client software stack. In this protocol, secrets are only transmitted to the client if the attestation is successful and the remote system can show the correct PCR values. Furthermore, the server binds the secret data to a key that is sealed to the required PCRs. That way the transmitted data can only be unsealed and used as long as the client system is in a trustworthy state. However, by relying only on TPMs, Sevinç's protocol has several drawbacks that have not yet been addressed.

Untrusted processes. The TPM cannot distinguish between trusted and untrusted processes. Even in trusted system states (i.e. the TRM is running and has not been tampered with) there will be untrusted user processes active in the system. In that case only trusted processes, such as the TRM itself, should be able to unseal the data. If the sealed data is intercepted during transmission, or is in any way available later on, any user process can request the TPM to unseal the data if only the PCRs still have the correct values. This bypasses TRM control on a software level.

Operating system. In order to distinguish trusted from untrusted processes, the TRM design may include operating system based protection mechanisms, such as access rights on files and directories. There are techniques to include executable content and security extended file attributes into the TPM measurement chain, such as the integrity measurement system and the extended verification module for Linux [7]. However, since in this case the user of the client system is an attacker, TPM based mechanisms are not sufficient to protect the sealed data that way. The user can mount the hard drive and access the sealed data in a secondary operating system. Even though the sealed data cannot be unsealed in this untrusted system state, the user can still make a copy of the encrypted data, without changing the original file meta data. When booting the unmodified operating system, the PCRs are filled with the correct values and the user can unseal the copied data.

Physical attacks. TPM based systems are vulnerable to attackers who have physical access to the machine. Because of this, the protocol does not protect against hardware based attacks. The user can still intercept plain text data directly at the board, for example at the communication bus between the processor and the TPM.

To conclude, the proposed solutions for a secure TRM implementation rely on establishing trust using a TPM, but are not sufficient to properly protect the transmitted data. This is mainly a result of the attacker model, which includes valid users of the client system

itself, who can use the TPM, launch untrusted system processes and have physical access to the hardware.

6 INTEGRATION OF SECURITY MECHANISMS

In figure 1 we outline how the security mechanisms and protocols, namely user-managed access, distributed usage control and remote attestation could be integrated into the communication sequence when a user approaches an interactive assistance system.

In the first step, the assistance system will announce its services to the user. The user receives this notification on his smartphone, where he can use his identity management app to create or maintain the user profile concerning the given assistance system. He can set permissions concerning live data acquisition to be performed by the assistance system and also concerning resources he wants to provide to the systems by granting access to an external resource server (UMA). In case some service requires an additional identity attribute, which is not yet stored on any attached resource server, the user can add this attribute via the identity management app and directly dispatch it to an existing resource server. Any permission or newly added attribute is automatically synchronized with the resource server responsible for the respective resource. In case the user wants to specify usage restrictions concerning his data, he also needs to choose according usage control policies, i. e. he wants his heart rate related data to be deleted after a trial period of 7 days (DUC). Finally, a reply containing the user's service selection is sent to the assistance system.

In the second step, the assistance system demands authorization to access the required resource(s) via the identity management server. However, before the identity management server will grant access to any resources, it demands remote attestation of the assistance system's integrity state. It asks the assistance system for a fingerprint of its integrity state, a so-called *quote*. At the same time it queries a quote of the last state known as trustworty from the remote attestation server, i. e. a trusted third party, which maintains the quotes capturing the states of somehow verified or certified systems. If the quotes are authentic and matching, we proceed with the deployment of usage control policies if such exist for the respective resources. Otherwise the process is aborted.

In the third step usage control policies are deployed at the assistance system in case the user specified such policies in the profile for the requested attributes. In other words, this step is only performed if the profile for the assistance system contains usage control policies. But in case it is performed, policy deployment must be successful. Otherwise the process is aborted.

If all steps have been passed through successfully, the identity management server will hand out an *authorization ticket* to the assistance system. Given this ticket, the resource server will finally grant access to the requested resources/identity attributes. While the data is processed, the TRM will monitor the enforcement of the according usage control policies within the assistance system.

7 CONCLUSION

In this paper we have shown how an identity management for interactive environments and the IoT can be built upon existing technologies and research results. User-managed access can be used to maintain consent decisions concerning data acquisition as well as permissions to access profile data. Usage control technology is required for enforcement of obligations such as deletion deadlines within an assistance system or some other data consuming service. Finally, we can only trust in the enforcement of usage control policies given that we make sure that the mechanisms have not been tampered with and that the according system is in a trustworthy state. We explained that this can be done based on hardware trust anchors, but not yet with satisfying guarantees. Future work is needed to explore whether TPMs can be combined with technologies such as Intel SGX so as to solve the open issues with TPM-based remote attestation.

8 ACKNOWLEDGEMENTS

This work was partially funded by the KASTEL project by the Federal Ministry of Education and Research, BMBF 16KIS0521.

REFERENCES

[1] Trusted Computing Group. [n. d.]. TCG architecture overview. (TCG Specification). ([n. d.]).
[2] Dick Hardt. 2012. The OAuth 2.0 authorization framework. (2012).
[3] Eve Maler. 2010. Controlling Data Usage with User-Managed Access (UMA). In *W3C Privacy and Data Usage Control Workshop, Cambridge*.
[4] John Marchesini, Sean W Smith, Omen Wild, Josh Stabiner, and Alex Barsamian. 2004. Open-source applications of TCPA hardware. In *Computer Security Applications Conference, 2004. 20th Annual*. IEEE, 294–303.
[5] Jaehong Park and Ravi Sandhu. 2004. The UCON ABC usage control model. *ACM Transactions on Information and System Security (TISSEC)* 7, 1 (2004), 128–174.
[6] Alexander Pretschner, Manuel Hilty, and David Basin. 2006. Distributed usage control. *Commun. ACM* 49, 9 (2006), 39–44.
[7] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leendert Van Doorn. 2004. Design and Implementation of a TCG-based Integrity Measurement Architecture.. In *USENIX Security Symposium*, Vol. 13. 223–238.
[8] Nat Sakimura, John Bradley, Mike Jones, Breno de Medeiros, and Chuck Mortimore. 2014. OpenID Connect Core 1.0 incorporating errata set 1. *The OpenID Foundation, specification* (2014).
[9] Ravi Sandhu and Xinwen Zhang. 2005. Peer-to-peer access control architecture using trusted computing technology. In *Proceedings of the tenth ACM symposium on Access control models and technologies*. ACM, 147–158.
[10] Paul Sevinç, Mario Strasser, and David Basin. 2007. Securing the distribution and storage of secrets with trusted platform modules. *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems* (2007), 53–66.

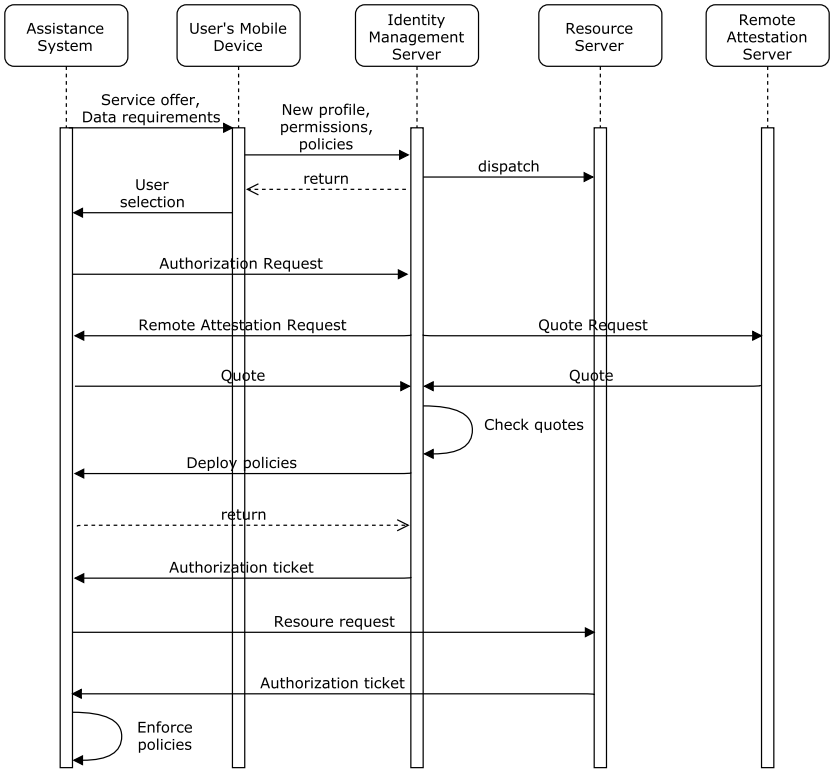

Figure 1: Integration of the described security mechanisms into the communication sequence when approaching an interactive assistance system