Facing the Blockchain Endpoint Vulnerability, an SGX-based Solution for Secure eHealth Auditing

Luigi Coppolino[1],
Salvatore D'Antonio[1], Giovanni Mazzeo[1], Luigi Romano[1], and Paolo Campegiani[2]

[1] University of Naples 'Parthenope', Naples, IT.
{luigi.coppolino,salvatore.dantonio, giovanni.mazzeo}@uniparthenope.it
[2] Bit4id S.r.l., via Diocleziano 107, 80125 Naples, IT.
pca@bit4id.it

Abstract

According to McAfee Labs, even in 2019, the eHealth sector is confirmed as one of the most critical in terms of cybersecurity incidents. It is estimated that more than 176 million patient records were target of attacks between 2009 and 2017, and with a single attack, in 2018, more than 1.4 million patient records were affected at UnityPoint Health. To cope with such a dramatic situation, one of the main strategic priority in the eHealth field is represented by the adoption of *Blockchain*. Specifically, according to a Deloittes survey, 55% of healthcare executives believe that blockchain technology will disrupt the healthcare industry. Unfortunately, while blockchain provides a valuable tool for enhancing the security of health applications and related data, it cannot be assumed as a panacea for data security. As an example, the so-called *Endpoint Vulnerability* issue is a well-known problem of *Blockchain*-based solutions: in such a case the attacker successful in gaining control of the end-point can tamper data off-chain during its generation and/or before it is sent to the chain. In this paper, we face such an issue by shielding the endpoint through the Intel Software Guard eXtension (SGX) technology. We demonstrate our solution for an auditing software belonging to the European eHealth management system (namely OpenNCP). We also discuss how our solution can be generalized to any other *Blockchain*-based solution. Finally, an experimental evaluation has been conducted to prove the actual feasibility of the proposed solution under the requirements of the real eHealth system.

1 Introduction

The *Blockchain* technology is becoming pervasive in today's societies. It is not just a buzzword, it is the hottest and fastest growing technological market in the IT sector. Statistical studies said that there are approximately 44% of companies from different fields (e.g., banks, agriculture, governments, accountants) that adopted a blockchain [6]. Traditional centralized services become powered by distributed systems based on the *Distributed Ledger Technology* (DLT) of the *Blockchain*. Distributed ledgers leverage synchronized databases, which provide an history of information visible to anyone within the network, rather than having a central administrator like a traditional database. Examples of blockchain-based services range from the most famous *Bitcoin* and *Smart Contracts* up to less notorious *Auditing* systems. In the healthcare context, there is a growing interest in *Blockchain*. In fact, the content recorded in the blockchain cannot be tampered, thus it can be used to record sensitive information in the eHealth management system. One of the most significant use case of blockchain in eHealth is mainly for auditing purposes, whose main potential objective is to create a secure, unforgeable registry for all log files regarding eHealth activities occurred, e.g., in a hospital management system. The integrity of log records is a fundamental element for eHealth security. For instance, once suspected or actual attacks occur, log data becomes important for identifying or isolating the malware. Thus, without solid audit logs, an attack may go unnoticed indefinitely and the particular damages done may be irreversible. The

intrinsic distributed property of *Blockchains* ensures to health-related data more confidence in terms of integrity. A successful attack should be able to modify the data retained by the *Blockchain* in all the nodes belonging to the network. In spite of this, there are still open issues affecting the Blockchain such as the *Endpoint Vulnerability* where data off-chain is at risk before being sent to the chain. An attacker could tamper an audit trail in the endpoint during its acquisition and before its encryption. The ultimate goal of this paper is to protect patient's privacy and ensure integrity of patients' data, while keeping track of critical operations in order to provide secure traceability support. We propose a solution for a *Blockchain*-based logging system that leverages *hardware-assisted trusted computing* to address the *Endpoint Vulnerability*. Sensitive functionalities carried out before sending the health-related logs to the *Blockchain* are executed within the Trusted Execution Environment (TEE) of Intel *Software Guard eXtension*, i.e., the widely-accepted technology of trusted computing released by Intel. Our work demonstrates through a security analysis and a performance evaluation that the proposed approach results usable in real contexts. Using requirements coming from a real healthcare management system, we demonstrate that the overhead introduced in the system is acceptable and does not alter the compliance with time-related constraints.
The proposed approach has been validated in the context of the KONFIDO project[1], whose main goal is to enhance security of the European eHealth system for cross-border data exchange, better known as *OpenNCP* [9]. This is a real healthcare management system, currently adopted by 11 EU member states to enable Patient Summary and ePrescription exchange accross countries. In the European Union, over 1.4 million people are working in a country different than those of residence [7], so they are natural possibile beneficiaries from this system, like also European citizens visiting a EU country. OpenNCP uses an ATNA-based auditing system, which is vulnerable to integrity attacks. For this reason, in KONFIDO, a blockchain-based service has been used to create a tamper-proof logging solution, as already explained by Castaldo et *al.* [4]. The only missing piece for closing the security loop is the protection from attacks targeting the Blockchain endpoint, which lies in the *National Contact Point* (NCP) of OpenNCP.

The remainder of this work is organized as follows. Section 2 overviews other research papers that used SGX with blockchain in other contexts. Then, Section 3 presents the two core technologies we adopted. Afterwards, Section 4 describes the OpenNCP eHealth system and provides pillars on the blockchain-based auditing service. Section 5 defines the problem and the challenges we face in this paper. The design and implementation details will be reported in Sections 6-7. Finally, Section 9 concludes the document.

2 Related Work

Other research works leveraged the security features of TEEs, in general, and of Intel SGX, in particular, for hardening blockchain-based services such as Bitcoin and Smart Contracts. Different security aspects of the blockchain have been covered by these works. To the best of our knowledge, this is the first research that faced the *Endpoint Vulnerability* and that propose the marriage of SGX and *Blockchain* in the eHealth sector. The security coming from their combination is extremely important in such a context where privacy and integrity requirements are particularly stringent.
Lind et *al.* [16] propose *TEEChain*, a payment solution that performs off-chain transactions asynchronously with respect to the underlying blockchain. The proposed idea wants to address the security of those payment networks that realize off-chain transactions instead of writing to the blockchain for each transaction in order to reduce the overhead. Authors propose an approach to prevent parties from misbehaving by hardening the transactions in SGX.
Yuan et *al.* [26] presents *ShadowEth*, a system where TEE and blockchain are combined to enable private smart contract based on public blockchains. The idea is to create a public smart contract named

[1] `www.konfido-project.eu`

"bounty contract" which performs the process of deployment and verification and stores the metadata of private contract. Additionally, they proposed an off-chain distributed storage named TEE-DS to store binary and states of private contracts.
Hardjono et *al.* [11] only explored how TEEs can be useful to harden individual nodes and systems in the blockchain infrastructure, and be the basis for secure group-oriented computations. Authors propose possible case studies such as the adoption of TEE to enable secure gateways for trust establishment across distinct blockchain systems.

3 Technologies Background

In the following, we present the core technologies used in this paper to build a trustworthy eHealth auditing system.

3.1 Blockchain

Blockchain is still an innovative technology which has yet to be properly sistematized. The current definition from the ISO/TC 307 on "Blockchain and Distributed Ledger Technologies" [1] provides for a tentative definition based on the concept of *ledger*. From ISO/TC307, a *distributed ledger* is a ledger shared across a set of nodes and synchronized between them using a consensus mechanism. Built on these definitions, a *blockchain* is a distributed ledger structured in blocks, organized in an append-only chain with cryptographic links, designed to be tamper-resistant, to create a definitive, final, and immutable ledger. These distinctive properties come from the clever combination of cryptographic techniques and incentive mechanisms for the different kind of players to keep the blockchain running and actively contribute towards its accessibility and availability, providing for an innovative solutions for the Byzantine Fault Tolerance problem [15] where different and isolated parties should reach consensus on a private value of information. The blockchain creates a chain of blocks ordered in a network of non-trusted peers. Every block points to the previous one and contains data, its own hash, and the hash of the previous block. A block stores encrypted details about the parties whose interaction resulted in the data stored in the block. BFT has been a research area for decades, when in 2009 Satoshi Nakamoto (a pseudonym for probably a group of people) came out with the original Bitcoin paper [20] where he proposed a crytpographic way to reach consensus between different parties with a so called Proof-of-Work (PoW) algorithm. As blockchains and distributed ledgers are being adopted in new areas, different and more specialized systems are put in place, including permissioned systems (when only authorized players could add transactions) and private ones (when authorization is required to read the transactions). An initial taxonomy is provided in [25].

3.2 Hardware-assisted TEE and Intel SGX

A *Hardware-assisted Trusted Execution Environment* (HTEE) has started to be widely adopted [5][8][3] —including in eHealth sector— to ensure that the confidentiality and integrity of both code and data are protected by means of hardware mechanisms. Unlike software-based solutions [14][18], a HTEE is able to guarantee security against attackers who have full control over the system, e.g., a malevolent cloud service provider exploiting its privileged position. Above all, a HTEE guarantees security features [17] such as *i*) *Isolation* of code and data residing inside against unauthorized access and modification; *ii*) *Attestation* of OS and/or application software; *iii*) *Dynamic Root of Trust* that builds trust chains. Over the years many HTEE implementations were released such as Intel SGX, AMD SEV, ARM TrustZone [17].
In this work, the focus is on the Software Guard eXtension (SGX) of Intel's CPUs [19], which allows the creation of protected memory regions, namely *enclaves*. In such HTEEs, code and data are protected from disclosure or modification thanks to address regions whose content is protected – via encryption

and hashing – from any software outside the enclave, included privileged ones (e.g., OS or Hypervisor). Only the enclave code can access any part of the address space, except those areas belonging to other enclaves. The boundary between enclave and non-enclave sections is governed by the processor who blocks any access attempt from unauthorized processes. An interface – defined in a domain-specific C language – is declared by the programmer to establish entry points, i.e., calls to/from an enclave (namely ECALLS and OCALLS). A HTEE such as SGX can be formalized as follows [23]. We define the enclave program as the tuple:

$$e = \langle init_e, config_e \rangle$$

where $config_e$ includes the entrypoints $config_e.entrypoint$, the virtual address range $config_e.evrange$, the access permissions $config_e.acl$, and other application-specific configuration data $config_e.app$. While $init_e$ brings the initial state of SGX such as values of memory pages pointed by $config_e.evrange$ that contains code and data. We define the enclave state in some instant of time as $E_e(\sigma)$, which is a projection of the machine state. Finally, enclaves' inputs are specified as $I_e(\sigma)$, i.e., a partial map from virtual addresses (outside $config_e.evrange$) to machine words. Enclaves' outputs are defined as $O_e(\sigma)$, i.e., a partial map from virtual addresses (outside $config_e.evrange$) to words. The semantics of enclave execution is the set of execution traces, containing an execution trace for each input sequence, i.e., for each value of non-enclave memory.

$$[[e]] = \{\langle (I_e(\sigma_0), E_e(\sigma_0), O_e(\sigma_0)), ... \rangle | init_e(E_e(\sigma_0))\}$$

The determinism property of SGX programs entails that a specific sequence of inputs $\langle I_e(\sigma_0), I_e(\sigma_1), I_e(\sigma_n) \rangle$ uniquely identifies a particular execution trace of $[[e]]$ under that sequence of inputs. An important feature of SGX is the *remote attestation*, which enables the integrity verification of an enclave residing in external domains via Intel's third-party server. Let $\mu(e)_{\mathcal{D}}$ be the quote of a generic enclave e generated in some domains $\mathcal{D}$, we say that attestation is verified when:

$$Att([[e]])_{\mathcal{D}} \Leftrightarrow \mu(E_e(\sigma))_{\mathcal{I}} = \mu(E_e(\sigma))_{\mathcal{D}}$$

where $\mu(E_e(\sigma))_{\mathcal{I}}$ is the quote generated in Intel's *Attestation Service* of the enclave $[[e]]$ in a determined state $E_e(\sigma)$.

4 The Blockchain Auditing for the eHealth OpenNCP System

The current deficiencies in security logging and analysis allow attackers to hide their location, malicious software, and activities on victim machines. Even if the victims know that their systems have been compromised, without protected and complete logging records they are blind to the details of the attack and to subsequent actions taken by the attackers. Without solid audit logs, an attack may go unnoticed indefinitely and the particular damages done may be irreversible. Logging records could also be the only evidence of a successful attack. Moreover, many healthcare organizations keep audit records for compliance purposes. Because of poor or nonexistent log analysis processes and log integrity, attackers sometimes control victim machines for months or years without anyone in the target organization knowing, even though the evidence of the attack has been recorded in unexamined log files.
Such issues also affected the logging system of the European eHealth management system, namely OpenNCP [9]. This is an implementation of the NCPeH protocol, adopted by National Contact Points (NCPs) to securely exchange eHealth data between European Union Member States (EU MSs). The NCPeH manages the exchanges of Patient Summary (PS) and exchange of Prescription (eP) as requested by a physician or a pharmacist who are providing health treatments in a EU MS B, where a citizen, originated from an EU MS A, is seeking medical assistance. An OpenNCP node interacts with the national infrastructure of EU MS B (where the requests for eHealth data are originated) and with the OpenNCP of EU MS A, which is in charge of providing data by asking the national infrastructure

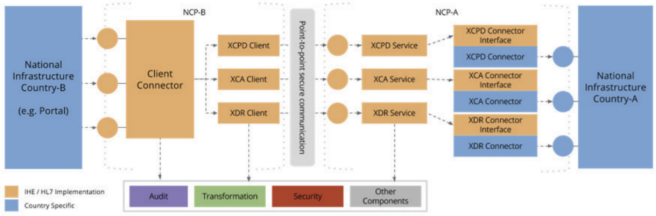

Figure 1: OpenNCP architecture

of EU MS B. An OpenNCP node plays both these two roles, according to the flow of requests. NCPeH comprises many IHE specific protocols, as shown in figure 1, from [10].
OpenNCP acts as an ATNA [12] client, to implement an Audit trail. The internal Audit Manager provides an interface for the Audit Trail Service, that keeps tracks of all of the security relevant events by logging them using the built-in OpenATNA server. In the context of the aforementioned KONFIDO project, the OpenATNA server has been replaced with the proprietary Hylos log server [2], which is IHE certified and has been extended to store a subset of all of the relevant logs inside a private blockchain system (Fig. 2). These logs, describing a transaction between two different EU MSs, are stored encrypted in a way that each one of the two MSs could decrypt them autonomously, while the actual content of the data are not available to the other EU MSs not participating in the transaction. As a result of that, each EU MS directly contributes towards the resilience and tamper-proof property of the blockchain, while privacy of the transactions is preserved. Should a contrast between two countries arise in the future regarding a specific transaction (as an example, Country B wants to show that data have been asked but not provided by Country A, resulting in a more complex health treatment for the citizen of Country A) each country (A and B) could access the data of its interest, being sure that they have not been tampered with (as the blockchain with the data is replicated by all the EU MSs). Details for the implementation are in [4].

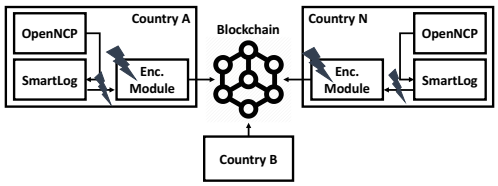

Figure 2: Components of the *Blockchain*-based Logging Service for OpenNCP

5 Problem Definition

The distributed properties typical of the blockchain provide more security guarantees to the sensitive health-related logs acquired by the system previously described, which creates tamper-proof audit trails of the OpenNCP activities. More precisely, the distributed consensus and storage of the blockchain ensure that information cannot be modified unless it is agreed across the entire DLT. In a nutshell, several nodes have the same information, thus an attacker should be able to get access and change all these distributed data in order to launch an attack to the data integrity. Even identities are usually held on the DLT. When a user has to be authenticated, there is no more a centralised party that checks and verifies data of individuals. Thanks to the DLT, the credential information are distributed. The possibility that an attacker manipulates identification data is highly reduced.

Even if a blockchain increases the resistance to considerable category of attacks, it does not make it immune against security vulnerabilities, which are still considered today as the most critical open issues of this technology [24]. It is important that these security risks are properly recognized and mitigated, especially when blockchain systems manage data subjected to strict privacy requirements such as in the eHealth sector. One relevant vulnerability of DLT is outside the blockchain itself. It is usually known as *Endpoint Vulnerability* [22]. Such an issue arises when new information of a blockchain-enabled service is inserted or withdrawn into/from the DLT. In such a situation, the endpoint represents the place where an individual or another software use to access the blockchain-based service. It is during the process of getting access to the blockchain that the data is most vulnerable. The *Endpoint Vulnerability* could lead to threatening the personal user security, stealing the private key, or introducing new malware. In the blockchain-based auditing system presented before, the audit trail is at risk in the NCP endpoint when it needs to be ciphered before being sent to the blockchain. Figure 2 depicts the specific weak points where the attacker could tamper the log. The integrity of the log may be affected. Something similar could happen even for other services. For example, it may occur in *Smart Contract* systems during the contract setting up process, and before this is deployed on the blockchain. Users' sensitive inputs are at risk off the blockchain. The attacker could inject fake data in the contract.
The goal of this work is to design a solution to the *Endpoint Vulnerability* by leveraging isolation and attestation features typical of the Intel SGX extension. The security-enhanced logging system should be able to:

(1) Securely acquire and generate an audit trail
(2) Shield the data ciphering procedure
(3) Protect the keys used for the encryption
(4) Harden the Endpoint-*Blockchain* communication

6 Solution Design

6.1 Dataflow Overview

Figure 3 shows the location of the SGX-enabled auditing system based on *Blockchain* in a typical healthcare dataflow such as a clinical document acquisition in OpenNCP. The prerequisite is the availability of the SGX hardware extension in each national gateway (i.e., the NCP node). When new events occur, e.g., a patient summary request or a document transformation, the auditing system has to securely store the activities in the *Blockchain*. The *SmartLog* and *Encryption* modules do not exist anymore. Instead, an SGX enclave responsible for the log generation is instantiated internally by OpenNCP. The audit record is securely built in the trusted execution environment with all the necessary information such as the event timestamp. In this regard, as explained below, our solution uses a trusted time source since a compromised timestamp could be harmful for the system. At this point, not even an attacker with *root* privileges can tamper the log during its generation. Then, the log must be sent to the *Blockchain*. Hence, the enclave encrypts and signs the audit record with a key stored in the SGX secure perimeter. From now on, logs content is confidential even off SGX and the integrity of the events as well as their order is ensured. The key management in terms of generation, distribution, and storage, is discussed further below. The cryptographic operations are shielded from any privileged insider thanks to the SGX hardware-enabled isolation features. Finally, in order to ensure the security of the communication channel, an SGX-secured TLS is established with the *Blockchain*. This means that the communication terminates directly in the enclave, which is also responsible of keeping the TLS private key. With our solution, we eliminate the risk of privileged attackers such as malicious insiders by enabling independent log integrity and time verification which cannot be backdated.

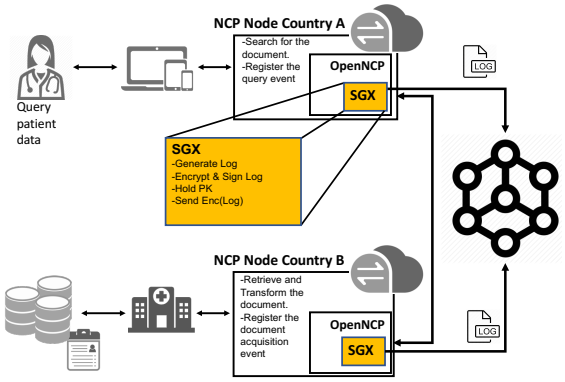

Figure 3: The customized SGX-enabled auditing system based on *Blockchain* in the eHealth dataflow

6.2 Timestamp Acquisition

For security reasons, most of the fields of a log record are directly generated in the SGX enclave, thus avoiding the risk of tampered entrypoints, e.g., malicious outputs returned to an OCALL (i.e., the function called from within the enclave). The only source of attack may be represented by the timestamp. In fact, services such as timer and monotonic counters are not supported inside the Intel SGX technology [13]. When a request for the timestamp is made, the enclave performs the untrusted OCALL. The attacker could manipulate the time value in order to generate a misalignement in the log system. For this reason, in our solution, the timestamp is provided by a separate TEE, and securely made available to Intel SGX enclaves. According to Intel [13], the Intel SGX software supports the implementation of such a secure service. It leverages the security capabilities of the Intel Converged Security and Management Engine (CSME). The CSME is an embedded engine running its dedicated firmware in the Platform Control Hub (PCH) on Intel platforms, and is separate from the CPU. Hence, the SGX enclave responsible for the OpenNCP log generation communicates over a secure channel with the CSME Protected Real-Time Clock (PRTC). The only issue is that the trusted time service provides coarse-grain timer values relative to a reference point. It does not provide a trusted wall clock time. The idea is that the trusted time service uses an epoch value to enable the enclave in detecting if there is discontinuity between different timer reads. A change of the timer source epoch value between two reads indicates that either the PRTC in the paired CSME has been reset due to events like battery replacement, or the PSE has paired with a different CSME due to unexpected event such as software attack.

6.3 Key Management

We implemented a dedicated approach for what regards the key management and distribution. The idea is to leverage the attestation features of SGX to avoid the use of a third-party key management server that could be potentially untrusted. At startup time, when the NCPs are setup, every enclave starts performing the remote attestation procedure to verify that all the other enclaves in the NCP nodes have not been tampered. Once completed this phase, keys are exchanged. Particularly, the way we do this is through the *Diffie-Hellman* scheme, which allows the exchange of keys over untrusted channels. At the end of this process, all the SGX enclaves will have a list of keys for all the involved countries. Keys are finally saved in a persistent storage after being ciphered.
It is important to notice that such an approach fits well for this particular case study where there is a limited amount of *Blockchain* endpoints involved. In a different situation, another possibility may be to

leverage a typical key management solution in which a third-party is responsible for the key generation and distribution.

7 Implementation Details

Designing an SGX-based solution requires a fundamental decision, i.e., *what should be placed inside and outside an enclave*. Or better, *which functions should be exposed to the outside world and to the enclave in the SGX interface*. This is particularly important since the decision affects both the security properties in terms of *Trusted Computing Base (TCB)* size, and the performance overhead. The methodology pursued for porting the C++ SGX enclaves in the Java-based OpenNCP is the *Ad-Hoc* approach that requires manual and dedicated partitioning of the Java software. We used a *Java Native Interface* (JNI) bridge to link the trusted and untrusted worlds. The Java code runs outside the enclave, and uses the JNI bridge to interact with trusted code components running within SGX secure enclaves. While this approach entails a non-trivial development activity on the software to be secured, it ensures a much smaller size of protected code and, at the same time, a reduced number of transitions between secure and insecure worlds.
Figure 4 shows a simplified call graph of the implemented solution in the extended OpenNCP system. When the user (e.g., a doctor) performs the query through the *WebPortal*, then the function *query()* is called in the endpoint node (PTA: the protocol terminator of the NCP A). The execution control will be directly moved into the SGX enclave via the init function *sgx_init()* that launches the key exchange procedure, in case keys have not been assigned yet. Afterwards, the query is carried out by the enclave and the event is registered in the blockchain by the *SGXLogger* object.

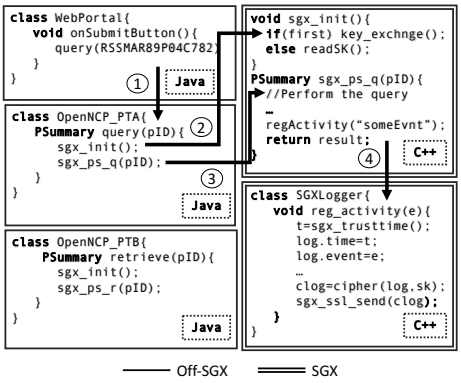

Figure 4: A simplified call graph of the SGX-enabled Logging System

8 Performance Evaluation

In this section, we present the evaluation activity conducted on the SGX-enabled auditing solution based on *Blockchain*, which has been realized using workloads typical of the OpenNCP eHealth system. We report the findings on transaction throughput and transaction latency measurements that we conducted using the *Bit4id* proprietary Blockchain *Hylos*. Our goal was to verify that the overhead introduced by the SGX component in terms of log generation and sending is acceptable and does not alter the compliance of the auditing system with OpenNCP requirements.
The experimental setup must include the OpenNCP endpoints and the *Blockchain* network. The former

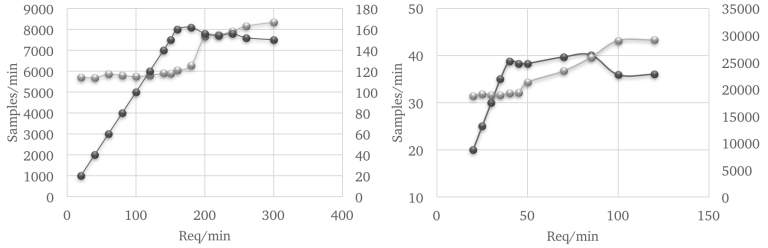

Figure 5: Capacity test results, with and without SGX

was simulated in a single node, which is SGX-enabled. This is reasonable since the transmission latency between the two NCP nodes is not of interest for the purpose of this evaluation. Such a node has the following specifications: an Intel Xeon E3-1270 v5 CPU with 4 cores at 3.6 GHz with SGX extension enabled, 8 hyper-threads (2 per core), and 8 MB cache. The host has 64 GB of memory and runs Ubuntu 16.04 LTS with Linux kernel version 4.2. The Intel SGX PSW and SDK v2.6 have been used for implementing and launching the SGX enclave. Regarding the *Blockchain*, the proprietary *Bit4id Hylos* based on Proof of Work (PoW) consensus algorithm and configured with 20 peers has been used. All peers ran on hardware machines belonging to the Italian Veneto eHealth system. Each machine had 8 vCPUs (4 cores at 3.6 GHz) and 16 GB RAM.
Our goal is to observe the impact given by SGX to the processing phases occurring in the endpoint before the data is sent to the *Blockchain*. To this end, we measure the *throughput* t of requests served per second, and the *service (or response) time* s that represents the interval of time needed to perform the log-related subroutines in the endpoint. The idea is to compare the performance of the native auditing system with the solution using SGX. The goal is to characterize the level of system performance given a specific workload that is typically the one expected for the immediate future. The workload in terms of events arrival rate depends on the number of requests typically made to the NCP endpoints. Such a number unfortunately is not available to the public, thus our approach is to start from EU statistics, similarly to [5], and estimate the possible workload scenarios. The worst case is given by the endpoint with the highest number of document requests in both directions. This means that we need to look at the country with high percentages of EU immigrants and large number of people accessing the health service. According to migration and migrant population statistics, Germany has 7.2% of immigrants. Furthermore, according to the 2018 health report of the German health ministry [21] the number of patients in a year is around 20M. Considering the peak in February it can be deduced that in this month there are about 1'214'000 patients, applying to these the percentage of foreigners originating from another EU country (i.e. 7.2%), 87408 request are obtained in one month; distributing them evenly per day and per hour the result is 130 requests per hour or 2.17 requests per minute. We derived the *Usable capacity* and the *Knee capacity*: the first represents the maximum throughput achievable without exceeding a pre-specified response-time limit, the latter is the point beyond which the response time increases rapidly but the gain in throughput is small. The measurements obtained here can be then used for the design of experiments. In Figure 5, it is possible to observe the graphs of throughput and response time as the request rates vary. In the native auditing system without SGX, the *Usable capacity* is $180 req/min$, while the *Knee capacity* is 160 req/min. In the other case, the *Usable capacity* is 85 req/min and the *Knee capacity* is 45 req/min. The comparison of such results shows that there is a non-negligible overhead in the auditing system. In spite of this, the solution ensures performance compliant with the considered worst case of system workload.

9 Conclusion

This paper proposed a promising approach to address the well-known *Endpoint Vulnerability* affecting the majority of *Blockchain*-based services. It described a solution based on the Intel SGX technology in which data is protected from tampering within the SGX trusted execution environment. Sensitive phases of data acquisition and encryption — occurring in the endpoint before sending to the *Blockchain* — are realized in the SGX enclaves. We implemented the proposed approach for a *Blockchain*-based auditing system used in the context of the European eHealth system, namely OpenNCP. The performance evaluation demonstrated that the SGX-enabled logging introduces an overhead, which is acceptable for the specific eHealth system requirements. It is important to notice that our approach is easily usable to any other *Blockchain*-based service where the *Endpoint Vulnerability* could pose at risk the integrity of data.

Acknowledgments

This project received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement No 727528 (KONFIDO).

References

[1] ISO/TC 307 Blockchain and distributed ledger technologies. `https://www.iso.org/committee/6266604.html`. Accessed: 2019-09-27.

[2] Bit4id. Hylos - Secure Log Server, 2019.

[3] F. Campanile, L. Coppolino, S. DAntonio, L. Lev, G. Mazzeo, L. Romano, L. Sgaglione, and F. Tessitore. Cloudifying critical applications: A use case from the power grid domain. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 363–370, 2017.

[4] Luigi Castaldo and Vincenzo Cinque. Blockchain-based logging for the cross-border exchange of ehealth data in europe. In Erol Gelenbe, Paolo Campegiani, Tadeusz Czachórski, Sokratis K. Katsikas, Ioannis Komnios, Luigi Romano, and Dimitrios Tzovaras, editors, *Security in Computer and Information Sciences*, pages 46–56, Cham, 2018. Springer International Publishing.

[5] L. Coppolino, S. DAntonio, G. Mazzeo, L. Romano, and L. Sgaglione. Exploiting new cpu extensions for secure exchange of ehealth data at the eu level. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 17–24, Sep. 2018.

[6] Deloitte. Global blockchain survey, 2019.

[7] A. Markowska M. Jones E. Fries-Tersch, T. Tugran, 2018.

[8] Christof Fetzer, Giovanni Mazzeo, John Oliver, Luigi Romano, and Martijn Verburg. Integrating reactive cloud applications in sereca. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, pages 39:1–39:8, New York, NY, USA, 2017. ACM.

[9] Marcelo Fonseca, Kostas Karkaletsis, Isabel A. Cruz, Alexander Berler, and Ilídio Castro Oliveira. OpenNCP: a novel framework to foster cross-border e-health services. *Studies in health technology and informatics*, 210:617–21, 2015.

[10] Marcelo Fonseca, Kostas Karkaletsis, Isabel A Cruz, Alexander Berler, and Ilídio Castro Oliveira. Openncp: a novel framework to foster cross-border e-health services. In *MIE*, pages 617–621, 2015.

[11] Thomas Hardjono and Ned M. Smith. Decentralized trusted computing base for blockchain infrastructure security. *CoRR*, abs/1905.04412, 2019.

[12] Integrating the Healthcare Enterprise IHE. Audit Trail and Node Authentication, 2019.

[13] Intel. Trusted time and monotonic counters with intel software guard extensions platform services, 2018.

[14] Ramya Jayaram Masti, Claudio Marforio, and Srdjan Capkun. An architecture for concurrent execution of secure environments in clouds. In *Proceedings of the 2013 ACM Workshop on Cloud Computing Security Workshop*, CCSW '13, pages 11–22, New York, NY, USA, 2013. ACM.

[15] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[16] Joshua Lind, Ittay Eyal, Florian Kelbert, Oded Naor, Peter R. Pietzuch, and Emin Gün Sirer. Teechain: Scalable blockchain payments using trusted execution environments. *ArXiv*, abs/1707.05454, 2017.

[17] P. Maene, J. Gotzfried, R. de Clercq, T. Muller, F. Freiling, and I. Verbauwhede. Hardware-based trusted computing architectures for isolation and attestation. *IEEE Transactions on Computers*, PP(99):1–1, 2017.

[18] Lorenzo Martignoni, Roberto Paleari, and Danilo Bruschi. Conqueror: Tamper-proof code execution on legacy systems. In *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA'10, pages 21–40, Berlin, Heidelberg, 2010. Springer-Verlag.

[19] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. Innovative Instructions and Software Model for Isolated Execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP, 2013.

[20] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[21] German Ministry of Health. Report on patients accessing health services, 2018.

[22] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles A. Kamhoua, Sachin Shetty, DaeHun Nyang, and Aziz Mohaisen. Exploring the attack surface of blockchain: A systematic overview. *CoRR*, abs/1904.03487, 2019.

[23] Pramod Subramanyan, Rohit Sinha, Ilia Lebedev, Srinivas Devadas, and Sanjit A. Seshia. A formal foundation for secure remote execution of enclaves. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 2435–2450, New York, NY, USA, 2017. ACM.

[24] Paul J. Taylor, Tooska Dargahi, Ali Dehghantanha, Reza M. Parizi, and Kim-Kwang Raymond Choo. A systematic literature review of blockchain cyber security. *Digital Communications and Networks*, 2019.

[25] UK Government Office for Science. Distributed Ledger Technology: beyond block chain, 2016.

[26] Rui Yuan, Yu-Bin Xia, Hai-Bo Chen, Bin-Yu Zang, and Jan Xie. Shadoweth: Private smart contract on public blockchain. *Journal of Computer Science and Technology*, 33(3):542–556, May 2018.