HistoTrust: Attestation of a Data History based on off-the-shelf Secure Hardware Components

Dylan Paulin, Christine Hennebert,
Thibault Franco-Rondisson, Romain Jayles, Thomas Loubier, Raphaollado

Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France
christine.hennebert@cea.fr

Abstract. Device- or user-centric system architectures allow everyone to manage their personal or confidential data. But how to provide the trust required between the stakeholders of a given ecosystem to work together, each preserving their interest and their business? HistoTrust introduces a solution to this problem. A system architecture separating the data belonging to each stakeholder and the cryptographic proofs (attestations) on their history is implemented. An Ethereum ledger is deployed to maintain the history of the attestations, thus guaranteeing their tamper-resistance, their timestamp and their order. The ledger allows these attestations to be shared between the stakeholders in order to create trust without revealing secret or critical data. In each IoT device, the root-of-trust secrets used to attest the data produced are protected at storage in a TPM ST33 and during execution within an ARM Cortex-A7 TrustZone. The designed solution aims to be resilient, robust to software attacks and to present a high level of protection against side-channel attacks and fault injections. Furthermore, the real-time constraints of an embedded industrial application are respected. The integration of the security measures does not impact the performance in use.

Keywords: Attestation · Secure Hardware · TPM · OP-TEE · Trust · Data History · Ledger · Embedded Industrial Application · IoT · Real-Time Performance

1 Introduction

Logs trace the activity of a device in the form of a data history of various kinds, such as its internal states, connection, communication activity or the data it produces. Their audit engages the accountability of the owner of the device as a legal entity. Within an ecosystem of stakeholders, each one is thus accountable to provide a trusted history of the data produced by its devices to an auditor in the event of a litigation. This confidential data is of great value for the business of the stakeholders.

Attestation schemes based on the use of a TPM offer standard solutions allowing the authentication of a platform by a remote device [1] [2], or even making the user anonymous [3]. But these schemes do not consider the real time data

emitted by the industrial applications embedded on the trusted IoT devices [4] [9]. The authors of [5] highlight this issue through the delicate question of the certification of sensor data, even by a trusted platform. The tension between privacy, which requires the protection of confidential data, and trust which requires guarantees between the stakeholders working in a given ecosystem is tangible.

The blockchain provides a technology that maintains by design a history of proofs or transactions [6]. Histotrust introduces a device-centric [10] solution based on Ethereum technology that conciliates the need for data security and privacy with the trust required between stakeholders. HistoTrust provides an architecture that ensures end-to-end security and privacy by design while satisfying the real-time data transmission needs of the embedded industrial application. The design of an enhanced wallet is outlined. It serves as root-of-trust for the data emitted by the IoT device.

The following section positions the work done in HistoTrust in relation to existing solutions. The use case, the threat model and requirements are then described in section 3. Section 4 outlines HistoTrust solution, its secure system architecture and the embedded implementation based on off-the-shelf secure hardware components for IoT devices. Section 5 is dedicated to the presentation of the results and discussions before the conclusion.

2 Related works

2.1 Secure data history with trusted hardware

The paper [7] shows the added value of blockchain technology to meet the specificities of a smart manufacturing use case. Compared to a centralized solution based on digital certificates and PKI, the Ethereum-based solution shows a more refined management of security and privacy at the expense of performance. In this paper, HistoTrust solution demonstrates that performance can also be maintained and met the needs of the use case when using a blockchain.

The authors of EmLog [4] present their framework as *"the first attempt at preserving off-the-shelf ARM development board hosting OP-TEE"*. EmLog implements a secure logging system from end-to-end between embedded constraint devices and a remote database. HistoTrust introduces an architecture design and an on-board implementation design using off-the-shelf secure hardware components, as OP-TEE and TPM 2.0 [11], that goes beyond EmLog solution and achieves the EmLog perpectives. Preserving forward security thanks to the one-way hash chain scheme introduced by Shneier and Kelsey [8], EmLog and SGX-Log [9] are not designed for multi-stakeholders contexts and may suffer of lost of data in case of power failure.

In the Logs system EngraveChain detailed in the paper [12], the data history is ciphered, then registered in an Hyperledger Fabric ledger. This implementation lacks agility because the blockchain is not designed to store large volumes of data, nor confidential data even encrypted. Moreover, the ciphering of recorded data in a ledger implies a complex key management.

The blockchain technology provides by design the tamper-resistance of the recorded transactions history forming the ledger. HistoTrust provides an attestation scheme securing the history of data issued from distributed devices. An Ethereum ledger maintains the history of cryptographic attestations of data produced by distributed devices owning per multiple stakeholders. The blockchain technology enables to share these cryptographic proofs between the involved stakeholders providing trust. In addition, the raw data are kept by their owner who ensures their persistence and confidentiality.

Based on an Ethereum blockchain, BlockPro [13] presents a decentralised architecture of IoT devices. The authenticity of the devices issuing data is achieved through a challenge to the IoT device submitted to its PUF (Physical Unclonable Function). Several improvements can be made to this scheme, in particular it is not mentioned how the account address issuing the transactions is built and how it is linked to the PUF. Paper [14] shows that dissociating IoT devices and validator nodes is a powerful architecture that exploits HistoTrust.

2.2 Attestation scheme

The principle of remote attestation is described in depth in [2]. The Trusted Platform Module (TPM) is the targeted device enabling the endorsement of attestation keys that may be owned by the manufacturer, the vendor or the owner. The attestation scheme follows recommendations and standards provided by the Trusted Computing Group (TCG) [1]. Attestation aims at proving to a remote verifier the property of a target by supplying a proof over a network. It consists in three stages : 1) key provisioning, 2) attestation process, 3) verification process.

TPM 2.0 includes an endorsement hierarchy enabling to derive from a secret seed, an attestation key named *ak* identifying the device. For HistoTrust purpose, *ak* is endorsed by the stakeholder owner of the device. In remote attestation schemes, this key is used to sign the TPM's PCR registry in order to prove to a remote verifier the state of the device. This scheme is employed in [15] where an infrastructure provider authenticates a smartphone before issuing confidential data to a service provider. HistoTrust provides an elegant solution to this problem in a decentralized context without infrastructure provider. Others studies as [16] or [17] exploit this attestation scheme to verify software and device integrity. [16] highlights that the real-time requirements of industrial IoT application must be tackled in complement. HistoTrust brings solutions to this request.

In a decentralized root-of-trust architecture, each device is responsible for protecting its secret. With HistoTrust, each device integrates a TPM provisioned with secret keys endorsed per its owner, ensuring root-of-trust. The ST33 TPM provides an EAL-5+ security level. An OP-TEE environment is used in addition through an ARM cortex-A7 to execute operations that are not supported by TPM 2.0 standard. Papers [10] and [18] provide an in-depth analysis of the security level offered by these two components against logical and physical at-

tacks. HistoTrust goes beyond the TrustZone-based wallet detailed in [19] and introduces an Ethereum compliant enhanced wallet relying on a TPM.

3 System requirements

3.1 Use case

In a factory, many actuators participate in the assembly of a product (a car for example) on a production line (see figure 1). These actuators are driven by physical devices that generate digital commands. These devices embed industrial applications that may include embedded artificial intelligence (AI). So, when an incident occurs, creating a financial loss (by stopping the production line for example), it is necessary to find the cause and eventually to charge the costs to the accountable stakeholder. However, the presence of AI makes it difficult to reproduce commands. In some cases, only the analysis of the logs allows to understand what happened and to find the origin. In this context, sharing the attestations of the digital data generated by independent devices that operate on the same production line provides transparency and trust to the stakeholders involved. Moreover for stakeholders who are not physically present in the factory and have left their devices. In case of litigation, each stakeholder should be able to provide its raw data that verifying the shared recorded attestations to an independent auditor, an insurance expert for example, proving that its raw data is authentic, not tampered, complete and ordered.

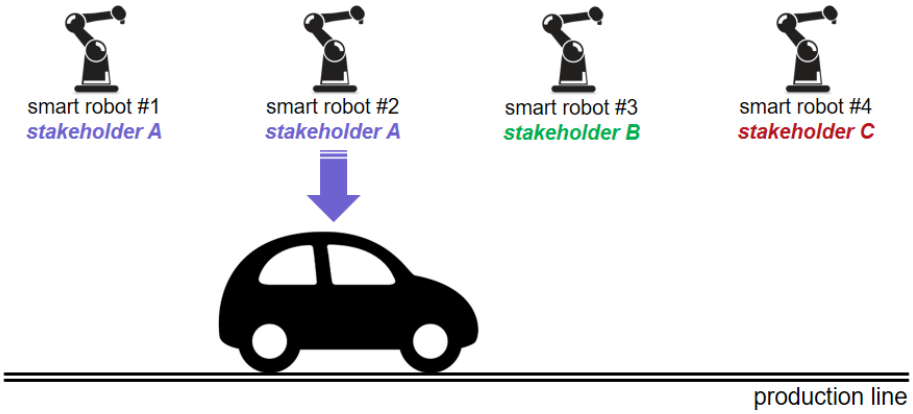


Fig. 1: Illustration of the use-case

3.2 Threat model

The profile of the attacker is that of a powerful stakeholder who, in a multi-stakeholders context, places the blame on another stakeholder of the eco-system for lack of proofs. So, HistoTrust brings counter-measures to these threats:

- a stakeholder who deletes or falsifies his data implicating him
- a stakeholder who falsifies another's data to put blame on him
- infiltration of a malware that generates fake data
- a user who make a mistake turns off the device to erase the proofs

3.3 Requirements

The requirements are formalized below. For a given stakeholder, the aim is to prove that its devices are guenuine and the data they issued are complete and of integrity.

- R1: *maintaining performance*: the security and privacy features shall not impact the industrial application performances.
- R2: *forward integrity*: the data attestation history must be immutable and transparent to the stakeholders. The raw data must be persistent and of integrity.
- R3: *public authentication*: any stakeholder should be able to authenticate the devices issuing data at a given time through the attestations history.
- R4: *power failure*: no raw data or attestations should be lost in the event of a power failure.
- R5: *privacy-preserving data*: The raw data shall not be exposed to the other devices.
- R6: *verifiability*: An accredited auditor must be able to verify the data attestations.
- R7: *multiple stakeholders*: the scheme shall support multiple-stakeholders owning multiple devices issuing data concurrently.

4 Design

This section details the architecture and implementation choices to meet the requirements of the use case.

4.1 Architecture design

All the devices are distributed on a local network in the factory, with an access point to communicate with the outside world. A consortium (permissioned and private) blockchain is deployed. Each stakeholder involved has a validator node, represented by a computer in the figure 2. Thus, the governance of the system is ensured with equity by all the stakeholders involved.

The IoT devices acting in the production line, are provided with an enhanced wallet, enabling to send transactions to the validator nodes. Each device is the

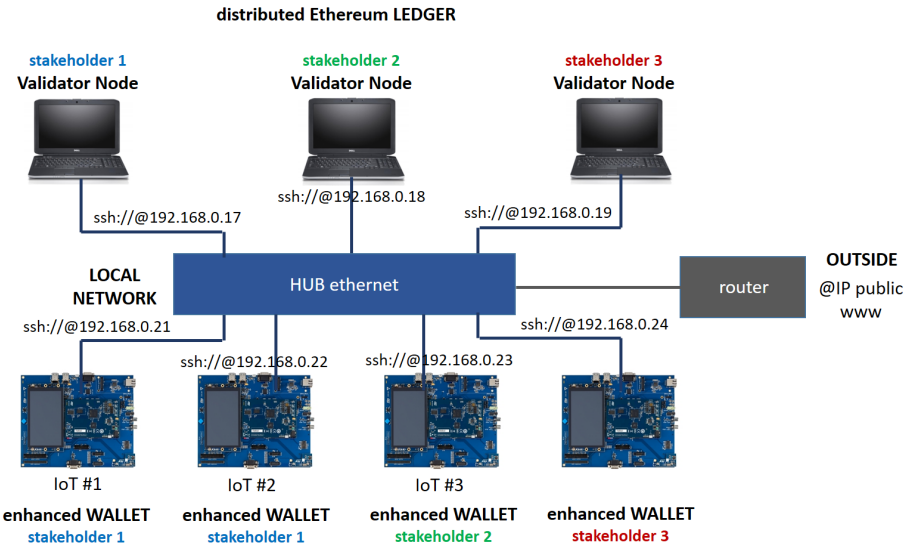


Fig. 2: network architecture

root-of-trust of the data it produced, forming a distributed root-of-trust network. IoT provisioning is done independently by each stakeholder, prior to the deployment of the hardware in the factory. The management of access rights and authorizations is done through smart contracts.

We make the choice to use Ethereum Go geth open-source solution to implement a permissioned blockchain based on a Proof-of-Authority (PoA) "Clique" [20]. Table 1 presents the features of the main blockchains used in the cross-industry domain. This choice is motivated by the availability of a large amount of code and open-source projects coming from the ethereum community, in particular the web3 library. Aiming to be implemented on constrained embedded devices, the code must be optimized at low level.

Table 1: main blockchain features

blockchain	**type**	**smart contract**	**open source**	**wallet crypto**	**consensus algorithm**
ethereum geth	permissioned	yes	yes	secp256k1	PoA Clique
ethereum mainnet	permissionless	yes	yes	secp256k1	PoW etash
hyperledger fabric	permissioned	yes	yes	pkcs#11	PBFT
hyperledger sawtooth	permissioned	yes	yes	secp256k1	PBFT or PoET
iota	permissionless	no	no	winternitz	fragment of PoW

4.2 Secure hardware

This section briefly presents the IoT platform design. A STM32MP157-EV1 evaluation board is associated with a STPM4RasPI TPM Expansion Board. Two independent trusted applications (apps), signed by the platform key, are embedded on the ARM Cortex-A7 microcontroller. One is dedicated to the industrial application and the other to the attestation process. The ARM Cortex-A7 includes an open source Trusted Execution Environment (OP-TEE) implementing the ARM TrustZone technology. At start, a secure boot process is achieved according the application note [21] relying on Brainpool 256 ECDSA key. Then, during execution measurements are achieved to check the integrity of the trusted apps and to monitor that the access rights to the file buffer #1 and #2 have not changed (see figure 4). To enable this measurement, the hash of the binary code of the two trusted app as well as the hash of the access right to the files are provisioned in the TPM PCR registry.

A private key noted *sk* is provisioned and endorsed following the TCG attestation scheme described in [1]. The public certificates required to verify the keys endorsement are recorded in the ledger through smart contracts. This enables all stakeholders to verify that devices issuing data are guenuine in the system. The digital signature with the elliptic curve secp256k1 required for Ethereum transaction is not supported by the TPM 2.0 standard. So, we have implemented this cryptographic function in the TrustZone in order to avoid exposing the private key *sk* to software attacks. This key is ciphered in the TPM key vault and is accessed from the TrustZone via the SPI bus. The evaluation board EV1 presents the benefit to enable the security of the SPI bus at low level.

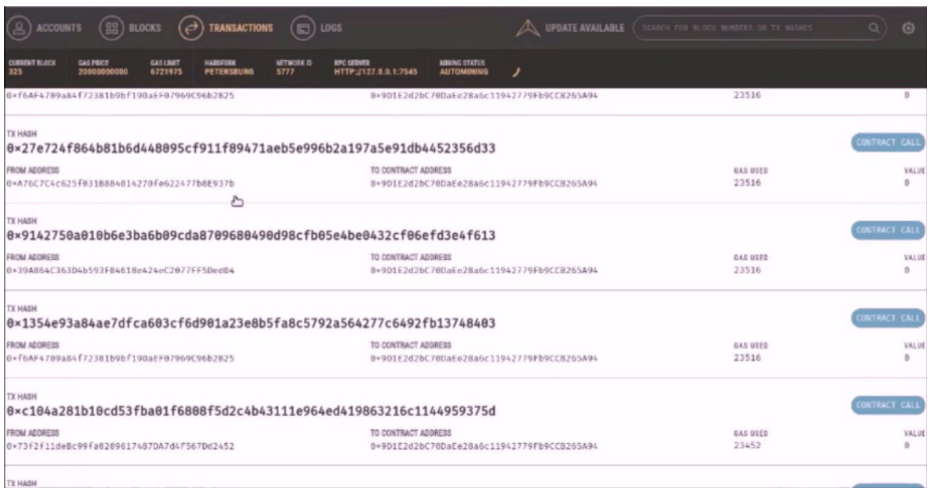


Fig. 3: history of data attestations recorded in the ledger

The attestation process uses *sk* to attest the data produced by the industrial application. During the production phase, the cryptographic attestations are registered in Ethereum ledger through a smart contract. The attestation history is transparent and available to all stakeholders. The figure 3 shows a portion of the ledger with the history of attestations on the data. Each record is a transaction signed by *sk*, emitted from the account of the issuing device, and sent to the smart contract. It includes the hash of the attested data set.

4.3 Implementation

The embedded system depicted figure 4 aims to integrate the security needed to meet the requirements without impacting the industrial application performances. All the data needed to be attested are timestamped and written in the file buffer #1 in real-time. Only the industrial application is allowed to write in this file. The attestation process is implemented as an independent trusted application allowed to read the file buffer #1. The reading of the freshly written data set is triggered by the receipt from the blockchain confirming that the transaction attesting the previous data set is recorded in the ledger.

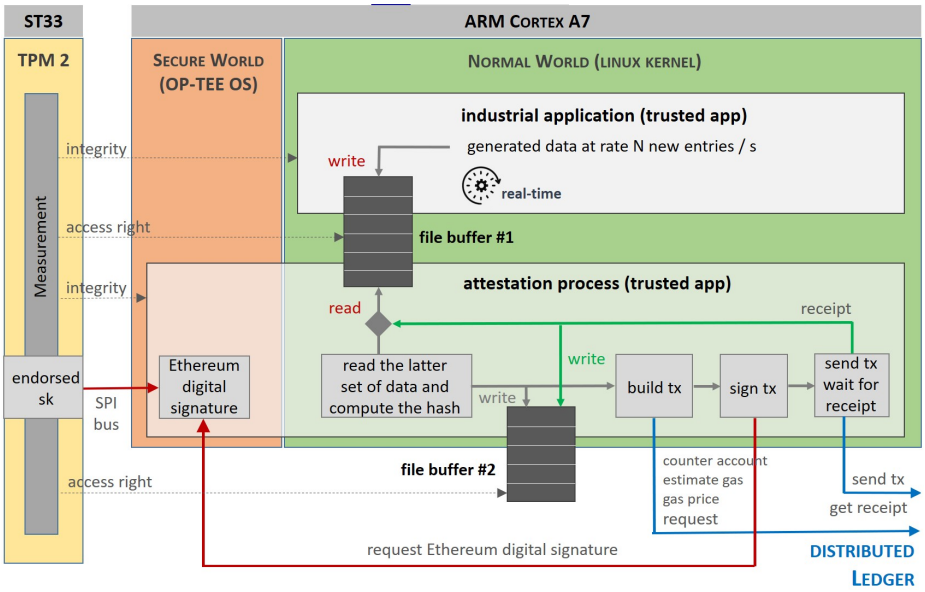


Fig. 4: embedded design in the IoT devices

The attestation process consists in computing the hash of the latter data set produced, that is included in the 'data' field of an ethereum transaction. This transaction is signed with the private key *sk* which is also used to build

the account address issuer. To achieve this signature, the private key *sk* is accessed in the TPM vault through the SPI bus. The signed transaction is sent to blockchain for validation and a receipt is returned if the registration in the ledger is confirmed. This receipt is written in a second file buffer #2. Only the attestation trusted app is allowed to write in this file.

The files #1 and #2 are stored in persistent memory. If a power failure occurs, the data is saved and the attestation process resumes where it left off when the power returns. The content of these files may also be ex-filtrated by their owner stakeholder via VPN through the access point. The read and write access rights to the two files are supervised by the TPM measurement process and an alert is raised if they are modified.

5 Evaluation

The evaluation aims at qualifying the performance of the embedded scheme in order to ensure the security and privacy requirements while keeping the rate and the efficiency of the industrial application.

Four IoT devices are deployed for the experiments. Each is composed of a STM32MP157-EV1 board and a STPM4RasPI TPM Expansion Board including a secure element STSAFE-TPM ST33. Each embeds one of the following configurations concerning the access to the private key *sk*, in order to evaluate the overhead of security on the overall system:

- Iot #1: *sk* is in clear in the OP-TEE non-volatile memory
- Iot #2: *sk* is ciphered in the OP-TEE non-volatile memory
- Iot #3: *sk* is in clear in the STSAFE-TPM ST33 vault
- Iot #4: *sk* is ciphered in the STSAFE-TPM ST33 vault

Each device produces digital data, builds attestations and emits transactions to the Ethereum ledger composed of three validator nodes. In a first stage, we use the Ethereum Ganache simulator to emulate the validator nodes in order to focus on the embedded design and the implementation of the IoT devices. In a second stage, one validator node is deployed per stakeholder involved in the consortium of the distributed system. Each one hosts and accesses the content of the ledger.

5.1 Performance

In this section, the aim is to evaluate the performance of the embedded implementation in the IoT devices according the architecture depicted in the figure 2, in presence of three stakeholders and four IoT devices producing data. One goal is to qualify the impact of security and privacy on the user experience and the execution of the industrial task. Another goal is to evaluate the security and privacy level with respect to the requirements defined in section 3.

The methodology followed consists in a first stage of implementing functional benchmarks of the applications including cryptographic operations on a personal computer (PC) in C language and performing intensive tests. In a second stage,

the code is well structured and transferred to the embedded devices. The functions embedded in the OP-TEE environment that accesses to the TPM, are isolated from those implemented in the Linux userland. Thus, the performances obtained on the PC are presented as functional benchmarks in figure 5, noting that no hardware security is present.

In the following, several experiments are launched with different data rate issued by the industrial application, from 33 entries per second to 10000 entries per second. For each experiment, 20000 entries are considered. An entry is a set of data composed of 50 bytes including such fields:

$$[index][timestamp][rawdata]$$

The *index* field enables to order the data entries. It is followed by the timestamp and the raw data produced by the application. The application used produces data of fixed size. For the purpose of performance testing, the rate of data produced is variable.

The left-hand graph in figure 5 shows the number of transactions sent to Ganache for each data rate. The green curve corresponds to the real-time of the data production, while the orange curve shows the processing time of the processor Core i5-7200U that fluctuates between 1 and 4 milliseconds. The numbers of transactions sent to Ganache follows the needs of the real-time constraints. When the data rate increases, more entries are included in the attested data set in a transaction and globally the number of transactions is smaller. The total processing time follows the amount of transactions built.

The right-hand graph illustrates the amount of entries attested per transaction with regards to the processing time required for one cycle, i.e. the process to build and issue a transaction to Ganache. The timing of the Ethereum ECDSA signature is quite stable whatever the data rate, while the timing of the process to build one transaction increases with the data rate. An in-depth analysis shows that the rise comes from the hash operation that takes more time when the amount of data increases.

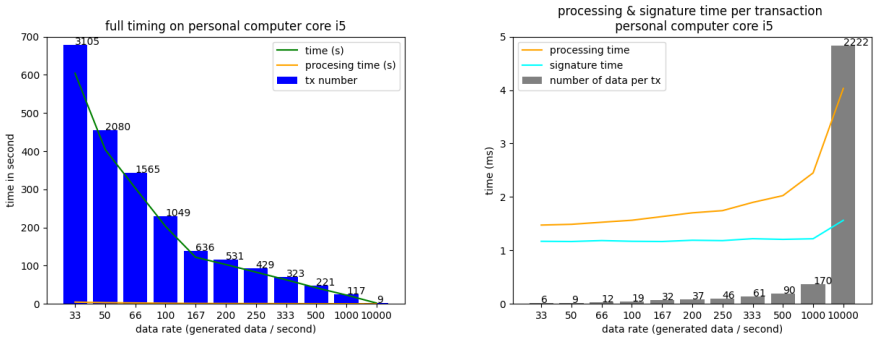


Fig. 5: benchmark on personal computer with Core i5

The figure 6 illustrates the results obtained once deployed onto the four boards running simultaneously. 20000 entries are generated by each board, leading to a number of transactions different according to the transaction processing time, illustrated board by board on figures 7 and 8. The real-time curve shows that the constraint is the same for the boards and for the benchmark PC. The total processing time is close regardless of the board configuration, showing that the use of a TPM to protect the private key with a higher security level, has not a big impact on the whole system performances.

The right-hand graph illustrates the ratio of transactions emitted by each board at a given data rate. When the private key is accessed in the TPM, this increases the timing process to sign the transaction and mechanically, more data entries are included in an attestation and the number of transactions sent to Ganache decreases.

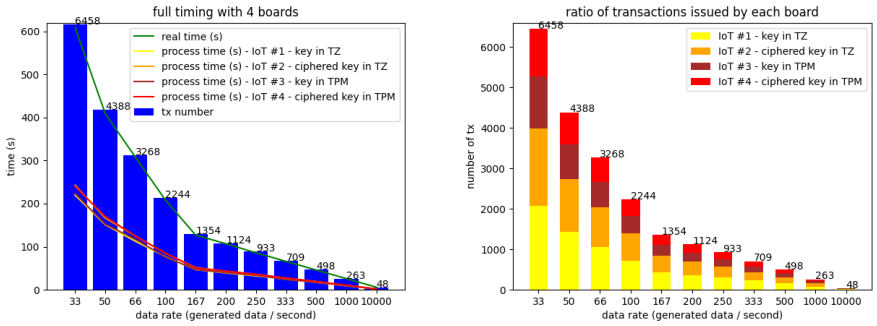


Fig. 6: full time processing with 4 boards

The timings presented in the figures 7 and 8 show that the processing time to build a transaction is quite stable whatever is the data rate for each configuration of access to the private key. The impact of storing *sk* in the TPM rather than disposing of the key directly in the OP-TEE memory is 70 ms for one transaction processing time. Ciphering *sk* takes 25ms additional to execute the AES deciphering before signing.

As reference and comparison, the paper [22] highlights the implementation performances of a IOTA light node on different ARM Cortex-M4, Cortex-M7, Cortex-M3 and Cortex-A53. These microcontrollers do not include secure hardware, but their firmware may be protected from tampering. IOTA uses Winternitz One-Time Signature Scheme [23] known to be robust to side-channel attacks. The processing time of this signature scheme takes 80ms on ARM Cortex-M7, 135ms on Cortex-M4, 683ms on Cortex-M3, 328ms on Cortex-A53. Our implementation of the Ethereum ECDSA signature on the Cortex-A7 TrustZone takes 102ms when *sk* is present in the OP-TEE memory, 173ms when *sk* is accessed in the TPM vault, and 20ms additional if the deciphering of *sk* is done in Trust-

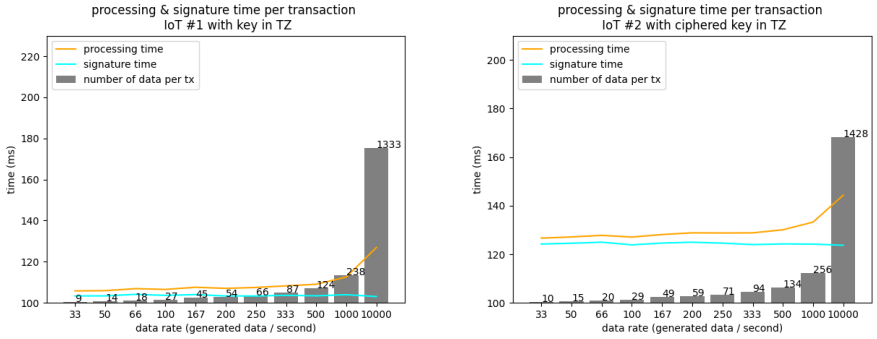


Fig. 7: one cycle processing with private key *sk* in TrustZone

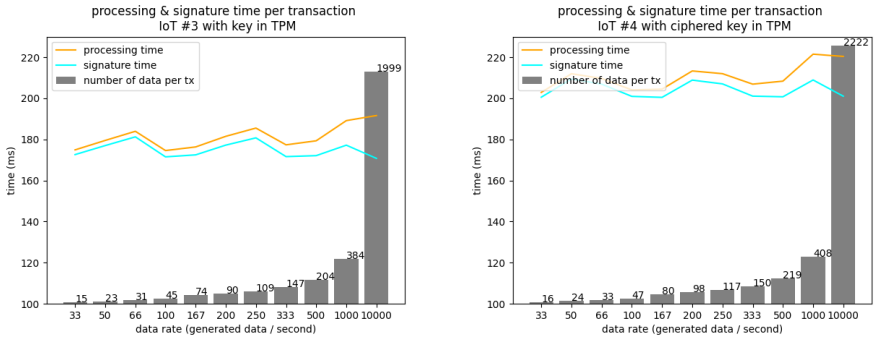


Fig. 8: one cycle processing with private key *sk* in TPM vault

Zone. These results seems very good in regards to the state of the art. These performance analysis show that:

1. the architecture and the embedded implementation design enable to follow real-time application constraints even for high data rates.
2. the storage of the private key *sk* in the TPM does not impact the global performances.
3. the implementation of Ethereum ECDSA signature on the elliptic curve secp256k1 in the TrustZone of the ARM cortex-A7 is quite efficient compared to the state of the art of similar studies.

As conclusion of this section, the implementation of security and privacy by design in embedded devices including TrustZone and TPM does not impact the performances of the industrial application, even when high data rates are considered. So the requirement R1 is fulfilled.

5.2 Security and Privacy

In this section, we examine whether the security and privacy requirements are satisfied.

R2: forward integrity. Solutions such as EmLog [4] or SGX-log [9] ensure forward integrity by forming a history of secure data blocks, based on Schneier technique [8], enhanced by the diversification of encryption keys forming a hash-key-chain, involving complex key management. Based on blockchain technology, which ensures by design the forward integrity of the information recorded in the ledger, HistoTrust introduces a new solution to the problem of log security. The history of cryptographic attestations of the data is maintained in the ledger, each attestation being a pointer to the raw data maintained outside the blockchain by its owner. Thus, any tampering or removal of raw data is detectable.

R3: public authentication. The recorded attestation authenticates the device issuer. It may also authenticate the stakeholder owner if its identity is public to the consortium. The consultation of the attestations history allows any stakeholder having access to the ledger to know:

- the devices that have issued data in a given time interval,
- the order in which the devices performed actions.

R4: power failure. Resilience in the event of a power failure means not losing raw data or cryptographic attestations. The choice of implementation using two independent files monitored in permanent memory ensures data persistence in case of power failure.

R5: privacy-preserving data. The privacy-preserving data requirement covers raw data at storage and during transportation. This requirement makes sense in a multi-stakeholders context where everyone wants to preserve the confidentiality of his data. With EmLog, SGX-log and EngraveChain, data are stored ciphered on a remote back-end common to all stakeholders, possibly in an enclave. With HistoTrust, raw data is stored locally in the memory of the device that produced it, and can be ex-filtrated via a VPN link by its owner. So, the privacy between devices is ensured. However, someone with physical access to the device can read the newly generated data before it is ex-filtrated. Thus, physical protection of the device in the factory is required to make access to the board peripherals difficult and detectable.

R6: verifiability. The correctness of the data history is achieved knowing both the raw data and the recorded attestations. In the context of HistoTrust, how a stakeholder proves to others that his data history is correct without providing them with his raw data? Two solutions are considered: An accredited stakeholder, an insurance expert or a judicial officer for example, could have access to the raw data of each stakeholder, as well as to the ledger, in order to carry out the verifications. Another solution is to share between stakeholders a trusted application that verifies the data history in an OP-TEE environment. Once a secure channel has been established between the OP-TEE and the server hosting a stakeholder's raw data, access to the ledger being authorized to all, the verification is carried out in the OP-TEE and the output report shared with all.

R7: multiple stakeholders. EmLog and SGX-log offer solutions where the number of stakeholders is limited by the technology. HistoTrust brings a solution where the number of stakeholders is not limited by using blockchain technology as a complement to existing technologies. The stakeholders ensure the governance together, each having a validator node.

The table 2 resumes this discussion.

Table 2: the satisfaction of needs by the main schemes

scheme	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**
EmLog [4]	✓	✓	>	>	>	✓	>
SGX-log [9]	✓	✓	x	✓	>	✓	>
EngraveChain [12]	x	✓	✓	x	>	✓	✓
HistoTrust	✓	✓	✓	✓	>	✓	✓

requirement: ✓met, > to improve, x not met

6 Conclusion

HistoTrust brings several contributions beyond the existing ones:

1. a scheme for attesting data histories produced at real-time by industrial applications embedded on independent IoT devices,
2. the deployment of a decentralized root-of-trust network based on the use of a TPM and an OP-TEE environment specific to each IoT device,
3. an architecture ensuring by design end-to-end security and privacy and providing trust within an ecosystem of independent stakeholders.

Among the perspectives considered, we will tackle the privacy-preserving data requirement in order to protect the data confidentiality in the device that produces it. For this, the structure of the embedded code will be revised: some parts of the code will be ported to the TPM in the form of a Java applet and thus becomes resistant to physical attacks. Others will be implemented in the OP-TEE to reinforce the confidentiality of the solution. It is also envisaged to test HistoTrust on mobile IoT devices.

Acknowledgment

This work is a collaborative research action that is partially supported by (CEA-Leti) the European project ECSEL InSecTT [1] and by the French National Research Agency (ANR) in the framework of the Investissements davenir program (ANR-10-AIRT-05, irtnanoelec)

References

1. Trusted Computing Group, "TCG Trusted Attestation Protocol (TAP) Use Cases for TPM Families 1.2 and 2.0 and DICE", Version 1.0, Revision 0.35, November 2019, `https://trustedcomputinggroup.org/wp-content/uploads/TCG_TNC_TAP_Use_Cases_v1r0p35_published.pdf`

[1] www.insectt.eu, InSecTT: ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Unions Horizon 2020 research and innovation program and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. The document reflects only the authors view and the Commission is not responsible for any use that may be made of the information it contains.

2. George Coker, Joshua Guttman, Peter Loscocco, Amy Herzog, Jonathan Millen, Brian OHanlon, John Ramsdell, Ariel Segall, Justin Sheehy, Brian Sniffen, "Principles of remote attestation", International Journal of Information Security, Volume 10, pp. 6381, Springer, 2011, doi: 10.1007/s10207-011-0124-7, https://link.springer.com/content/pdf/10.1007/s10207-011-0124-7.pdf
3. Kang Yang, Liqun Chen, Zhenfeng Zhang, Christopher J.P. Newton, Bo Yang and Li Xi, "Direct Anonymous Attestation with Optimal TPM Signing Efficiency", eprint 1128, 2018, https://eprint.iacr.org/2018/1128.pdf
4. Carlton Shepherd, Raja Akram and Konstantinos Markantonakis, "EmLog: Tamper-Resistant System Logging for Constrained Devices with TEEs", In Proceedings of the 11th IFIP International Conference on Information Security Theory and Practice, WISTP'17, pp. 75-92, Springer International Publishing, 2017, doi: 10.1007/978-3-319-93524-9_5, https://hal.inria.fr/hal-01875526/file/469589_1_En_5_Chapter.pdf
5. Stefan Saroiu and Alec Wolman, "I am a sensor, and I approve this message", In Proceedings of the Eleventh Workshop on Mobile Computing Systems and Applications, HotMobile'10, ACM Publisher, 2010, doi: 10.1145/1734583.1734593, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.155.242&rep=rep1&type=pdf
6. Thomas Hardjono and Ned Smith, "An attestation architecture for Blockchain networks", arXiv:2005.04293 [cs.CR], 2020, http://export.arxiv.org/abs/2005.04293
7. Christine Hennebert and Florian Barrois, "Is the blockchain a relevant technology for the industry 4.0?", In Proceedings of the 2nd IEEE Conference on Blockchain Research & Applications for Innovative Networks and Services, BRAINS'20, pp. 212-216, IEEE Publisher, 2020, doi: 10.1109/BRAINS49436.2020.9223290, https://ieeexplore.ieee.org/document/9223290
8. Bruce Schneier and John Kelsey, "Cryptographic support for secure logs on untrusted machines", In Proceedings of the 7th Conference on USENIX Security Symposium, Volume 7, SSYM98, USENIX Association, 1998.
forensics", ACM Transactions on Information and System Security, 1999.
9. Vishal Karande, Erick Bauman, Zhiqiang Lin, Latifur Khan, "SGX-Log: Securing System Logs With SGX", In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17, pp. 1930, ACM Publisher, 2017, https://doi.org/10.1145/3052973.3053034
10. Dhiman Chakraborty, Lucjan Hanzlik and Sven Bugiel, "simTPM: User-centric TPM for Mobile Devices", In Proceedings of the 28th Conference USENIX Security Symposium, SSYM19, USENIX Association, , pp. 533-550, 2019, isbn: 978-1-939133-06-9, https://www.usenix.org/conference/usenixsecurity19/presentation/chakraborty
11. Carlton Shepherd, Ghada Arfaoui, Iakovos Gurulian, Robert P. Lee, Konstantinos Markantonakis, Raja Naeem Akram, Damien Sauveron and Emmanuel Conchon, "Secure and Trusted Execution: Past, Present, and Future - A Critical Review in the Context of the Internet of Things and Cyber-Physical Systems", In Proceedings of the IEEE Trustcom/BigDataSE/ISPA, pp. 168-177, IEEE Publisher, 2016, doi: 10.1109/TrustCom.2016.0060, https://eprint.iacr.org/2016/454.pdf
12. Louis Shekhtman and Erez Waisbard, "EngraveChain: Tamper-proof distributed log system", In Proceedings of the 2nd Workshop on Blockchain-enabled Networked Sensor, BlockSys19, ACM Publisher, 2019, doi: 10.1145/3362744.3363346, https://dl.acm.org/doi/pdf/10.1145/3362744.3363346

13. Uzair Javaid, Muhammad Naveed Aman and Biplab Sikdar, "BlockPro: Blockchain based Data Provenance and Integrity for Secure IoT Environments", In The 1st Workshop on Blockchain-enabled Networked Sensor Systems, BlockSys'18, ACM Publisher, 2018, doi: 10.1145/3282278.3282281, https://dl.acm.org/doi/pdf/10.1145/3282278.3282281
14. Atis Elsts, Efstathios Mitskas and George Oikonomou, 2018, "Distributed Ledger Technology and the Internet of Things: A Feasibility Study", In The 1st Workshop on Blockchain-enabled Networked Sensor Systems, BlockSys'18, ACM Publisher, 2018, doi: 10.1145/3282278.3282280, https://dl.acm.org/doi/pdf/10.1145/3282278.3282280
15. Urs Hengartner, "Location Privacy based on Trusted Computing and Secure Logging", In Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, SecureComm'08, ACM Publisher, 2008, doi: 10.1.1.216.7307, https://core.ac.uk/download/pdf/21748895.pdf
16. Nikos Koutroumpouchos, Christoforos Ntantogian, Sofia-Anna Menesidou, Kaitai Liang, Panagiotis Gouvas, Christos Xenakis and Thanassis Giannetsos, "Secure Edge Computing with Lightweight Control-Flow Property-based Attestation", In Proceedings of the IEEE Conference on Network Softwarization, NetSoft'19, pp. 84-92, IEEE Publisher, 2019, doi: 10.1109/NETSOFT.2019.8806658, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8806658
17. Roberto Casado-Vara, Fernando de la Prieta, Javier Prieto and Juan M. Corchado, "Blockchain framework for IoT data quality via edge computing", In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems, BlockSys'18, pp. 1924 ACM Publisher, 2018, doi: 10.1145/3282278.3282282, https://dl.acm.org/doi/pdf/10.1145/3282278.3282282
18. Mohamed Sabt, Mohammed Achemlal and Abdelmadjid Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not", In Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrusteCom'15, IEEE Publisher, 2015, doi: 10.1109/Trustcom.2015.357, https://hal.archives-ouvertes.fr/hal-01246364/document
19. Miraje Gentilal, Paulo Martins and Leonel Sousa, "TrustZone-backed bitcoin wallet", In Proceedings of the 4th Workshop on Cryptography and Security in Computing Systems, CS2'17, ACM Publisher, pp. 2528, 2017, doi: 10.1145/3031836.3031841, https://dl.acm.org/doi/pdf/10.1145/3031836.3031841
20. Pr Szili, "EIP-225: Clique proof-of-authority consensus protocol", Ethereum Improvement Proposal, https://eips.ethereum.org/EIPS/eip-225
21. STMicroelectronics, "STM32MP15ROM code secure boot", https://wiki.st.com/stm32mpu/wiki/STM32MP15_ROM_code_secure_boot
22. Diego Stucchi, Ruggero Susella, Pasqualina Fragneto and Beatrice Rossi, "Secure and Effective Implementation of an IOTA Light Node using STM32", In the Proceedings of the 2nd Workshop on Blockchain-enabled Networked Sensor, BlockSys19, ACM Publiher, 2019, doi: 10.1145/3362744.3363344, https://dl.acm.org/doi/pdf/10.1145/3362744.3363344
23. Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hlsing and Markus Rckert, "On the Security of the Winternitz One-Time Signature Scheme", In Progress in Cryptology, AFRICACRYPT 2011, pp. 363378, Springer Berlin Heidelberg, 2011, doi: 10.1007/978-3-642-21969-6_23, https://eprint.iacr.org/2011/191.pdf