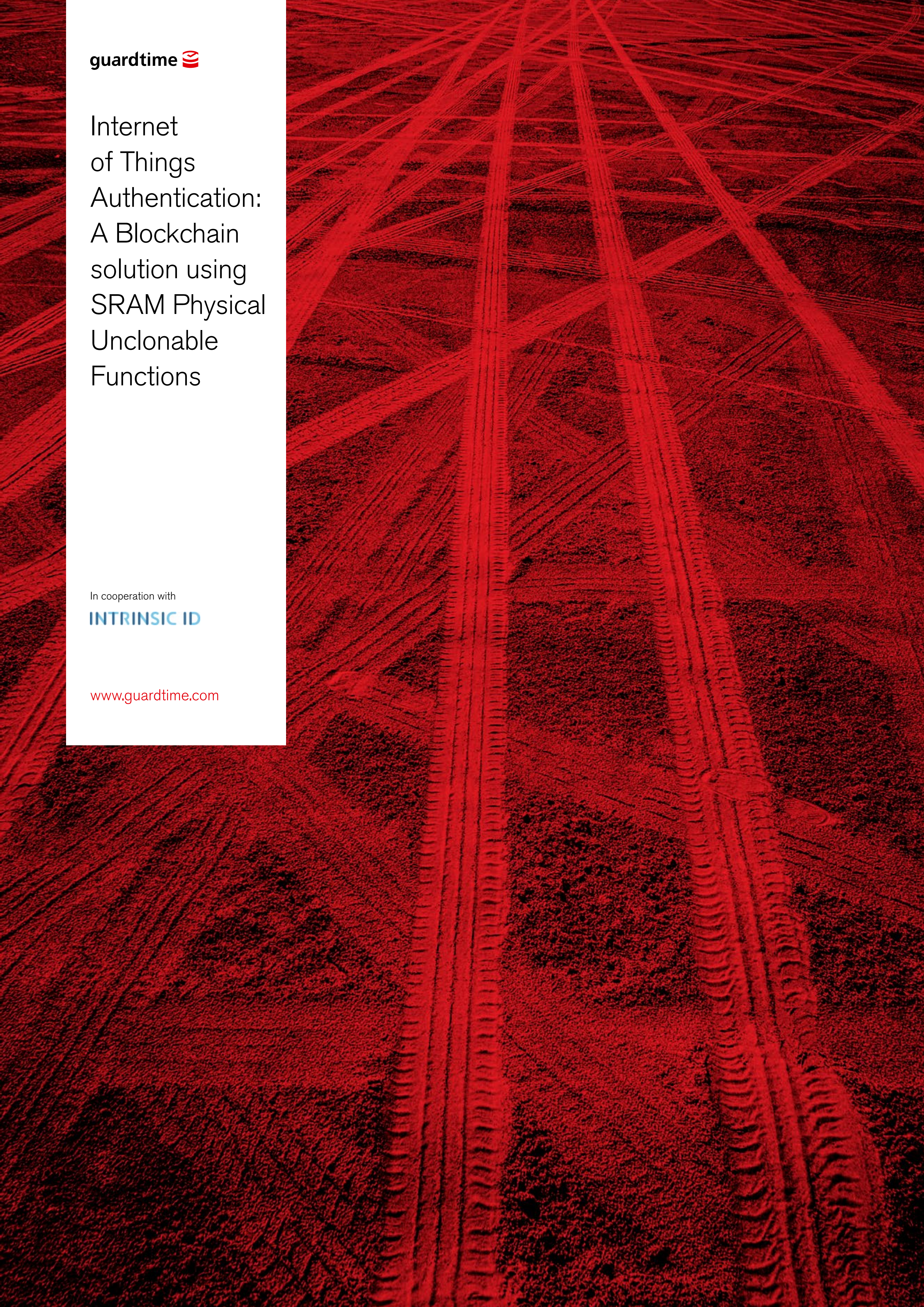
guardtime

Internet of Things Authentication: A Blockchain solution using SRAM Physical Unclonable Functions



In cooperation with

INTRINSIC ID

www.guardtime.com

Introduction

Abstract
Authentication of devices and data is a key success factor for the Internet of Things. A single compromised node can be turned into a malicious one that brings down whole systems or causes disasters with cars, planes, drones, the grid etc. on a scale that humanity has never seen before. In this white paper we therefore propose to combine two key authentication technologies: Blockchain and SRAM Physical Unclonable Function (PUF). We will explain how this is done and discuss its benefits to Multi-Factor Authentication, Continuous Authentication and Provenance Trails.

Keywords: *Blockchain, PUF, Physical Unclonable Function, Authentication, Integrity, Device Fingerprint, Multi-Factor Authentication, Continuous Authentication, Provenance Trail*

The Internet of Things, a term used for a network of devices comprising refrigerators, security cameras, cars, planes, computers etc has been around for a while under different forms and names. The ability to connect, communicate with, and remotely manage an incalculable number of networked, automated devices is growing at an alarming pace. With increased dependency on intelligent, interconnected devices in every aspect of our lives, it becomes imperative to provide security and privacy. Security and Privacy will become part of the quality measures that guarantee reliable operation of connected devices and compliance with stringent regulatory requirements where applicable. There is a battle for standards with lots of proprietary protocols and security - in particular data integrity has been on the back burner and acquired a minor role in the debate.

Technology standards for IoT networks have not evolved yet, but many stakeholders are starting to focus increasingly on the security aspects. Security in IoT has to be implemented at various layers – the supply chain, the chip, OS, SW, device, network and the system level. On top of this it needs to be adapted to the constraints presented by the devices that comprise the IoT network.

The IoT environment has several constraints:

- Real-time infrastructures cannot be brought down for security updates and patching
- Low-latency, proprietary protocols limit the ability to deploy antivirus and anti-malware software
- Embedded processors have limited processing power and memory to execute security software
- IoT devices have a small form factor, limited connectivity and are designed for very low power consumption
- Many IoT devices are physically accessible to the attacker

Firewalls, Intrusion Detection Systems/Intrusion Prevention Systems (IDS/IPS) systems, Security Information and Event Management (SIEMs), antivirus software and various types of access controls help keep malicious activity off the IoT networks. There is however a glaring gap with respect to the practical concerns with IoT security that include immutable proof of software integrity and device authentication.

IoT devices send out sensitive information that must be protected from unauthorized usage or disclosure. A key primitive to achieve this goal is to establish an immutable trust-base that cannot be tampered with.

Despite all these threats, two key areas of IoT security that have not received much attention are:

- Software integrity: Ensuring the authenticity and integrity of the software on the device. These measures will guarantee that only the software that has been authorized to run on that device, and signed by the entity that authorized it, will be loaded.
- Device authentication: Authentication of the end devices before they can transmit or receive information

Therefore the known shortcomings of knowledge-based authentication approaches like passwords and PINs have to be augmented with standard solutions like PKI in conjunction with new technologies like Physical Unclonable Functions (PUFs). These provide measures to strengthen IoT security from a self-enforced identity perspective. Most commonly used authentication approaches are based on online trust anchors/trusted third parties whereas a PUF based solution provides an offline method with a tamper resistant ID and resiliency. The PUF technology aids the shift from a user-centric world to a device-centric one by enabling:

- **Trusted Discovery/Enrollment –** Secure registration of the IoT device
- **Trusted Interaction** – Authenticity and integrity of the communication amongst the IoT devices in the network

Using a Blockchain to store data that has been secured with PUF derived keys and attributes provides an immutable assurance that data has not been tampered with, in addition to providing traceability and transparent auditing capabilities.

PUF Technology

Physical Unclonable Functions or (PUFs) use device unique random patterns to differentiate chips from each other. PUFs, and SRAM PUFs in particular, are designed to be impossible to duplicate, clone or predict. This makes them very suitable for applications such as secure key generation and storage, device authentication, flexible key provisioning and chip asset management. PUFs are actively stimulated and executed to exploit the randomness in their behavior. A good way to look at a PUF is as a device fingerprint.

The noisy behavior of this device fingerprint is also utilized to the advantage of the system. The noise entropy is harvested to create strong, independent random numbers with high entropy. Strong independent random number generators are needed in all kinds of cryptographic protocols and are often the weakest link in a cryptographic implementation.

Intrinsic-ID's SRAM PUF Technology is the most secure and robust approach to embedding PUF in integrated circuits. The SRAM PUF makes use of the unique characteristics deep down in the transistors of the SRAM Memory inside the device. Due to deep-submicron manufacturing process variations, every transistor in an Integrated Circuit (IC) has slightly different physical properties. These lead to measurable differences in terms of electronic properties like threshold voltage and gain factor. Since these process variations are uncontrollable during manufacturing, the physical properties of a device can neither be copied nor cloned. It is impossible to purposely create a device with a given electronic fingerprint.

The SRAM PUF is used to derive a device-unique cryptographic key. Since the fingerprint is noisy, a Helper Data algorithm or Fuzzy Extractor is needed to reconstruct exactly the same cryptographic key every time and under all (environmental) circumstances i.e in Death Valley, in Alaska and twenty five years from now. For details on the reliability we refer to [SSFP].

This way of deriving a key from the unique fingerprint of the device has great security advantages compared to traditional key storage in non-volatile memory. Because the key is not permanently stored, it is not present when the device is not active (no key at rest) and hence cannot be found by an attacker who is opening up the device and compromising all the memory contents. Additionally, it provides a flexible way of provisioning keys into devices that is scalable (towards billions of devices), secure and reduces the liability for the semiconductor manufacturer. For details we refer to [KP].

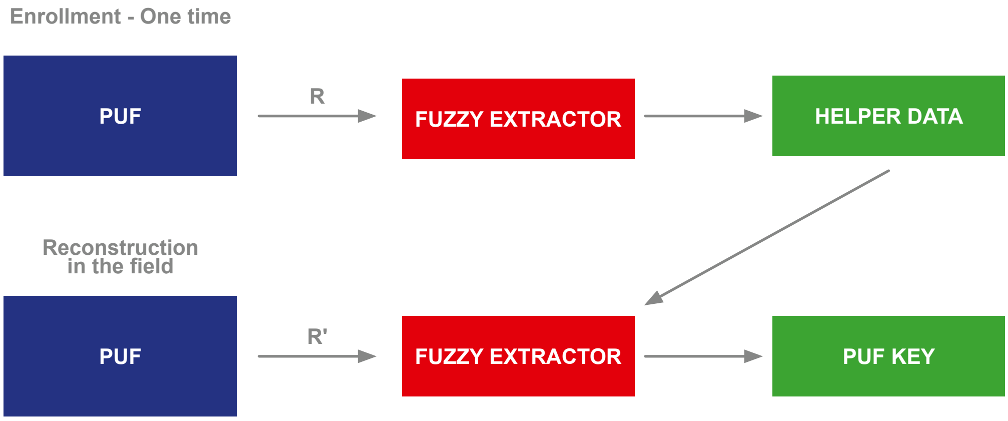


Figure 1: *Enrollment and reconstruction phase for the generation of PUF keys (Note - R is the initial PUF response during enrollment and ' is the PUF response in the field with a noise component*

Keyless Signature Technology

Keyless Signatures Infrastructure™ (KSI) is a data-centric security technology based on cryptographic hash functions, requiring only the use of hash-values and binary trees. KSI-based detection of modification and attribution of those changes requires only access to a record of *widely witnessed events*, which take the form of regularly published codes or root hash values. The actual data under KSI-protection is never exposed, and the need to store cryptographic keys or transfer sensitive information over the signing infrastructure is eliminated. KSI provides firmware code, session and data integrity between the physical device and external IoT systems.

Guardtime's ledger is separate from the KSI blockchain. The Blockchain acts as an integrity layer for the ledger sealing the ledger state and providing a means of portability of ledger entries. Each ledger is created for a particular use case and is private to the customers using that ledger.

The KSI offers:

- Executable integrity – Integrity of the code executed by the device
- Session log integrity – Integrity of the log of entities that communicated with the device
- Machine data integrity – Integrity of the data collected by the machine
- Auditability - Independent mathematical audit trail for what happened accross all networks and devices
- Traceability – Record and play back events over time to aid in discovery and root cause analysis
- Immutable Assurance – Verification of the reliability and integrity of the data, preserving time and authenticity
- Identity - Authentication and authorization of physical devices with IoT applications

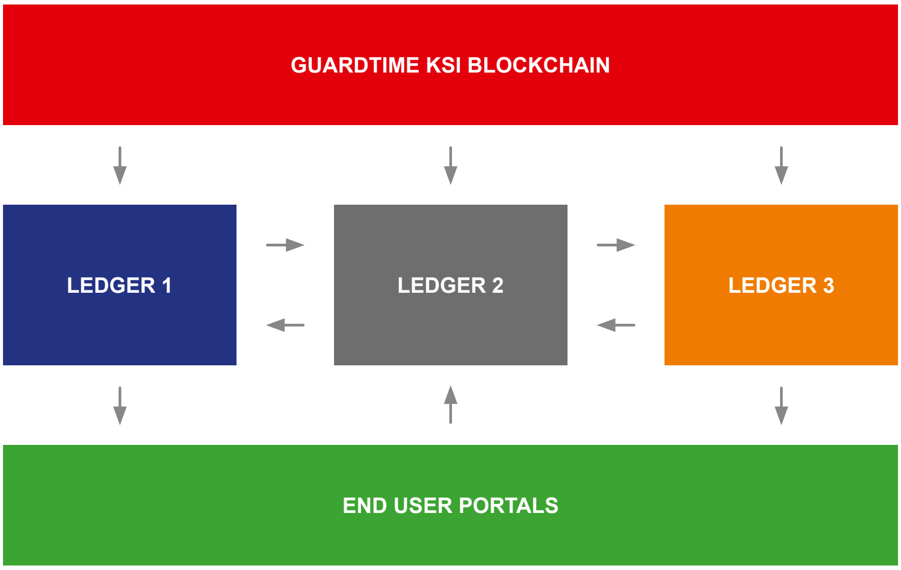


Figure 2: *KSI Blockchain and Ledger*

- Scalability – With billions of devices being added to the IoT network, KSI offers a scalable infrastructure to verify the integrity of critical assets
- Attribution – By enriching all data with a new form of metadata, all activity can be attributed back to a source and time
- 100% accountability – Data events are captured and record the time, asset integrity, and signer origin
- Immutable ledger – It is impossible for anyone to tamper with the calendar block chain
- Universal time source – Time is an inherent property of the KSI system so that events can be unified across distributed systems
- Auditing – Independent verification of digital assets is accomplished without disclosing the underlying specifics of the data. In this way, enabling trusted long-term retention of critical assets within archive repositories.
- No single point of failure - since the core and aggregation network are fully distributed systems.

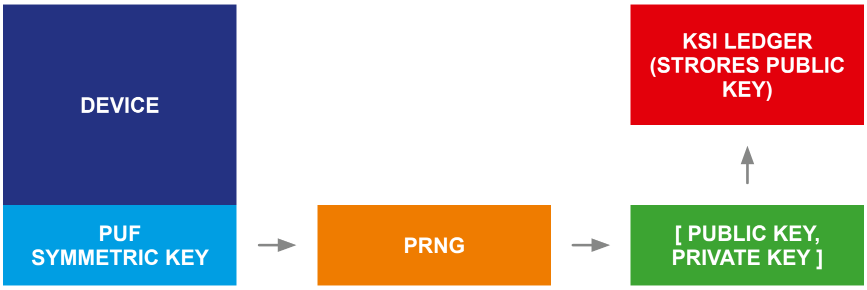


Figure 3: *PUF Asymmetric Key Generation*

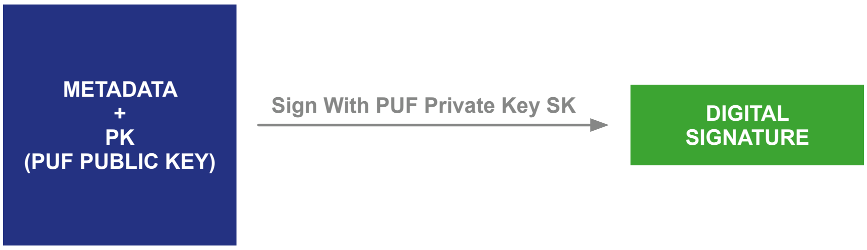


Figure 4: *Generating a PUF signature based on PUF private key*

Use Cases Combining SRAM PUFs and KSI

☑ Multi-factor authentication with SRAM PUF and biometric credentials

Since current policy and industry best practices require a SmartCard for authentication and authorization, developing new authentication and authorization schemes that use true Multi-Factor Authentication is not possible. SmartCards can only provide “something you have” (Card / Private Key) and “Something you Know” (PIN). Due to their form factor, SmartCards cannot provide multi-factor authentication schemes that implement biometrics, location, proximity etc. The DoD has struggled to find the best way to integrate smart ID cards with Government Furnished Equipment (GFE) like smart phones.

We propose a way to avoid the use of a smart card but allow a mobile device to authenticate a user based on a multiplicity of factors. On a mobile device, biometric credentials in conjunction with the PUF-generated keys can be used as a multi-factor authentication mechanism to authenticate that the intended user is the one managing/using the device. The biometric credentials are run through feature-extractors and a biometric template file is derived. We distinguish between two situations. In a first situation the biometric authentication is performed on a server. Then, the biometric template file along with an associated device ID is securely stored on the ledger during enrollment. A second method is to perform the biometric authentication locally on the device without storing the template file on the ledger. Depending on the footprint of the end IoT device, the device can locally verify the biometric credentials and use the result to unlock the PUF symmetric key.

Below we describe the steps that outline how an SRAM PUF based solution in conjunction with a Blockchain/ledger can be used to authenticate IoT devices:

Step 1

Generate a PUF Symmetric Enrollment Key **(K)** - which is the same as PUF key in Figure 1 using Intrinsic-ID’s key generator [SSFP]. The SRAM PUF response along with internally generated random data will be used to derive Helper Data. Helper Data is non-sensitive public data that is used by the Fuzzy Extractor to correct noisy PUF bits and extract the same cryptographic key every time. The Helper Data will be stored close to the end IoT device for retrieval during the reconstruction phase.

Step 2

This symmetric enrollment key will be run through a PRNG to derive an asymmetric public/private key pair **(PK, SK)**. The public key **(PK)** is stored on the ledger.

Step 3

Create a data structure **(PK, E_K(B), D-ID, H(PK), A)** whereby:

- **PK** is the SRAM PUF based public key
- **E_K(B)** is the encryption of the extracted biometric template file **(B)** with the symmetric enrollment key **(K)**,
- **D-ID** is a device ID[1] (to associate the biometric data with)
- **H(PK)** is the hash **(H)** of **PK**, it is used as a unique fingerprint-ID for ledger lookup
- **A** stands for any other deterministic attributes.

[1] We note that the device ID can also be derived from the SRAM PUF.

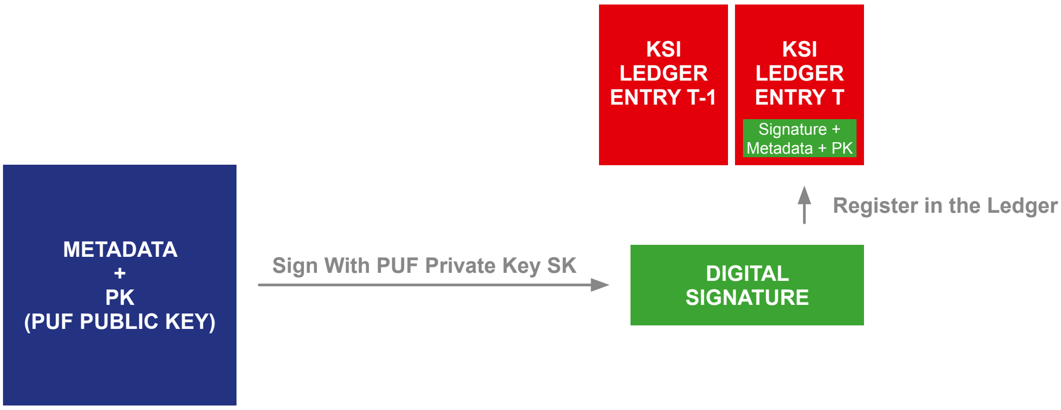

KSI LEDGER ENTRY T-1
KSI LEDGER ENTRY T
Signature + Metadata + PK
Register in the Ledger
METADATA + PK (PUF PUBLIC KEY)
Sign With PUF Private Key SK
DIGITAL SIGNATURE

Figure 5: *Register PUF based signature in the KSI ledger*

This data structure **(PK, E_K(B), D-ID, H(PK), A)** is signed with the private key of the PUF **(SK)**. We denote the signature with **$(PK, E_K(B), D-ID, H(PK), A)**. Note that we encrypt the biometric template file before storing on the ledger to avoid transmitting the file in the clear which makes the communication prone to man-in-the-middle attacks.

Step 4

Register this signed data structure **[(PK, E_K(B), D-ID, H(PK), A),$ (PK, E_K(B), D-ID, H(PK), A)]** in the IOT ledger. The device has now been added to the ledger (after checking this device has not been registered before). This ledger entry will now be associated with the user who owns the device. Note that a user can own multiple devices.

Step 5

When a user wants to authenticate her/him self to the service, the following steps are executed.

1. The device captures a fresh biometrics **B**' of the user "
2. The symmetric enrolled key is reconstructed using the current noisy PUF response and the Helper Data retrieved from the device
3. The corresponding asymmetric key pair **(SK, PK)** is derived
4. Using the key fingerprint ID **(H(PK))**, the corresponding encrypted biometric template file **(E_K(B))** is retrieved from the ledger
5. The data **(E_K(B))** is decrypted and compared locally to the fresh measurement **B'** captured on

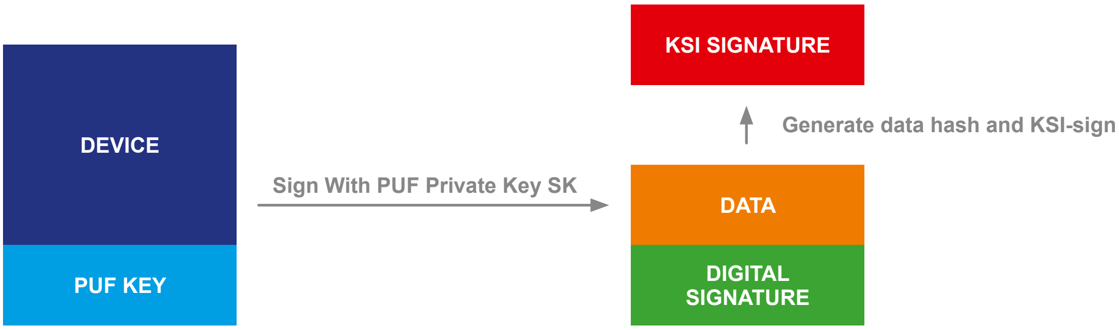


Figure 6: *Generate KSI signature on hash of the data generated by the end device*

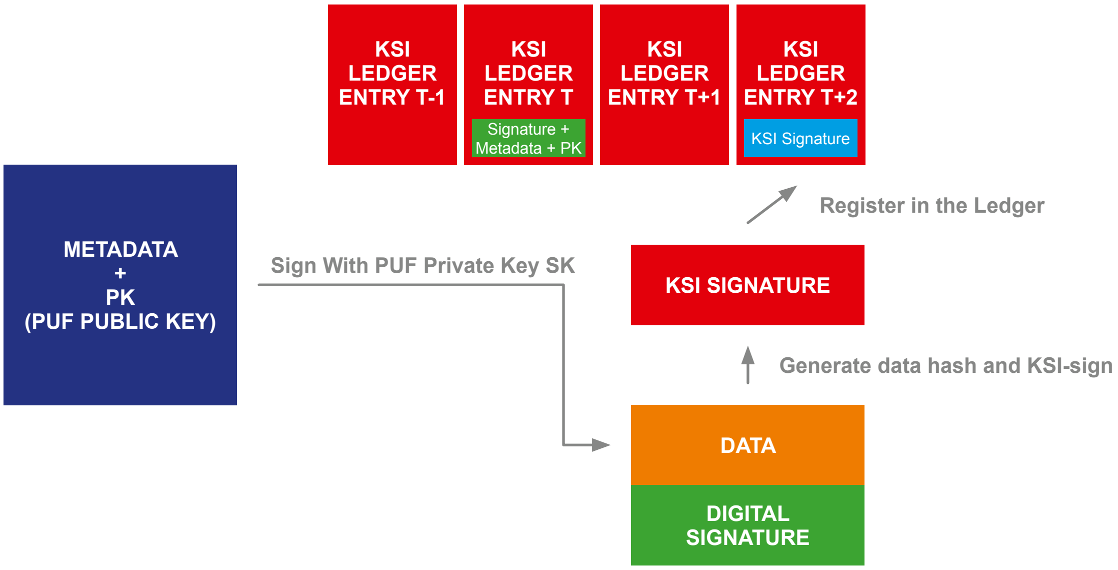


Figure 7: Register KSI signature into the ledger

the IoT end device. If the fresh measurement **B'** matches with **B**, the device has authenticated the user and it will continue its operation.

Alternatively, the biometric credential verification can happen local to the IoT device as described before.

Step 6

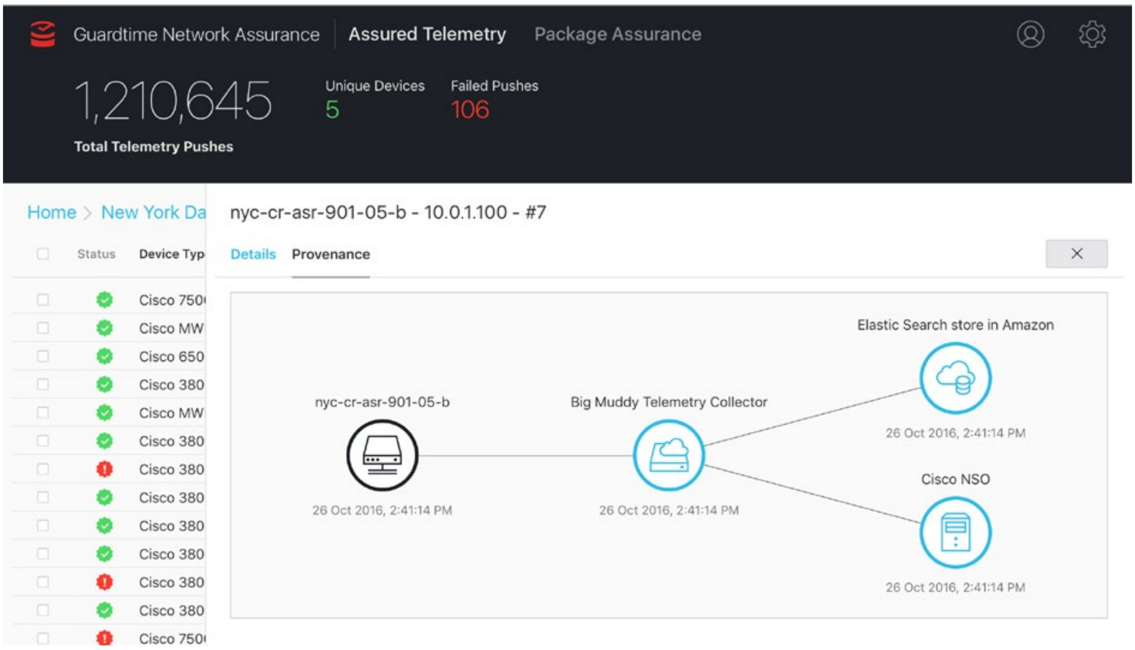
After device initiation, it is assumed that biometric authentication is not needed for continued trust. When the device generates data **(D)**, it is signed with the PUF private key **(D, $D)**. The PUF signature allows the data to be authenticated as having come from the device. A hash of the data is then signed with KSI to ensure integrity: **(H(D), $H(D))**.

Step 7

The KSI signature is then registered in the ledger.

☑ Continuous authentication for managed vehicles

Access control in the context of Unmanned Aerial Vehicles (UAV) is crucial. It ensures that access is authorized properly and only authenticated devices can enter certain areas of operation. First, using KSI, the vehicles could prove that i) their contextual information/configuration is ok and ii) the code they are executing is in a known good state. Each time they enter certain areas of operation, the UAVs and the command post would mutually authenticate to each other, with each UAV being uniquely identified using a SRAM PUF Key. The KSI provides the exchange medium to facilitate this transaction without requiring heavy encryption algorithms. Additionally, KSI also provides the audit trail and accountability for any updates to the vehicle machine code or coordinate information.



1,210,645 Total Telemetry Pushes

Unique Devices 5

Failed Pushes 106

nyc-cr-asr-901-05-b - 10.0.1.100 - #7

☐	Status	Device Typ
☐	✓	Cisco 750
☐	✓	Cisco MW
☐	✓	Cisco 650
☐	✓	Cisco 380
☐	✓	Cisco MW
☐	✓	Cisco 380
☐	!	Cisco 380
☐	✓	Cisco 380
☐	✓	Cisco 380
☐	✓	Cisco 380
☐	!	Cisco 380
☐	✓	Cisco 380
☐	!	Cisco 750

nyc-cr-asr-901-05-b: 26 Oct 2016, 2:41:14 PM

Big Muddy Telemetry Collector: 26 Oct 2016, 2:41:14 PM

Elastic Search store in Amazon: 26 Oct 2016, 2:41:14 PM

Cisco NSO: 26 Oct 2016, 2:41:14 PM

Figure 8: *Sample screenshot of a KSI-enabled provenance trail*

☑ Provenance trail

Data sets are increasingly under significant scrutiny, due to compliance and regulatory guidelines, or unfolding laws around data privacy and residency. In order to enable this process, Provenance can be used. Provenance is a process that ascertains the quality and lineage of data based on the data origin in a warehouse, its derivations and the nodes it went through. It allows re-enactment of transformations to update the data of interest. And finally, it also helps to provide an audit trail for regulatory purposes. A Provenance process based on KSI and SRAM PUF provides the following features:

- **Assured Identity** - Authentication based on identity is paramount to security of IoT systems for impersonation prevention. KSI in conjunction with the SRAM PUF-based keys and identifiers of the endpoint devices offers a means to cryptographically assure robustness of endpoint identities. Therefore one includes this SRAM PUF identity as part of the KSI signature. The provenance trail, ensures that the path of the IoT data is verifiable and auditable.

- **Proof of participation** – Additionally, the combination of KSI with the SRAM PUF Identities of the intermediate nodes provides a proof of participation of each node in any given hash chain (with the PUF ID uniquely identifying the node). This prevents that malicious nodes can hide their tracks.

Conclusion

There is no unique or revolutionary solution tailored for protecting integrity of assets in an IoT network. Whether accidental or malicious, interference with the controls of a pacemaker, a car, or a nuclear reactor poses a threat to human life.

Authentication mechanisms can no longer be based on simply stored secret data or PKI as the secret keys could get stolen or leaked. This issue is addressed by using the physical properties of a device (the device fingerprint) as the unique secret keys for authentication purposes. These physical properties are unpredictable and unclonable due to un-controllable process variations during manufacturing. The most secure, robust and deployed PUF is the SRAM PUF. Hence, SRAM PUF based IDs are unique per device and can be used for authentication in addition to aiding the creation of unique cryptographic keys.

Security and Authentication in particular cannot be thought of as an add-on feature, but have to be an integral part of the device's reliable functioning. Due to its importance and to avoid disasters, Security and Authentication will become a necessary requirement for IoT just as quality.

When it comes to the widespread IoT ecosystem, we propose authentication and integrity schemes based on a combination of two strong and proven technologies: SRAM PUF and KSI Blockchain. This combination provides a scalable, widely witnessed Blockchain technology that provides Multi-Factor Authentication, Continuous Authentication and Provenance Trails.

References

[KP] Secure Key Provisioning with Quiddikey: in process

[SSFP] SRAM PUF: The Secure Silicon Fingerprint, https://www.intrinsic-id.com/physical-unclonable-functions/free-white-paper-sram-puf-secure-silicon-fingerprint/

TRAIL