VTT Technical Research Centre of Finland

Security of Field Devices in Future Water Management

Koskela, Pekka; Kylänpää, Markku

Published in:
Sensors & Transducers

Published: 05/01/2024

Document Version
Publisher's final version

Sensors & Transducers, Vol. 264, Issue 1, February 2024, pp. 11-18

Sensors & Transducers

Published by IFSA Publishing, S. L., 2024
http://www.sensorsportal.com

Security of Field Devices in Future Water Management

[1, *] Pekka KOSKELA and [2] Markku KYLÄNPÄÄ

[1] VTT Technical Research Centre of Finland Ltd., Kaitoväylä 1, Oulu, 90570, Finland
[2] VTT Technical Research Centre of Finland Ltd., Kemistintie 3, Espoo, 00240, Finland
[1] Tel.: +358 20 722 111, fax: +358 20 722 7001
[*] E-mail: Pekka.Koskela@vtt.fi

Received: 13 December 2023 /Accepted: 5 January 2024 /Published: 28 February 2024

Abstract: Water management as a part of critical infrastructure is undergoing transformation alongside the advancement of digitalization. Future water management systems will incorporate both edge and cloud services. Increased connectivity of systems and the use of remote management together with growing heterogeneity and complexity of systems will bring new demands and challenges for security systems. In order to address these future security challenges, we study the zero trust approach and its possible realization with a physical unclonable function facility. Especially in our focus are resource-constrained devices like sensors in the field and their safety.

Keywords: Sensors, Energy efficiency, Zero trust, Physical unclonable function, Hashgraph, Water management.

1. Introduction

In the future, the needs of remote connections will increase, where users use and connect to the water management and control systems remotely. Increased connectivity also enables new services, like remote software updates for Internet of Things (IoT) devices in the field. Monitoring, control, and management operations can also be done over the Internet. These needs must be fulfilled both in normal and in abnormal situations. For example, during a pandemic situation control room and office work was restricted. This also caused changes in work practices in water management and accelerated technology adoption [1].

These requirements and technology development will lead water management systems towards cloud environments, where both edge computing and cloud services will be available. However, this development will also increase attack surface, emphasizing the need to develop holistic security architecture.

There are few security architecture proposals for water management systems. Ntuli and Abu-Mahfouz describe an architecture that is focused on securing access to IoT devices [2] utilizing common IoT protocols, e.g., Constrained Application protocol (CoAP) and MQ Telemetry Transport (MQTT) and publish-subscribe communications model. The importance of secure boot and secure firmware update is also mentioned. Xia et al. [3] describe a blockchain-based system that is focusing on decentralizing transactions between the actors and is also proposing the use of peer-to-peer (P2P) networks for communications. Lee et al. [4] describe the future water management platform. Their system relies on bi-directional communication, intelligent network, and centralized management of smart water grid.

In this paper, we present our work towards security architecture for future water management systems. The main contributions of our work are:

- Utilization of Physical Unclonable Function (PUF) as basis for immutable device identity;
- Impact of Zero Trust security architecture and its realization;
- Discussion of system integrity verification and secure software update topics.

The rest of this paper is structured as follows. First, in Section 2 we give a brief overview of water management. Next, in Section 3 we introduce a key security primitive called Physical Unclonable Function (PUF). In Section 4 we describe elements of the Zero Trust (ZT) security architecture. Then, in Section 5 we discuss potential threats and limitations of our approach. Finally, in Section 6 we conclude the paper.

2. Water Management

Water management consists of both fresh water supply and distribution and wastewater treatment and disposal. These operations are part of the critical infrastructure of society that should work reliably in every situation. Water management IoT devices are sensors, pumps, and valves. These can be controlled and monitored on site or remotely from a control room. Water management system architecture depends on the size of the management system and the level of digitalization. It is expected that future water management systems utilize multiple new technologies including cloud-based services [5]. This trend has also been recognized by cloud providers [6].

In Fig. 1 we present our generic view of the potential future water management architecture, where remote connections and cloud services, both internet and edge clouds, are used. Some of the field devices can be directly connected to edge services to achieve fast response time or improved security in the sense of reduced attacking surface, where information is used locally near the source. The next operation level devices are connected to the control room with field buses, for instance industrial ethernet (Ethernet/IP, Profinet), Modbus or Profibus. The control room is connected to the Internet so that it is possible to utilize cloud services. However, cloud dependency should be analyzed from a reliability and security point of view. Critical functionality should also work without cloud connectivity. Moreover, the control room can also support remote connections. In cases, when the movement of citizens is restricted, like during the COVID-19 epidemic, it may be necessary to have capability to do remote management outside of the control room. Also digitalization of water management systems and increased heterogeneity of suppliers and devices will increase requirements to have remote connections and to keep the devices up-to-date and secure.

Cloud environments with remote connections will open new attack surfaces and increase potential vulnerabilities of the system.

In that kind of environment, it is not enough to trust firewalls as barriers. Trust between all actors including users, services and applications of the system must be kept up continuously. In order to tackle this challenge, we study exploitation of a zero trust concept together with a physical unclonable function (PUF) concept for future water management.

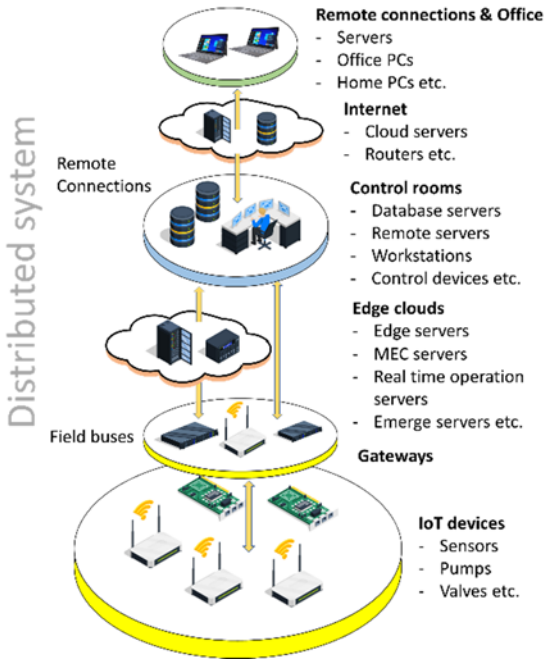

Fig. 1. Future water management architecture.

3. Physical Unclonable Function

PUF (physical unclonable function) is a security technology that creates unique, secure, and unclonable identifiers or keys from the physical characteristics of a device or component. PUFs have the following features: they are robust (they do not change over time), unique (no two PUFs are the same), easy to evaluate (they can be implemented practically), hard to replicate (they cannot be duplicated), and very difficult or impossible to predict (their responses cannot be guessed). The physical characteristics that PUFs use can be based on various types of differences, such as electric, optical, chemical, acoustic, and thermal noise. For example, electric PUFs can originate from differences in bistable state, time, voltage-current, and capacitance [7]. One type of electric PUFs is silicon based PUFs, which use the natural variations and imperfections that occur during the fabrication of integrated circuits or other physical devices. Some examples of silicon-based PUFs are SRAM-PUFs, ring oscillator PUFs, and arbiter PUFs. For instance, SRAM-PUF assigns a value to each SRAM cell based on its unique bistable state when it is powered up. The value can be zero, one, or unstable and unreachable, as shown in Fig. 2.

During manufacturing, the system can create digital fingerprints based on its unique physical properties i.e. PUFs. These fingerprints are the result of applying different challenges (such as bistable state, see above, voltage-current, time etc.) to the system and recording the corresponding responses. When the system faces the same challenge again, it produces the same response, which verifies its identity. PUFs can be classified into weak and strong PUFs, depending on how many challenge-response pairs they can generate.

Weak PUFs have only one or a few pairs, while strong PUFs have many pairs. The main advantage of PUFs is that they are theoretically unclonable because the physical variations that cause them cannot be easily replicated. This makes them useful for various security-critical applications. However, PUFs also have challenges, such as noise, reliability, and the need for proper error correction and management to ensure their practical utility in secure systems. The main purpose of a PUF is to establish a local root of trust in a device or a system, which can then be used for a wide range of applications, such as secure key storage, device authentication, remote attestation, secure bootstrapping, anti-counterfeiting, and more.

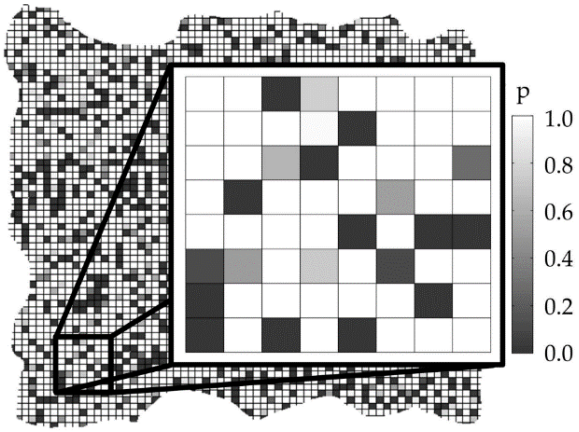

Fig. 2. A 64-bit fingerprint, shown within a larger fingerprint for context. The lightness of the shading of each cell indicates p, the probability of powering-up to 1 [8].

4. Zero Trust

Rose et al. [9] proposed a set of basic principles for Zero Trust Architecture (ZTA). ZTA treats all resources (such as data sources, computing services, and communication channels) as valuable assets that need protection. Access to these resources is granted on a per-session basis, with dynamic policies that consider various factors such as client identity, application/service identity, and behavioral attributes. Authentication and authorization of resources are strictly enforced, and all communication is secured regardless of network location. The enterprise monitors and measures the integrity and security posture of all owned and associated assets, collecting information about the current state of assets, network infrastructure, and communication channels to continuously improve its security posture. Fig. 3 shows a generic architecture model of the zero trust concept and one of its realizations.

Zero trust is based on the idea of sharing resources on a per-session basis, where privileges are minimally shared to perform services and minimize the impact of attacks. To achieve this, the system must have means of isolation and micro-segmentation, such as binding roles to the actors, granular access, provisioning control, key hierarchy and prohibiting lateral movement. In the next sections, we will discuss possible realizations of zero trust, especially in field devices, where resources are scarce.

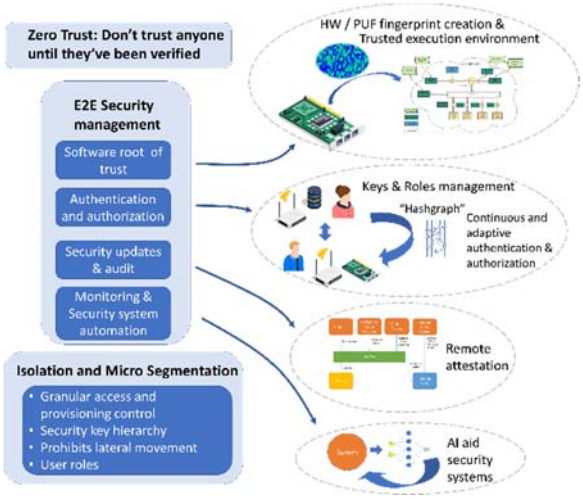

Fig. 3. Zero trust model.

4.1. Root of Trust

A Root of Trust (RoT) is a fundamental concept in computer security that refers to a reliable and secure origin of a computing system. The RoT must be inherently trusted, and it cannot be validated. The RoT can be used to establish a chain of trust that ensures the integrity, authenticity, and confidentiality of system components and applications. Based on the RoT and device certificates the actors can verify the security posture of each other using remote attestation. The actors can also authenticate and authorize to share resources and services. In evolving infrastructure, to keep the system secure, there is a need for a secure update mechanism and system audits monitoring the system's health. A simple RoT can be an immutable code block (e.g., boot ROM implementing secure boot) and a device-specific secret. The code block should guard access to a device-specific secret that is then used to derive the root key of the system also known as a Hardware Unique Key (HUK).

The RoT can also lean on isolated and secure execution areas such as secure enclave, secure element, trusted execution environment (TEE), trusted platform module (TPM) or hardware secure module (HSM). Resource constrained IoT devices should be relatively cheap. Due to the cheapness requirement, additional hardware components like TPM or custom hardware-based security modules are not preferred. TEE-based approaches like ARM TrustZone are integrated to a CPU and are nowadays available also in micro-controller class devices [10]. There are also specific trusted computing concepts for micro-controllers, e.g., the DICE concept [11]. A set of minimal functionality that is required to support remote attestation is discussed in [12].

A device-specific key must be immutable and different in each device instance. It must be kept secret and be directly accessible only from the RoT code. Chip manufacturers have used specific one-time-programmable (OTP) memory areas called electronic fuses to implement a device-specific secret. Early versions were factory-programmable but now most systems are field-programmable. These techniques can raise the bar of attacks but physical attacks utilizing electron microscope are still possible and additional steps are needed in chip manufacturing [13].

A root of trust device-specific secret can also be created with the aid of a PUF concept, where the device secret is built using existing inherent unique physical properties such as small manufacturing variations in silicon chips, which are used to generate unique, unclonable identifiers that can be used to derive cryptographic keys. From a resource point of view, PUF alternatives, which exploit existing hardware, like SRAM (Static Random Access Memory) based, are interesting, because most of the field devices have some form of RAM. In RAM based PUFs each time the SRAM block powers on, the memory cells come up as either 1 or 0. The start-up values create a random and repeatable pattern, which is unique to each chip [14].

In our study, we used the root of trust realization setup called *asvin* [15], the E1 development board from OKDO. This development board has an LPC55S69xx microcontroller from NXP. The microcontroller contains onboard PUF using SRAM to form a unique repeatable pattern. This pattern is further added to an activation code, which is then turned into a digital fingerprint i.e. root of trust device secret, which provides foundation for security subsystems.

4.2. Authentication and Authorization

The ZT principles state that there is no inherent trust among the actors of a system (users, applications, services, and hardware). This means that every actor must have a provable identity for authentication and authorization between each other. To achieve a lightweight authentication protocol, the communication between field devices and authentication services should be minimal and the cryptographic algorithms should be energy efficient such as, based on symmetric encryption. For this purpose, we chose a lightweight authenticated encryption scheme (ASCON) to implement. The ASCON is NIST recently proposed to standardization based on open competition, where addition of security, the selection criterion was performance and flexibility in terms of speed, size and energy use [16, 17]. ASCON can perform authenticated symmetric encryption with associated data (AEAD) and hashing [18]. As the name implies, AEAD ensures the privacy, integrity, and authenticity of ciphertext data and the integrity of unencrypted associated data. AEAD implements a method where data is first encrypted and then an authentication tag, i.e. message authentication code (MAC), is calculated. Fig. 4 shows the procedure where: 1) The key encrypts messages, and a random number is added to ensure privacy. 2) Calculates an authentication tag (MAC) that ensures that both encrypted and unencrypted parts of a message are not tampered with. Because MAC is done after encryption it makes it possible to exclude all tampered messages before encryption.

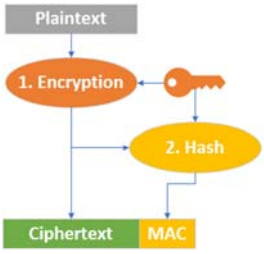

Fig. 4. Authenticated encryption with associated data (AEAD) in encrypt-then-MAC method.

4.3. Key and Role Management

As a part of authentication, we also study a distributed ledger-based approach, where field device identity information is typically stored in a blockchain. The selected ledger implementation depends on the use case. The ledger can be either private or public and either permissionless or permissioned, see Fig. 5.

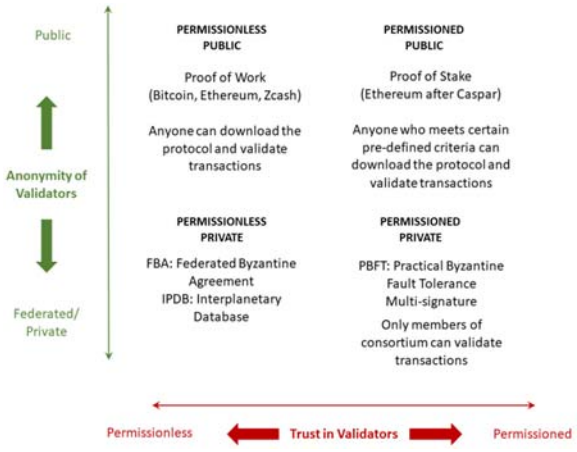

Fig. 5. Type of blockchain [19].

Concerning the nature of critical infrastructure, where access to a network is restricted and only permissioned actors are allowed, the best approach is a private and permissioned ledger. There are numerous consensus mechanisms like proof of work, proof of stake etc., which are popular in public networks but need to be excluded, because those mechanisms are energy and resource hungry [20]. However, there are still many consensus mechanisms based on Byzantine fault tolerance or other voting mechanisms that

perform better in terms of speed, energy consumption, and resource usage. One such example is Redundant Byzantine Fault Tolerance (RBFT) [21], where a preselected primary node creates a new block and other nodes follow only if the primary node is not malicious.

The performance of a consensus protocol has a significant impact on various aspects of the system, including fast transaction speed, quick recovery time from failures, scalability, low computation overhead, and energy consumption [22, 23]. Because of efficiency, we decided to study hashgraph-based approaches [24] as they provide fast consensus mechanism [25] operating even 120000 transactions per second (TPS) [26] with minimal communication overhead [24], and low energy consumption 3 mWh/transaction [27] due to minimal radio operations [28]. The hashgraph consensus mechanism is based on virtual voting, where only data of data (i.e., hashes) is shared between the nodes (i.e. gossip of gossiping). The data is shared using a flooding mechanism, where every node randomly sends its own information to neighbor nodes. Thus, during the consensus, there is a need to distribute two hashes (The size of the hashes is between 160 to 384 bits, depending on the security level) and timestamps (64 bits). This is a small amount compared to the actual data and the flooding mechanism provides fast consensus. However, if only hashes are distributed and the original data is only available in the source node, there must be other means to replicate data, e.g., using mirror nodes. Otherwise, the original data is lost if the source node is lost.

We propose to use node identities and database addresses, such as URL or IP addresses, as transaction data on the hashgraph, as shown in Fig. 6. By including the hash of the data in the database, we can verify its integrity. The data stored in the database can be specific to each node and sensor, and can be provided by different actors in the hashgraph community, such as fresh water companies, sensor manufacturers, sensor sellers, software companies, etc. For example, sensor manufacturers can provide calibration and SRAM-PUF data. The data can be either public (e.g., calibration data) or private (e.g., SRAM-PUF data), and can be accessed with appropriate methods.

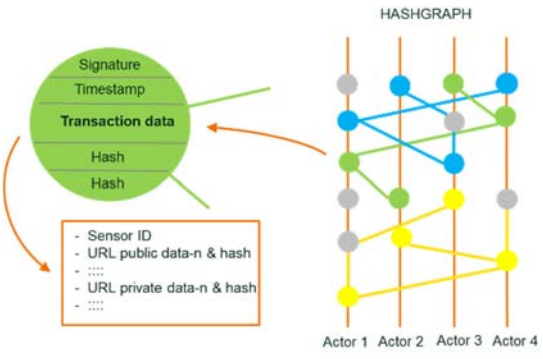

Fig. 6. Hashgraph and transaction data.

4.4. System Integrity Verification

When the system is up and running, the integrity of the field devices must be verified. Trustworthy way requires that the field devices have hardware-based root of trust that is protecting a device secret. In our case, the device secret is based on SRAM-PUF. The root-of-trust is based on ARMv8-M Trusted Firmware-M (TF-M) security architecture that is an implementation of ARM Platform Security Architecture (PSA) [29]. The security architecture provides isolation between secure and non-secure environments and allows secure boot implementation. Based on these building blocks, remote parties can request integrity verification of the system using a remote attestation protocol.

The Internet Engineering Task Force (IETF) RATS working group has specified an architecture for remote attestation that is specifying actors and data flows between the actors [30]. The architecture data flow is presented in Fig. 7.

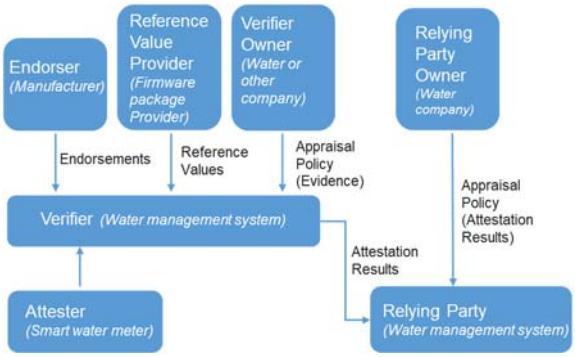

Fig. 7. IETF Remote ATtestation ProcedureS (RATS) Architecture conceptual data flow.

In the context of water management, the attester actor can be a smart water meter that needs to provide a proof of its integrity to the water management system that acts as the relying party actor of the architecture. The endorser actor can be a manufacturer of the smart water meter that could, e.g., issue a device certificate for the smart water meter. The attester should provide signed integrity measurements of the system components, e.g., cryptographic hashes. These cryptographic hashes can be compared to known good reference values provided by the reference value provider. For example, a firmware package provider could provide these reference values. The verifier actor is verifying the attestation evidence and then enforcing the policy (also known as appraisal) set by the verifier owner, e.g., omitting verification of some components, and then passing the attestation result to the relying party (now the water management system). The water management system can then act according to the appraisal policy, e.g., logging attestation errors or blocking access from components whose verification has failed.

We propose that distributed ledger technology can be exploited for remote attestation systems in context

of water management (see Fig. 8). The figure present W3C (Word Wide Web Consortium) model of digital verification architecture [31], which is fit to remote attestation verification model, where the water meter act as Holder, verifier owner and manufacture of meter as Issuers, water management system as Verifier, reference value provider located in Verifiable Data Registry implemented with distributed ledger technology. The functionality of reference value providers can be automated by using smart contracts, where verification is made.

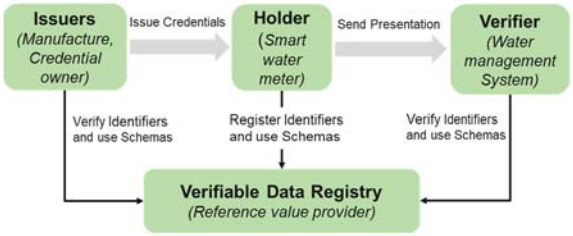

Fig. 8. Remote attestation model (RATAS) fit to W3C's Verifiable Credentials Data Mode.

5.3. Secure Software Updates

Software updates are either functionality or security updates. There is a need to either deploy new functionality or fix problems in old functionality. Security updates are fixing detected vulnerabilities. Update may be either field update or over-the-air (OTA) update. Especially OTA updates should be optimized by size. Update packages must be signed, and packages should be verified before installation. Packages should have a version number and rollback prevention should prevent installation of old vulnerable versions to already updated devices. Trusted Firmware-M includes Firmware Update (FWU) component that allows testing the installed update and either rolling back or accepting the update. Software update requires additional storage as the system must be able to store the original and the updated system image.

5. Discussion

5.1. Security and Zero Trust Concept

In Future water management system with an untrusted cloud environment, the attack surface will be much larger than in conventional water management systems having only some remote connections. The development and digitalization will lead to cloud-based systems, where clear barriered areas with firewalls cannot be defined. In that kind of environment, a zero trust approach will enrich methods to create sufficient protection and improve management of the system, especially under attacks and even attacks coming from inside of the system. One key point to build up security based on zero trust is that every actor, like users, devices, services, and programs, need to authenticate before getting rights to operate and use the resources. To do authentication actors will need a robust root of trust. Especially, field devices with a few energy resources will need energy efficient and reliable means to have root of trust like SRAM-PUF fingerprint.

SRAM-PUFs are a hardware security primitive that generate unique responses based on the physical variations in SRAM cells. These responses are not stored as conventional software-based cryptographic keys, making them resistant to software attacks that exploit vulnerabilities or weaknesses in software programs. However, SRAM-PUFs are not immune to side channel attacks, which are attacks that exploit the physical characteristics or behavior of the hardware. For example, cloning attacks [32, 33] can copy the SRAM-PUF responses by measuring the power consumption or electromagnetic radiation of the device. Data remanence attacks [34] can recover the SRAM-PUF responses by reading the residual charge left in the SRAM cells after power-off.

Data remanence is the phenomenon of data persisting in some physical form even after it has been erased. This can depend on the temperature, as data can remain for about 1.5 hours at 75 °C, 3 days at 50 °C, almost two months at 20 °C, and about 3 years at 0 °C [35, 36]. A data remanence attack is a type of fault injection attack that tries to recover data by introducing faults into the target system. They have shown that the data remanence effect can also be used to inject faults by changing the temperature of the SRAM [37].

The goal of cloning attacks is to clone the original SRAM and replace it with a fake SRAM that acts like the real one even if the attacker controls it. One way of cloning attack is to exploit a complementary metal-oxide-semiconductor aging mechanism, called Bias Temperature Instability (BTI), to alter the original start-up values of an SRAM-PUF and create physical clone [32]. Another method is to use a focused ion beam circuit to measure SRAM-PUFs and make a physical clone of SRAM. To measure the SRAM, they captured the near infrared (NIR) photonic emissions of SRAM and used them to make a clone.

Both types of side-channel attacks require deep knowledge of the specific hardware and often require laboratory equipment and settings, making them difficult to execute in the field. In our implementation, if the device is turned off and on again, the new fingerprint is also generated and the old one is discarded, which requires a new authentication. This will effectively prevent side-channel attacks that reboot the system, such as cloning attacks, and make it hard to inject faults that cause system rebooting and require new authentication. However, if the attacker can easily power off and on the field devices and the reauthentication requires administrative support, this will open up a new vulnerability in the system.

5.2. SRAM Energy Efficiency

SRAM stands for static because it does not need constant power bursts like DRAM (dynamic random-access memory) to keep the data stored. It can retain data as long as a small steady current is supplied, which reduces static energy dissipation to less than one to tens of nanowatts (nW) [37, 38]. The main source of static energy dissipation is leakage current, which causes continuous power consumption in the idle state. However, SRAM consumes more energy in the active state, when it is accessed for reading or writing, ranging hundreds of microwatts (μW) [39]. The amount of energy consumed during a read or write operation depends on factors such as the size of the memory cell, the operating voltage and temperature, and the access time. If SRAM-based caches are used for fast data access, they need to be kept in standby mode, where they may be partially active, consuming a small amount of power to maintain data integrity and respond quickly to requests.

5.3. Hashgraph

To avoid a single point of failure, the system should be decentralized. One way to achieve secure distribution is to use distributed ledger technology, which includes blockchain and hashgraph technologies. From a security perspective, the distributed ledger technology provides a verified data registry, which has data integrity assurance and support for different consensus mechanisms as inherent properties. This ensures data consistency even if some actors are malicious. Several public and commercial ledgers (e.g., Bitcoin, Ethereum, Hedera) have cryptocurrencies, which are used to reward participants of the ledger block creation. However, from a private distributed ledger perspective, cryptocurrency is an extra functionality, which is not necessarily required. Nevertheless, if the platform service is bought from an external provider or if there is a need to create some incentive or exchange system among community members or online shops, a cryptocurrency can facilitate those functionalities.

6. Conclusions

The research paper proposes a security architecture for future water management systems based on zero trust and physical unclonable function (PUF) concepts. Water management is a critical infrastructure that needs to operate reliably and securely in various situations. Future water management systems will use cloud and edge services, remote connections, and heterogeneous devices, which will increase the complexity and vulnerability of the systems. PUF is a technology that generates unique and unclonable identifiers or keys based on the physical properties of a device or component. PUF can provide a local root of trust for resource-constrained devices, such as sensors, in the field. As the resource efficient and suitable sensor devices was used SRAM-PUF. Zero trust is a security paradigm that assumes no inherent trust between actors in a system, such as users, devices, services, and applications. Zero trust requires continuous authentication and authorization of resources, secure communication, and dynamic policies. The paper describes the elements of zero trust architecture and how they can be applied to water management systems. The paper discusses how to verify the integrity of field devices using remote attestation protocols and how to perform secure software updates using trusted firmware components. The paper also considers the challenges and limitations of these approaches.

Acknowledgements

We would like to thank Jukka Julku for his main contributions to the remote attestation implementation and Aada Illikainen for her work with the Hashgraph.

References

[1]. S. Renukappa, A. Kamunda, S. Suresh, Impact of COVID-19 on water sector projects and practices, *Utilities Policy*, Vol. 70, 2021, 101194.

[2]. N. Ntuli, A. Abu-Mahfouz, A simple security architecture for smart water management system, *Procedia Computer Science*, Vol. 83, 2016, pp. 1164-1169.

[3]. W. Xia, X. Chen, C. Song, A framework of blockchain technology in intelligent water management, *Frontiers in Environmental Science*, Vol. 10, 2022, pp. 1-12.

[4]. S. W. Lee, S. Sarp, D. J. Jeon, J. H. Kim, Smart water grid: the future water management platform, *Desalination and Water Treatment*, Vol. 55, 2015, pp. 339-346.

[5]. C. V. Martinez, Turning the tide: The digital future of water management, https://www.autodesk.com/design-make/articles/water-management

[6]. E. Bindler, P. Keaney, K. Martz, Securing Water Utilities with AWS, Amazon Cloud, https://aws.amazon.com/blogs/industries/new-cybersecurity-white-paper-securing-water-utilities-using-aws/

[7]. T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, R. J. Young, A PUF taxonomy, *Applied Physics Reviews*, Vol. 6, Issue , February 20191, 011303.

[8]. D. E. Holcomb, W. P. Burleson, K. Fu, Power-up SRAM state as an identifying fingerprint and source of true random numbers, *IEEE Transactions on Computers*, Vol. 58, 2008, pp. 1198-1210.

[9]. S. Rose, O. Borchert, S. Mitchell, S. Connelly, Zero Trust Architecture, Special Publication (NIST SP), *National Institute of Standards and Technology*, Gaithersburg, MD, 2020.

[10]. ARM, TrustZone for Cortex-M, ARM, https://www.arm.com/technologies/trustzone-for-cortex-m

[11]. DICE Layering Architecture, *Trusted Computing Group*, Jul. 2020.

[12]. A. Francillon, Q. Nguyen, K. B. Rasmussen, G. Tsudik, A minimalist approach to Remote Attestation, in *Proceedings of the Design, Automation & Test Europe Conference & Exhibition (DATE'14)*, 2014, pp. 1-6.

[13]. A. Shamsoshoara, A. Korenda, F. Afghah, S. Zeadally, A survey on physical unclonable function (PUF)-based security solutions for Internet of Things, *Computer Networks*, Vol. 183, 2020, 107593.

[14]. D. E. Holcomb, W. P. Burleson, K. Fu, Initial SRAM state as a fingerprint and source of true random numbers for RFID tags, in *Proceedings of the Conference on RFID Security*, Vol. 7, 2007, p. 01.

[15]. Asvin, Physically Unclonable Function (PUF)–Setup, https://asvin.io/physically-unclonable-function-setup/

[16]. NIST Selects 'Lightweight Cryptography' Algorithms to Protect Small Devices, *National Institute of Standards and Technology*, 2023.

[17]. I. Elsadek, S. Aftabjahani, D. Gardner, E. MacLean, J. R. Wallrabenstein, E. Y. Tawfik, Hardware and energy efficiency evaluation of NIST lightweight cryptography standardization finalists, in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'22)*, 2022, pp. 133-137.

[18]. M. S. Turan, K. McKay, D. Chang, J. Kang, N. Waller, J. M. Kelsey, L. E. Bassham, D. Hong, Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process, NIST Interagency/Internal Report (NISTIR), *National Institute of Standards and Technology*, Gaithersburg, MD, 2023.

[19]. B. Varghese, M. Villari, O. Rana, P. James, T. Shah, M. Fazio, R. Ranjan, Realizing edge marketplaces: challenges and opportunities, *IEEE Cloud Computing*, Vol. 5, 2018, pp. 9-20.

[20]. J. Sedlmeir, H. U. Buhl, G. Fridgen, R. Keller, The energy consumption of blockchain technology: Beyond myth, *Business & Information Systems Engineering*, Vol. 62, 2020, pp. 599-608.

[21]. P.-L. Aublin, S. B. Mokhtar, V. Quéma, RBFT: Redundant byzantine fault tolerance, in *Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems*, 2013, pp. 297-306.

[22]. S. M. H. Bamakan, A. Motavali, A. B. Bondarti, A survey of blockchain consensus algorithms performance evaluation criteria, *Expert Systems with Applications*, Vol. 154, 2020, 113385.

[23]. J. Zhang, Y. Rong, J. Cao, C. Rong, J. Bian, W. Wu, DBFT: A Byzantine fault tolerance protocol with graceful performance degradation, *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, 2021, pp. 3387-3400.

[24]. L. Baird, The SWIRLDS hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance, Tech Reports SWIRLDS-TR-2016-01, *SWIRLDS*, Vol. 34, 2016, pp. 9-11.

[25]. L. Baird, A. Luykx, The hashgraph protocol: efficient asynchronous BFT for high-throughput distributed ledgers, in *Proceedings of the International Conference on Omni-layer Intelligent Systems (COINS'20)*, 2020, pp. 1-7.

[26]. N. Gao, R. Huo, S. Wang, T. Huang, Y. Liu, Sharding-hashgraph: a high-performance blockchain-based framework for industrial internet of things with hashgraph mechanism, *IEEE Internet of Things Journal*, Vol. 9, 2022, pp. 17070-17079.

[27]. J. I. Ibañez, F. Rua, The energy consumption of Proof-of-Stake systems: A replication and expansion, *arXiv Preprint*, 2023, arXiv:2302.00627.

[28]. P. Koskela, Energy-efficient solutions for wireless sensor networks, PhD Thesis, *Acta Universitatis Ouluensis*, 2018.

[29]. T. Martin, Designing Secure IoT Devices with the Arm Platform Security Architecture and Cortex-M33, *Elsevier Science*, Oxford, 2022.

[30]. H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan, Remote ATtestation procedureS (RATS) Architecture, IETF RFC 9334, https://www.rfc-editor.org/info/rfc9334

[31]. M. Sporny, D. Longley, D. Chadwick, Verifiable Credentials Data Model v1.1, W3C Recommendation, 03 March 2022, https://www.w3.org/TR/2022/REC-vc-data-model-20220303

[32]. S. Duan, G. Sai, BTI aging-based physical cloning attack on SRAM PUF and the countermeasure, *Analog Integrated Circuits and Signal Processing*, Vol. 117, Issue 1-3, Dec 2023, pp. 45-55.

[33]. C. Helfmeier, C. Boit, D. Nedospasov, J.-P. Seifert, Cloning physically unclonable functions, in *Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'13)*, 2013, pp. 1-6.

[34]. C. Yehoshuva, R. Raja Adhithan, N. Nalla Anandakumar, A survey of security attacks on silicon based weak PUF architectures, in *Proceedings of the 8th International Symposium on Security in Computing and Communications (SSCC'20)*, Chennai, India, 2020, pp. 107-122.

[35]. P. Gutmann, Data remanence in semiconductor devices, in *Proceedings of the 10th USENIX Security Symposium (USENIX Security'01)*, 2001, pp. 39-54.

[36]. N. A. Anagnostopoulos, T. Arul, M. Rosenstihl, A. Schaller, S. Gabmeyer, S. Katzenbeisser, Attacking SRAM PUFs using very-low-temperature data remanence, *Microprocessors and Microsystems*, Vol. 71, 2019, 102864.

[37]. G. Prasad, A. Anand, Statistical analysis of low-power SRAM cell structure, *Analog Integrated Circuits and Signal Processing*, Vol. 82, 2015, pp. 349-358.

[38]. D. Anitha, M. M. Ahmad, Ultra-low leakage static random access memory design, *International Journal of Reconfigurable and Embedded Systems*, Vol. 12, 2023, pp. 60-69.

[39]. P. Sandeep, P. A. H. Vardhini, V. Prakasam, SRAM utilization and power consumption analysis for low power applications, in *Proceedings of the International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT'20)*, 2020, pp. 227-231.