Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 1164 – 1169

Procedia Computer Science

The 11th International Symposium on Intelligent Techniques for Ad hoc and Wireless Sensor Networks, (IST-AWSN 2016)

A Simple Security Architecture for Smart Water Management System

Nonhlanhla Ntuli*, Adnan Abu-Mahfouz

Meraka Institute, Council for Scientific and Industrial Research (CSIR), Pretoria, 0001, South Africa

Abstract

Water scarcity and water stress issues have become clear threat to the global population. This makes water management a critical aspect to ensure sustainable water. An efficient water management system requires thousands of constraint devices (sensors and/or actuators) to be deployed across the water distribution network to enable near-real time monitoring and control of the water grid components. Security at both the device and network level is critical to the operation of such a system. Although several IT security controls have developed over the past few decades, they cannot be used directly with such constraint devices. This is due to their limited resources and unique requirements. However, some of these techniques can be adapted and used with these constraint devices. In this paper, we propose security architecture for smart water management systems, the architecture leverages existing security solutions and design patterns

Peer-review under responsibility of the Conference Program Chairs

Keywords: smart water management system, security architecture, access token, smart water, iot security;

Nomenclature

AAC	Authorization and Access Control
MQTT	MQ Telemetry Transport
CoAP	Constrained Application Protocol

***** * Corresponding author. Tel.: +27-12-841-4760; fax: +27-12-841-4720.
E-mail address: nhlanhlayami@gmail.com

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.239

1. Introduction

We live in a technological world where Information and Communication Technology (ICT) has become crucial for sustainable development. Water management[1] is one of the areas where ICT plays a key role in addressing various associated challenges such leakage detection and dynamic optimization[2]. While addition of ICT to critical infrastructures enhances performance, it also leaves the infrastructure vulnerable to cyber-attacks.

Building an efficient smart water management system requires deployment of thousands of sensors and/or actuators to monitor and control water grid components such as water meter, pumps, and pressure sensors[3,4,5]. These devices are likely to be controlled over a wireless network, and if a hacker gains access to the network they can breach confidentiality and integrity. While there is a lot of research that has gone into addressing security over the past decade, many of the existing solutions are not suitable for water management systems due to unique requirements and constraints such as limited computing resources and over-the-air communications[6].

We address this problem from the design perspective; while some of the existing solutions can be re-used, design of the security architecture has to evolve to address the new requirements. In this paper, we adapt and combine some of the existing security techniques to propose a simple security architecture. We discuss threads and new design patterns in Section 2. In Section 3 we propose our solution and the paper concludes in Section 4.

2. Security Threads and Design Patterns

When it comes to intelligent water management system threats, we deal with more than just hacking into the network. One of the biggest concerns is physical attacks that can be launched on a device. Most field devices are portable, making capturing easy. After capturing the device the attacker can clone the device, install new firmware or learn sensitive information stored in the device[7]. Unlike conventional networks that are physically protected, devices in the field are accessible to anyone. And it is easy to pinpoint their location by tracking signals[8].

For efficient functioning of the infrastructure, in addition to physical attacks, we have to protect against attacks that do not require the hacker to be part of the network. This class of attacks includes radio jamming to prevent communication between legitimate devices and eavesdropping using sniffing tools[5,9]. Attacks that hinder communication between devices, affect the real-time dynamic intelligent part of the system such as the supervisory control. These real-time components are also affected when the sensor/meter data is tempered with, resulting in erroneous optimization[4,10].

Finally, updating the firmware in devices that are already deployed should also be secure. Because updates can only be transferred over the air, attackers can use this to their advantage and install rogue updates[8]. What's even more difficult for a new device is to establish the root of trust; which other devices to trust? Furthermore, the new device has to prove to other devices that it is legitimate and trustworthy. If this step is not secure, any security steps that follow would not be useful.

The unique threats call for a shift in the way we design security architectures. We consider new design patterns:

- Publish / subscribe communication model. One device may play many roles; it can be a producer collecting and publishing water reading and/or a consumer processing data published by a producer. Establishing and maintaining connections with each and every other peer device would be heavy for a constrained device. A publish/subscribe model introduces a broker, which is a capable server that would be responsible for pushing published data to consumers. This model also eliminates the need for edge devices to have vulnerable open inbound ports[11]. Standardized, simple and lightweight

communication protocols like Constrained Application protocol (COAP) and MQ Telemetry Transport (MQTT) already support publish / subscribe model and they have been proved to be robust[12].

- Offload security from edge devices (sensors/actuators) to trusted gateways. Because of constrained resources, existing security solutions are too heavy for edge devices. These solutions need to be adapted to offload heavy operations to trusted gateways. The gateways should be responsible for creating, updating and distribution of certificates and access secrets[13]. This allows edge devices to focus on reporting about their surroundings and/or control water grid components.
- Stateless Authentication. The less we do at the edge, the longer the life span of the devices. Traditional server-based authentication methods would not scale in smart water systems. For that reason, many authors have proposed token-based authentication[11,12,13]. The merits include stateless in nature and the ability to work across domains. Since the token itself is enough to authenticate a device, this method eliminates the need to store session data.
- Block-chain technology. Bitcoin is a digital currency that relies on block-chain technology, which is a ledger of all legitimate transactions that have occurred on a network[14,15]. The ledger is maintained by all users in the network. It can be used for a range of other non-monetary transactions, including creation and exchange tokens and to track/verify other kinds of digital exchanges[16]. For example, Name-coin block-chain records who owns which name in a namespace, and hence to track the history of devices in network, providing a trustless and decentralized network.

These design approaches can assist mitigate a lot of existing threats. In the following section, we propose a simple security architecture that incorporates these design patterns.

3. The Security Architecture

In this section we propose a simple security architecture, shown in Fig 1., which leverages the design concepts discussed in section 2. We assume that publish-subscribe messaging pattern is used and that publishers and consumers only interact with an MQTT broker like Mosquiitto, and not directly with each other. We introduce Authorization and Access Control Servers (AAC) into our architecture as shown in Fig 1. The role of the AAC is to authenticate devices and provide them with access tokens. These access tokens can be used later for stateless authentication. For simplification purposes, the broker/gateway and AAC are shown as separate components.

Publishers that wish to join the network firstly need to connect to an AAC server pre-configured before deployment. Using block-chain technology the device creates a unique ID and stores it in its crypto-chip. The device sends its identity to the AAC server. The AAC verifies the identity and authenticates the new device. If authentication is successful, the AAC server creates a new access token with the ID of the device and expiry date. The token is signed and transmitted to the device. After authenticating itself with the AAC server, the device can publish or read updates depending on the type of access granted.

To allow the broker to authenticate an update, the publisher must append the write access token with every update request. Likewise, when a consumer wishes to receive updates, the consumer must first obtain a read access token from the AAC server. The access token will be used to subscribe to updates through the broker. The AAC server is the only fully trusted entity in the network. It manages both publishers and consumers, validates their identities, issues access tokens, handles caching and other security operations. By moving these responsibilities to the AAC server, the edge-devices can save energy by focusing on lightweight tasks like forwarding observations.

Notice that there is no direct interaction between publishers and consumers. By decoupling publishers and consumers, we allow flexibility and thus scalability. A publisher could be serving hundreds of consumers, it need not worry about them and whether data was received and if they have privileges to

receive that data. It also allows for smooth replacement of publishers should existing publishers die or be compromised.

We consider three security goals: secure booting, secure communication and secure firmware updates. By addressing all these goals we can improve end-to-end security.

3.1. Secure Booting

The first step towards device security is ensuring secure booting. Secure booting prevents installation of malicious code onto the device. By making sure that the booting process is secured, we can establish securely the root of trust for the device. Public key cryptography is utilized at this stage. During manufacturing, the public key to be used for verification is pre-loaded into the non-writable storage on the device. At power-on the device uses the pre-loaded key to verify the code, to ensure it has been authorized, before running it.

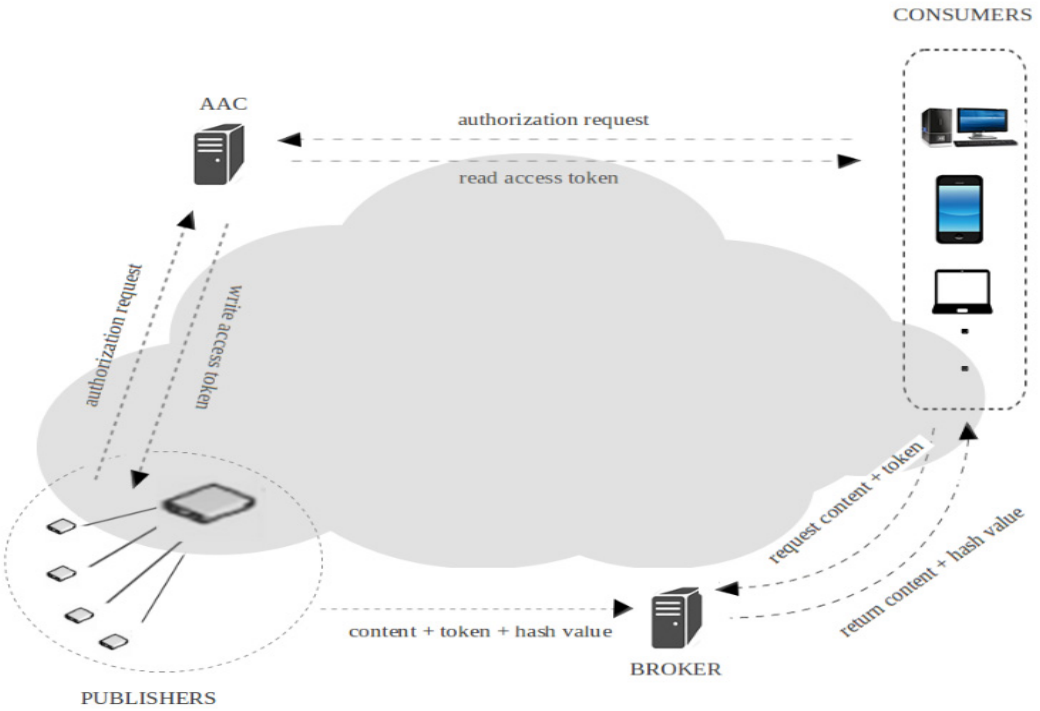

Fig. 1. Security Architecture

3.2. Secure Communication

While public key cryptography can be used in the first step (secure booting), it would be too heavy to use during normal operation. For that reason, we move away from using public key cryptography to secure communications. Instead, we rely on cryptographic hash functions and stateless access tokens. Before the producer publishes content, a cryptographic hash function is used to produce a hash value. This hash value is sent along with the original content and access token. When the consumer receives content, it uses the hash value and the access token to validate the identity, integrity and authenticity of the publisher / content.

3.3. Secure Firmware Updates

The other important security issue is to securely firmware update on already deployed devices. To secure firmware updates, we use public key cryptography and hash functions. When a device downloads updates from the network, it validates those updates by 1) calculating the hash value and comparing it 2) Check the signature to ensure the identity of the server. These steps verify both integrity and authenticity. There are two ways in which the device can learn about the public key of the update server. The first option would be to pre-load the key during manufacturing, this option is limited, but would be enough for some environments. The second option would be for the AAC to give this information to the new device during initial authentication together with access tokens.

4. Conclusion

In this paper, we proposed simple security architecture for smart water management systems to ensure secure booting, secure communications and secure firmware updates. It is based on new design patterns such as publish-subscribe and stateless authentication. By coupling cryptographic hash functions and stateless access tokens, we can efficiently validate the identity of the device and the integrity/ authenticity of the content. Trusted entities were introduced to handle heavy security tasks such as validation and authentication of devices, issuing of access tokens and so on. We also introduced brokers to manage publishers and consumers, further moving heavy operations away from edge devices. The proposed system will be implemented and tested using WSN testbed[17,18]. Extra security features and techniques will be added and implemented in the final smart water management system[2].

References

1. Bello O, Piller O, Abu-Mahfouz AM, Hamam Y. Management problems and their approaches in water distribution networks. Submitted for publication, 2016
2. Abu-Mahfouz AM, Hamam Y, Page PR, Djouani K, Kurien A. Real-time dynamic hydraulic model for potable water loss reduction. Submitted for publication, 2016
3. Mudumbe M, Abu-Mahfouz AM. Smart Water Meter System for User-Centric Consumption Measurement. In: Proceedings of the IEEE International Conference on Industrial Informatics, Cambridge, UK 2015; p993–998
4. Seung Won Lee, Sarper Sarp, Dong Jin Jeon, & Joon Ha Kim. Smart water grid: the future water management platform. Desalination and Water Treatment Volume 55, Issue 2, 2015; 339-346.
5. Xiaohui Liang, Xu Li, Rongxing Lu, Xuemin Shen, Xiaodong Lin, & Haojin Zhu. Securing smart grid: cyber attacks, countermeasures, and challenges. IEEE Communications Magazine, 2012;, 8-45
6. Louw J, Niezen G, Ramotsoela TD, Abu-Mahfouz AM. A Key Distribution Scheme using Elliptic Curve Cryptography in Wireless Sensor Networks, Submitted for publication, 2016
7. Baker, D. Making a Secure Smart Grid a Reality. In: Ensec.org; 2009
8. Burns P. A Simple Proposal To Improve Security for the Internet of Things. Retrieved October 20, 2015, from Linkedin: https://www.linkedin.com/pulse/simple-proposal-improve-security-internet-things-pat-burns

9. Hyunwoo Lim, Jongbin Ko, Seokjun Lee, Jongwan Kim, Mijoo Kim, & Taeshik Shon. Security Architecture Model for Smart Grid Communication Systems. In: Proceedings of the International Conference IT Convergence and Security (ICITCS) 2013;p 16-18
10. Zubair BA. Abdul-Raoof A. An Analysis of Smart Grid Attacks and Countermeasures. Journal of Communications 2013; 8.8, 473-479
11. O'Connor, M. C. (n.d.). A New Approach to IoT Security. Retrieved September 26, 2015, from Pubnub.com: https://www.pubnub.com/static/papers/IoT_Security_Whitepaper_Final.pdf
12. Kothmayr T, Schmitt C, Wen Hu, Brunig M, Carle H. A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication. In: Proceedings of the IEEE 37th Conference on Local Computer Networks (LCN Workshops), 2012, 956-963, 22-25
13. Mališa V, Bernard T, Franck R, Andrzej D, Laurent D, Roberto G. OSCAR: Object security architecture for the Internet of Things. Ad Hoc Networks Internet of Things security and privacy: design methods and optimization 2014; p3-1679.
14. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. Consulted 2008; (p. 28
15. Blockchains and the Internet Of Things. (n.d.). Retrieved from Postscapes.com: http://postscapes.com/blockchains-and-the-internet-of-things
16. O'Connor, M. C. (2015, August 05). Three Approaches to IoT Security. Retrieved September 28, 2015, from Iotjournal.org: http://www.iotjournal.com/articles/view?13351
17. Abu-Mahfouz AM, Steyn LP, Isaac SJ, Hancke GP. Multilevel Infrastructure of Interconnected Testbeds of Large Scale Wireless Sensor Network (MI2T-WSN). In: Proceedings of the International Conference on Wireless Networks ICWN '12, 16-19 July, Las Vegas, Nevada, USA, 2012, p 445-450.
18. Dludla AG, Abu-Mahfouz AM, Kruger CP, Isaac JS. Wireless sensor networks testbed: ASNTbed. In: Proceedings of the IEEE IST-Africa Conference and Exhibition (IST-Africa), 29-31 May, Nairobi, Kenya, 2013, p1-10