Secure Provisioning for Achieving End-to-End Secure Communications

Patrícia R. Sousa¹[0000-0002-0268-9134]</sup>, João S. Resende¹[0000-0003-0125-4240]</sup>, Rolando Martins¹[0000-0002-1838-1417]</sup>, and Luís Antunes¹[0000-0002-9988-594X]</sup>

DCC-FCUP/CRACS-INESC TEC {patricia.sousa,jresende,rmartins,lfa}@fc.up.pt

Abstract. The growth of the Internet of Things (IoT) is raising significant impact in several contexts, e.g., in cities, at home, and even attached to the human body. This digital transformation is happening at a high pace and causing a great impact in our daily lives, namely in our attempt to make cities smarter in an attempt to increase their efficiency while reducing costs and increasing safety. However, this effort is being supported by the massive deployment of sensors throughout cities worldwide, leading to increase concerns regarding security and privacy. While some of these issues have already been tackled, device authentication remains without a viable solution, specially when considering a resilient decentralized approach that is the most suitable for this scenario, as it avoids some issues related to centralization, e.g., censorship and data leakage or profit from corporations. The provisioning is usually an arduous task that encompasses device configuration, including identity and key provisioning. Given the potential large number of devices, this process must be scalable and semi-autonomous, at least. This work presents a novel approach for provisioning IoT devices that adopts an architecture where other device acts as a manager that represents a CA, allowing it to be switched on/off during the provisioning phase to reduce single point of failure (SPOF) problems. Our solution combines One Time Password (OTP) on a secure token and cryptographic algorithms on a hybrid authentication system.

The final authenticated version is available online at https://doi.org/10.1007/978-3-030-31831-4_34

1 Introduction

Internet of Things (IoT) is an umbrella concept detailing how technology will interact with users in the coming years. The highly paced technological development surrounding it is exposing several novel challenges, namely on privacy and security. In particular, there is a need to adopt secure solutions for IoT devices characteristics due to their intrinsic limitations [1] (e.g., battery life and memory space). Among the set of security and privacy requirements necessary to securely

support IoT, we highlight the user and device identity management, authentication, the confidentiality of data exchanged in communications, network access control to allow only authorized devices, and the availability of resources and systems [2].

Most past systems base their identity and authentication through the use of Public Key Infrastructure (PKI) [3]. However, there are known limitations of this technology as it depends on a centralized Certification Authority (CA), which once compromised, the entire system is compromised as well. On top of this, many security and privacy issues are caused by human configuration errors [19], and to solve them, we need systems with better interfaces to users and better tools to help with the provision of new devices.

Nowadays, PKI-based solutions are still unable to provide security by default, as systems rely on user-provided security through manual device provisioning configurations. For this reason, the paper presents a decentralized secure device-to-device communications solution in which device provisioning is focused on improving usability while providing security by default. The solution focuses on the use of a PKI where the CA is represented by a manager device that can be switched on/off to reduce single point of failure (SPOF) problems. Our solution combines public key cryptography and symmetric keys with the One Time Password (OTP) concept using a secure token. Device identity is guaranteed by physical access to this physical token. In addition to generating an OTP, the physical token also stores a public key to be transmitted to target devices only, eliminating attacks such as impersonation or man-in-the-middle. It also improves usability as we exclude configuration errors and difficulty choosing the right settings while provisioning the device. Although there is manual interaction to use the secure token, the process itself, is as simple as finding the device to be provisioned and plugging in the secure token.

Section 2 presents the related work with some systems related with our approach. Section 3 describes an overview of the proposed system and the characteristics of the system. The implementation details of the system is described in Section 4. Section 5 defines a threat model and an attack model, in order to see the vulnerabilities from the point of view of defender's and attacker's. Lastly, Section 6 presents the conclusions of this work and some future work.

2 Related Work

There are many applications that provide identity, authentication and authorization across multiple contexts.

PKI provides important core authentication technologies for IoT. A study by *Ponemon* [11] claims that, 42% of devices will continue to use digital certificates for authentication and identification in the next two years. The SSL/TLS [13] or *Kerberos* [12] are some examples of authentication systems based on a PKI.

Some of the current solutions have scalability limitations regarding their authentication protocols. For example, the author *Sousa et al.* [15] solves peer-

to-peer authentication in a decentralized way by using Short Authentication Strings, but it does not scale well for multiple devices (M-to-N authentication).

The following papers from the authors Antonio Solano, et al. [14] and Pascal Hirmer, et al. [16] are focused on the provisioning of IoT devices but are not focused on usability, because the authors propose manual solutions where all device data must be entered manually by the users.

There are several end-to-end solutions more related with OTP solutions. Some of them, are based on temporary passwords and/or unique numbers (OTPbased solutions [5,4,6], for example), as well as Physically Unclonable Functions (PUF)-based solutions [7,8]). Daniel Kelly et al. [5] claim that IoT devices with single factor authentication are not sufficient for secure communication. A solution presented by Shivraj et al. [4] creates a lightweight, robust and scalable OTP technique developed by using the principles of IBE-Elliptic Curves Cryptography (ECC) allowing two-factor authentication. The work done in [6] also has an authentication through OTP to the application level information security.

3 System Overview

This section describes all components of the architecture, as well as the different phases to achieve a decentralized secure end-to-end communications. The description includes the provisioning phase, authentication mechanisms and scalability extension.

3.1 Manager Setup Phase

The manager device represents the system CA that plays an essential role in a certification system by signing public keys (or certificates). This device should be assumed to be trusted and controlled only by trusted persons (such as the network owner). All certificates signed by the device will be implicitly trusted. Currently, systems that manage a PKI require a high degree of security and are installed on an isolated machine. In this proposed system, the PKI is installed on the manager device that is hybrid, meaning it may be offline from the network when not in use, to prevent the possibility of the private key being stolen in a possible network intrusion.

For added security, the manager device can use Intel SGX [22] in order to secure all the cryptographic assets in a Trusted Execution Environment (TEE).

The manager begins by setting a CA using 256-bit Elliptic Curve Digital Signature Algorithm (ECDSA) and a 256-bit Elliptic Curve Integrated Encryption Scheme (ECIES) key pair to generate a shared key without the need for Diffie-Hellman exchange. This option was chosen because the ECC is better for low resource devices as ECC requires fewer resources and provides the same level of security as Rivest-Shamir-Adlema (RSA) cryptography with a smaller key [17]. In brief, it is possible to use RSA also at this stage, but as the focus of the approach is on IoT, ECC was chosen.

The manager device also features an OTP manager for allowing authentication through the use of a secure token, i.e., "something that you have". It works like a key to a door in a house where the key holder has access to the house. For this reason, anyone who has access to this secure token can authenticate with the manager. This secure token is not limited to generating OTPs, as the idea is also to store the manager's public key for transmission to trusted devices.

3.2 Alice and Bob device authentication

The authentication between a new device and the manager is essential for ensuring that it is added to the trusted device pool. To do this, the owner inserts the secure token into the target device and then the new device is added to the pool (Alice and Bob device) (Figure 1).

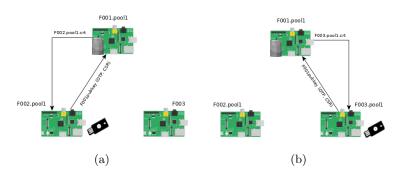


Fig. 1. Alice (a) and Bob (b) Device Authentication

As previously described, the secure token also has the manager's device public key. The new device starts by sending the Certificate Signing Request (CSR) of itself (which contains only the public key, not the private key, so the private key has not been compromised). When the new device sends the CSR to the manager, the later will produce a signed x509 Certificate. Furthermore, it also sends OTP to verify that the new device is in physical presence with the secure token and is therefore the correct device to authenticate. All this information is sent encrypted with the shared key (ECDHE) generated for both parties (client and manager) to encrypt a message that can only be decrypted by the manager. After the authentication is successful, the manager device sends back the signed certificate to certify that the client is a secure device that can be added to the trusted device pool. These certificates are used to establish trust between client devices and provide decentralized secure communication between them without the intervention of the manager device.

3.3 Unique Service Name

The manager needs to have a well-known service domain to offer to their users so that he can be discovered by them. At first, only the manager has his URL like identifier (Unique Service Name (USN)) along with the domain "pool1" chosen only for example purposes in this paper. USN is a unique identifier that serves to uniquely identify a specific service to allow identical services to be differentiated. Once a device is authenticated with the manager, it gains its domain, and therefore the USN is modified from "F002" to "F002.pool1", for example, to add the domain to which it now belongs. Thus, for example, when the device "F001.pool1" wants to communicate with the "F002.pool1", it can use a peer discovery protocol Simple Service Discovery Protocol (SSDP), to discover the device with this USN to authenticate with it.

3.4 Decentralized Secure End-to-End Communications

After the discovery process, devices need to authenticate with each other. For mutual trust, both devices must exchange manager's signed certificates. After verifying the authenticity of certificates, a symmetric key is generated between both devices to establish secure communication.

Symmetric key generation needs authentication so that the nodes know each other. ECDSA was chosen for signing and verification and ECIES was chosen for encryption. Then, Diffie-Hellman Ephemeral (DHE) or Elliptic-curve Diffie-Hellman Ephemeral (ECDHE) is used for key exchange. Ephemeral mode is important because if the key pair is used for more than a few hours, it must be stored somewhere because devices can be turned off. There is always some risk that a stored key pair may be compromised, although a wide variety of methods can be and are used to mitigate this issue. This mode avoids this type of attack by not storing key pairs and generating a new key pair every millisecond, thus ensuring Perfect Forward Secrecy.

After the establishment of a shared secret using ECDHE, the devices can exchange data with symmetric encryption using the secure cipher AES256 to encrypt messages.

When all devices are provisioned, the manager can be turned off until a new device needs to be added to the pool or there is a change on the device pool, such as certificate revocation or renewal.

3.5 Merge Two Trusted Devices Pools

An identity and authentication system must be flexible and highly scalable enough to handle billions of device infrastructures in multiple environments such as smart home and smart cities in general. This system must support different types of environments, given the heterogeneity of applicability that exists in IoT scenarios.

For greater scalability, there needs to be a usable way to integrate different device pools to make the system more practical as it would not be feasible to

re-provision devices already provisioned with another manager, so that devices from different pools can communicate with each other.

To address this issue, the system replicates the traditional mechanisms of having multiple CAs supported by a client. It is essential to ensure two points to deploy this in a real world configuration: Use a secure token authentication scheme to enable enrollment and trust between different managers; and information dissemination on new pools among all new devices.

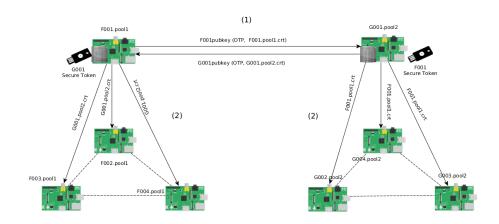


Fig. 2. Trust between two device pools

To allow two pools to connect to each other, the authentication between managers (Figure 2 in 1) uses the same mechanism described in (Section 3.1). After both managers perform mutual authentication, the next step is to spread the information across devices among different pools. To do this, the manager must send the signed and encrypted information to the devices. This allows any of the devices to read the information and verify the manager's signature. Figure 2 represents the agreement between both managers and the corresponding spread of information from managers to their peers when they begin to trust each other and can announce on the network that others should move to include these new trusted colleagues in their trusted network.

4 Implementation

In order to implement the system, it is used *Raspberry PI*'s to represent all devices and the secure token is represented by the Yubikey NEO and USB pen drive. The OTP generator is the Yubikey NEO and the USB pen drive replicates the extra storage that contains the public key.

The secure token needs to be configured with the manager's public key and with an OTP server. The manager's public key is generated through ECDSA based on a GitHub repository [21]. The OTP Server is represented by *privacyIDEA* [9] that is a modular authentication system. To enroll the secure token on *privacyIDEA*, it is used a test account and the "Enroll Token" in the "Yubico AES mode: One Time Passwords with Yubikey" option. This system allows the administrator to revoke or disable registered tokens so that there is a security guarantee in case of stealing or losing tokens.

After the enrollment, it is possible to authenticate with the secure token. The exchange of OTP and public key to authenticate is done through ECIES. The implementation is based on a Python implementation - *Elliptic Curve Integrated Encryption Scheme (ECIES) with Rabbit* [18].

For the device-to-device authentication, it is implemented the ECDHE algorithm.

5 Security Analysis

In this section, it is defined a threat and an attack model with a security analysis of the proposal, exploring the vulnerabilities from the defender's and attacker's point of view.

5.1 Threat Model

From the defender's perspective, this section identifies system (assets) and potential threats against the proposed system.

Physical devices

The device manager acts as a dynamic validation point in the device provisioning phase. Devices must be reliable to ensure that they are not broken or stolen. The network owner is supposed to avoid exposing the device. In addition, the fact that it is not an Internet-wide exposed CA and that it is possible to turn off the manager when there are no new peers to join the network, makes it less exposed to potential attacks eliminating the SPOF.

If the secure token is used as a first factor authentication, it means that all accounts configured with this token are vulnerable to theft of that physical token. To mitigate this situation, it is possible to revoke the secure token on the OTP manager if it is lost and with an expiration time.

Cloning a device is a problem for current IoT devices. When there is a physical access to any device, security can be compromised. Current research challenges are focusing on deploying trust zones (such as Intel SGX), where parts of the code can safely compute secrets and store authentication keys and device identity. Future work will focus on this to mitigate and ensure device integrity.

On the other hand, using secure token helps protect against hacking as physical access to the secure token is required to generate OTP.

Unique Service Name

A USN is theoretically easily changeable and it is possible for an attacker to

register a fake name to impersonate another device. However, in this proposal, the attacker must have a certificate signed by the manager to ensure that the USN belongs to him, in addition to changing the USN, to be able to impersonate another device.

Configurations for mitigation of attacks

There is an option in *privacyIDEA* which is the Maximum Fail Counter to avoid brute force attacks when an authentication request occurs. If the fail-counter exceeds this number, the token cannot be used unless the fail-counter is reset. This system has a fail counter of 5 and this option must be manually reset to prevent this type of Brute Force attack.

5.2 Attack scenarios

This section identifies attack modeling from an attacker's point of view to analyze how the system could be exploited in terms of vulnerabilities in IoT solutions.

Tag impersonation attacks are safe because the secure token generates OTPs and also stores the manager's public key for transmission to trusted devices. This prevents the manager from being impersonated even if the attacker could change his name. An attacker must have their data (authenticated OTP and its public key) in the physical secure token, which is impractical.

Also, the system is safe from replay attacks because the secure token uses a set of volatile and non-volatile counters that ensures that an OTP can no longer be used after validating once [10].

For man-in-the-middle attacks, there are two options: unknown peers and peers that are already authenticated. The first issue is resolved because this approach uses a secure token with the receiver's public key, so it is transmitted only over USB. This mitigates the unknown attacker's access to the public key. For authenticated peers, as already described in the replay attack, the problem is solved using an OTP that can no longer be used after validating once [10].

6 Conclusions and Future Work

This work presents a design and implementation of a new approach to IoT device provisioning, giving an identity to a *thing*, eliminating the risk of impersonating attacks and allowing devices to be authenticate. In addition, in terms of security, the single point of failure problem is mitigated with a hybrid solution to allow the manager device to shut down when not needed.

Device provisioning is based on a secure token proposed in this paper. In addition to generating an OTP, this secure token stores a public key to be transmitted only to the target devices, to improve security.

Although using a secure token requires a USB port, it can be adapted to other technology such as RFID or similar. However, given the study by authors Singh, Kiran Jot, and Divneet Singh Kapoor [20], there are USB ports on most devices that are used in IoT.

This work proves the feasibility of the solution by presenting a wide range of options needed in order to be deployed in real-world scenarios.

In future work, we plan to deploy this architecture on a Porto street to fully understand the impact on IoT devices in a smart city deployment. It is necessary to analyze the energy impact of the solution on low resource devices and measure authentication latency between devices. Revocation and information dissemination protocols should be developed and tested in a real environment to analyze the associated capacity and delay.

Acknowledgment

This work of Patrícia R. Sousa and João S. Resende was supported by Fundação para a Ciência e Tecnologia (FCT), Portugal

(SFRH/BD/135696/2018, PD/BD/128149/2016).

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within projects: UID/EEA/50014/2019 and CMU/CS/0042/2017.

This work has been supported by the EU H2020-SU-ICT-03-2018 Project No. 830929 CyberSec4Europe (cybersec4europe.eu).

References

- 1. Yang, Yuchen, et al. "A Survey on Security and Privacy Issues in Internet-of-Things." IEEE Internet of Things Journal (2017).
- H. Sundmaeker, P. Guillemin, P. Friess and S. Woelffle, "Vision and Challenges for Realising the Internet of Things," Cluster of European Research Projects on the Internet of Things, 2010.
- 3. Kuhn, D. Richard, et al. Introduction to public key technology and the federal PKI infrastructure. National Inst of Standards and Technology Gaithersburg MD, 2001.
- Shivraj, V. L., et al. "One time password authentication scheme based on elliptic curves for Internet of Things (IoT)." Information Technology: Towards New Smart World (NSITNSW), 2015 5th National Symposium on. IEEE, 2015.
- Kelly, Daniel, and Mohammad Hammoudeh. "Optimisation of the public key encryption infrastructure for the internet of things." Proceedings of the 2nd International Conference on Future Networks and Distributed Systems. ACM, 2018.
- Rajagopalan, Sundararaman, et al. "IoT Framework for Secure Medical Image Transmission." 2018 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2018.
- Aman, Muhammad Naveed, Kee Chaing Chua, and Biplab Sikdar. "Physically secure mutual authentication for IoT." Dependable and Secure Computing, 2017 IEEE Conference on. IEEE, 2017.
- Aman, Muhammad Naveed, Kee Chaing Chua, and Biplab Sikdar. "Mutual authentication in IoT systems using physical unclonable functions." IEEE Internet of Things Journal 4.5 (2017): 1327-1340.

- 10 Patrícia R. Sousa et al.
- Kolbel, C. "privacyIDEA Authentication System, Release 2.17." online], [retrieved Jan. 23, 2017]. Retrieved from the internet.
- Popp, Nicolas. "Token authentication system and method." U.S. Patent No. 8,639,628. 28 Jan. 2014.
- Ponemon, 2018 global PKI trends study https://bit.ly/2EEkQjJ [Accessed 13-11-2018]
- Neuman, B. Clifford, and Theodore Ts'o. "Kerberos: An authentication service for computer networks." IEEE Communications magazine 32.9 (1994): 33-38.
- Rescorla, Eric. SSL and TLS: designing and building secure systems. Vol. 1. Reading: Addison-Wesley, 2001.
- 14. Solano, Antonio, et al. "A self-provisioning mechanism in OpenStack for IoT devices." Sensors 16.8 (2016): 1306.
- 15. Sousa, Patrícia R. Sousa R., et al. "pTASC: trustable autonomous secure communications." 20th International Conference on Distributed Computing and Networking 2019.
- Hirmer, Pascal, et al. "Automating the Provisioning and Configuration of Devices in the Internet of Things." CSIMQ 9 (2016): 28-43.
- 17. Gueron, Shay, and Vlad Krasnov. "Fast prime field elliptic-curve cryptography with 256-bit primes." Journal of Cryptographic Engineering 5.2 (2015): 141-151.
- Elliptic Curve Integrated Encryption Scheme (ECIES) with Rabbit. https://bit. ly/2JPoNGv [Accessed 29-05-2019]
- Hwang, Yong Ho. "Iot security & privacy: threats and challenges." Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security. ACM, 2015.
- 20. Singh, Kiran Jot, and Divneet Singh Kapoor. "Create Your Own Internet of Things: A survey of IoT platforms." IEEE Consumer Electronics Magazine
- A set of tools and scripts useful to learn the basics about Elliptic Curve Cryptography. (2015) https://github.com/andreacorbellini/ecc [Accessed 29-07-2019]
- 22. Brekalo, Helena, Raoul Strackx, and Frank Piessens. "Mitigating password database breaches with Intel SGX." Proceedings of the 1st Workshop on System Software for Trusted Execution. ACM; New York, NY, USA, 2016.