Key-Server Adaptation to IoT Systems

Jacek Wytrębowicz
Institute of Computer Science, Warsaw University of Technology, ul. Nowowiejska 15/19, Warszawa, Poland

Keywords: Digital Certificate, Key-Server, IoT System.

Abstract: This paper presents an opinion on future evolution of secure communication in IoT systems. Due to advances in cryptography, in processing power of integrated circuits, and in energy harvesting, the constraints of today's IoT devices will weaken and asymmetric encryption could be widely applied. The number of IoT related certificates would grow; so appropriate certificate servers should appear to support them. The paper points a direction for further works on such infrastructure, indicating suitable technology to be applied.

1 INTRODUCTION

The need of securing IoT systems is incontestable. However, it is not easy to satisfy that necessity, especially when an IoT system is based on Wireless Sensor Networks (WSN). The difficulties in protecting the IoT systems arise due to limited computational resources, which result from expected low price of the IoT devices and energy supply constraints. There are a lot of research and industrial solutions that try to tackle this problem. A holistic view on security requirements, and attacks with their corresponding countermeasures in WSNs is presented in the article (Wang et al. 2006). The applicability and limitations of existing Internet security protocols in the context of IoT are discussed by (Nguyen et al. 2015). A survey of the main research challenges and the existing solutions in the field of IoT security is given by (Sicari et al. 2015). Those three above-mentioned surveys are just examples, however they give reach reference lists to related publications. Moreover, a comprehensive description of today's solutions and research directions for identity management for IoT can be found in the book (Mahalle and Railkar 2015). There is also an active document presenting State-of-the-Art and Challenges for the IoT Security (Garcia-Morchon and Sethi 2017), which is regularly updated by IETF from 2011 (last seen update in September 2017).

In recent 25 years the development of secure industrial solutions was incentivised by growing market of smartcards and Automatic Teller Machines, and by increasing threats in computer networks. Today we have co-processors designed to perform computationally intensive cryptographic operations – called cryptography accelerators. They are parts of processors (e.g. Intel's AES-NI or Analog Devices' ADuCM302x) or separate integrated circuits (e.g. Microchip's ATSHA204A, ATAES132A and ATECC508A). ADuCM302x is assigned for IoT devices; it supports AES 128/256 and SHA256 cryptographic operations. ATECC508A is also assigned for IoT devices, and it supports ECDH (Elliptic Curve Diffie–Hellman) key agreement computations. Moreover, there are IP cores for semiconductors for the security and cryptography, e.g. from IP Cores, Inc. The progress in integrated circuit design allowed construction of secure cryptoprocessors (named also embedded secure elements) that are built in smartcards, ATMs, SIM cards, TV set-top boxes, and military applications. A secure cryptoprocessor does not output sensitive data and minimizes the need to protect the rest of the device.

Most of research work in IoT security is based on past experiences and existing standards, trying to overcome both past and today's problems. However, regarding the advances in integrated circuit technologies and in energy harvesting for IoT, we could ask the following questions. Would the constraints to processing power, memory size and communication volume, be valid in the future? Would the future technical challenges be different? Would they rather arise from IoT application needs? One of such needs, we can anticipate, is certificate

Wytrębowicz, J.
Key-Server Adaptation to IoT Systems.
DOI: 10.5220/0006670201550160
In *Proceedings of the 7th International Conference on Sensor Networks (SENSORNETS 2018)*, pages 155-160
ISBN: 978-989-758-284-4

management problem, which would swell with the increase of IoT deployments. The need of standardization of mechanisms for device bootstrapping and key management was already advocated by (Keoh et al. 2014), who give a good review of IETF efforts to standardize security solutions for the IoT ecosystem.

Digital certificates are efficient for authentication of communicating parties, for integrity checking of received data, and for exchange of temporal ciphering keys if confidentiality is needed. The rising number of IoT devices together with their short life-cycle make any central or even hierarchical certificate management inefficient. Today’s solutions either do not provide any certificate management functionality or provide a local or private solution that is not certain for every Internet user. Hence a certificate management infrastructure available in the global Internet is to provide. Let us imagine the requirements for such infrastructure; further we call it Certificate Management Infrastructure for IoT (CM4IT).

2 WHO USES DIGITAL CERTIFICATES IN IoT?

There is no single architecture for an IoT system to be deployed. However, there are always IoT devices, and very often gateways separating them from the outside network. Depending on application of the system, different parties can be engaged in its development, maintenance and usage, and of course different security requirements. Anyway we can distinguish the parties, and the trust dependencies between them. Let’s assume that the general IoT system consists of: numerous devices (microcontrollers embedded in “things”), gateways separating devices from the Internet or Intranet, servers laying in direct proximity to a gateway or somewhere in the network cloud. There are end-users accessing (locally or remotely) services of the servers and an administrator of the given IoT system. We can distinguish the following engaged parties: owner of the system, a maintenance company (usually integrator of the system), software development companies, and hardware manufacturers.

The access management for human users is behind our consideration – there are many well-known solutions for it. For sure, digital certificates of public keys belonging to humans can be applied. Here we are interested in securing communication between components of an IoT system and in sure identification of the components. The communication enables transmission of collected data and control commands, as well as configuration parameters, test sequences and software updates.

A human user can wish to check the identity of the network services he is connected to. The owner of the IoT system is the trusted party for him. In many cases the owner is the company employing the user. Hence identity certificates of the services should be signed by the company or at least by the company maintaining the IoT system. The maintenance company is a trusted party for the owner; they are bound together with some contracts with defined responsibilities. If the maintenance company is not the integrator of the system, then it has to trust the integrator and have an appropriate legal agreement with it. Then the identity certificates of the services should be also signed by the integrator. It is worth pointing out that the maintaining company of a given IoT system as well as the integrator company can change over time. In that case new signatures should be added to identity certificates according to the new responsibilities.

The integrator should verify identity of the components (hardware and software) used to build the system. By hardware components we mean IoT devices and gateways. The verification can be done apart from the system before assembling it or within the system during its operation, or both. Checking the components’ identity in run time strengthens security of the system – it facilitates software component upgrades, and it protects against unsolicited substitutions of hardware components. Hence all critical components should have identity certificates. The identity of a software component can be just its cryptographic hash signed by the issuer’s private key. The best identity of a hardware component is its public key signed by the manufacturer, and the corresponding private key should be protected in a cryptoprocessor built-in the hardware.

According to the above-mentioned scenario, an identity certificate of hardware or software component should be signed by its producer, by the system integrator and if needed by a maintenance company. The identity of critical IoT services available in the network should be signed at least by the integrator and if possible by owner of the system.

We can also notice that there is digital equipment that has to be approved by a certification body, e.g. some medical devices. An IoT system can fall into such category. Therefore, the certification body is a party involved in the system exploitation and could digitally sign its elements to endorse their legality.

The certificates should be stored in a distributed manner for resiliency reason, and should be accessible for the IoT system in order to enable run-time verification. Any changes in the composition of the system (e.g. software updates, adding or replacing IoT devices and services) should be reflected in the certificate databases.

An identity is certified by its issuer (e.g. device manufacturer, service administrator). Next, several certificates can be attached to the identity, which reflects state or ownership changes of the related item. That creates cryptographically secure history of the item. The history records are not supposed to be numerous, thus not leading to scalability problem.

As it has been shown, there are many parties involved in issuing and signing an identity, and there is no common infrastructure for exchanging and managing certificates for IoT systems.

3 DO TODAY’S TECHNIQUES SUIT IoT SYSTEMS?

Digital certificates have been used in the Internet for almost 30 years. Law regulations adopting Public Key Infrastructure (PKI) for commercial operations started to appear in the mid of 90-ties in the last century. PKI is based on X.509 standard of digital certificates. The Certificate Management Protocol defined in RFC 4210 is used for exchanging X.509 digital certificates. PKI is based on a certificate authority which is responsible for identity vetting of a given entity, keys generation, issuing certificate and maintenance of certificate revocation list.

Another popular approach to manage digital certificates is the web-of-trust scheme, which is based on self-signed certificates and third party attestations of those certificates. Here, the OpenPGP standard (RFC 4880) is used for certificate representation. The certificates are stored on and distributed by key servers using HKP (HTTP Keyserver Protocol, however it can work over HTTPS too) or LDAP/LDAPS. Moreover end users can exchange OpenPGP certificates by other means, without a key server intermediary.

A new approach is to apply blockchain technology, which provides a distributed and unalterable ledger of information. Public keys and certificates could be the information registered on such ledger. The main advantage of that approach is unalterable full trace of all interactions kept reliably. Its main disadvantages are: growing size of the register and energy consumption of related servers if a proof-of-work schema is used to guarantee correctness of registered records.

All these approaches have their known advantages and limitations. PKI is widely used to give some security while accessing web services. All web browsers have built-in trusted root certificates and mechanisms for checking X.509 certificates. Hence it is straightforward to provide the PKI certificates to web servers providing IoT services. Even though the resulting trust level is not very high, it is satisfactory for many users of IoT services. In order to provide more secure communication, a custom application for user devices is often proposed, and additional identity vetting procedures are applied.

An IoT system integrator composes the system from software components coming from different developers. An IoT device (and a gateway too) in the system should accept only upgrades accepted by the integrator/maintainer, thus the updating module pre-installed on the device should check the integrator signature (and probably also the solicited time of update). Then the device does not have to check signatures coming from different developers, probably even unknown in advance. Integrator needs a system for update management to process this task and minimizing risk of failures. The system should allow to move back an update and allow to log performing tasks. It also should verify signatures of the software components. A local certificate server could be helpful for managing signatures of all software components installed in the IoT system over the time. The certificate server and the update management system should support different certificate standards used by software developers. The developers can sign their code using public or private PKI, or using their own self-generated key embedded or not in the OpenPGP structure.

IoT services running on network servers should also be updated as well as the underlying operating systems. Most operating systems (Windows, Mac OS X, and most Linux distributions) have built-in mechanisms for updates using signed code, and a system administrator just has to control the process according to defined policy. However, IoT services running on the servers should be updated in the same way as software components running on IoT devices (as described above).

IoT devices and gateways can contain built-in mechanisms for firmware updates. These mechanisms should store the manufacturer signature and check origin and integrity of firmware updates.

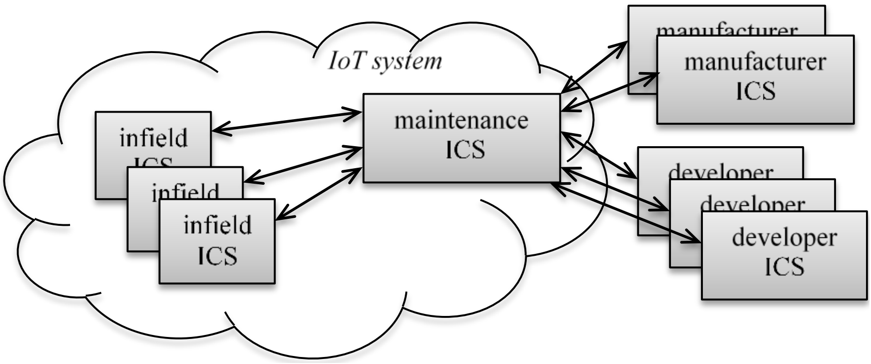

Figure 1: Example structure of identity certificate servers.

Administrator of the system should only control the process. In that case, similarly to the updates of operating systems, integrator signatures cannot be involved.

An IoT integrator/maintainer composes the system from numerous devices coming from different manufacturers. A database of installed and approved devices is required not only for maintenance purposes, but also to enable secure communication inside the system disabling any attempt of intrusion of illegal devices to the system. Communicating entities should identify each other, and only those of them whose identity certificates are successfully checked in the database can cooperate. It is the integrator/maintainer who approves the entities. He can sign the approval digitally on the identity certificate of the entities.

It would be helpful for the integrator to have an identity certificate server, which synchronizes selected data with corresponding servers belonging to IoT device manufactures and software component developers. As far as we know there is no such possibility yet.

4 ARCHITECTURE OF CM4IT

The below-described architecture of Certificate Management Infrastructure for IoT is just illustrative, it depicts the idea of an infrastructure supporting cooperation between different parties involved in construction, deployment and maintenance of an IoT system. The infrastructure includes several identity certificate servers (ICS). Lets call them: infield, maintenance, manufacturer, and developer ICSes, see Fig. 1.

The infield ICS is installed in the proximity of a gateway (or several gateways). It can even be embedded in a gateway. Its aim is to provide run-time certificates for all communicating entities belonging to the cloud under control of a given gateway (or gateways). Access time to the server from the entities should be minimized, and it should have high availability enabling reliable communication. The infield ICS stores signed identities of all entities in the cloud, and entities outside the cloud which are allowed to communicate with the cloud entities. A gateway can serve a star-topology network of directly connected devices, or a mesh network – which is much more complex and difficult to secure. The work on key management protocol to secure multi-hop communication in sensor networks (Guermazi et al. 2017) demonstrates the difficulties and a complex solution based on symmetric cryptography. If the sensors would have cryptoprocessors supporting asymmetric cryptography and the gateway would have an infield ICS, then neighbour discovery and routing in the mesh could be simpler and much more secure.

The maintenance ICS is needed for management of the IoT system, and it is part of the IoT system. It stores all certificates of entities belonging to the system, entities that are active, that were active in the past, and that are planned to be activated. It can as well store certificates of ICSes allowed to alter its data.

A manufacturer of IoT devices (or IoT gateways) supports an ICS with identity certificates of all sold devices. The ICS is accessible via the Internet from maintenance ICSes. An IoT integrator/maintainer can download selected certificates, it can also upload a signed and time stamped status of a given device, e.g. “in-use”, “destroyed”, “stolen”. As a result, any other Internet user can check the device status and recognise the status issuer, either on the maintenance or the manufacturer ICS. An identity certificate of a device or gateway should contain the public key (of the device) signed by the manufacturer. Corresponding private key should be stored inside the secure cryptoprocessor assembled in the device.

A software developer of IoT components supports an ICS with stores identity certificates of all sold code and code updates. The certificates do not contain any public keys, just signed and time stamped hashes of provided code. The certificate can have a status signed by the developer, e.g. “depreciated code”. The status should be time stamped too.

If a given code, when running, is a communicating entity in the IoT system, then the entity has to be authenticated. If the entity runs on a device having a cryptoprocessor, then the cryptoprocessor can sign an authentication message to be sent, and even to cipher messages if needed. An IoT device or gateway does not have to be equipped in a cryptoprocessor, when it is placed in a physically secure environment. In that case the ciphering keys are generated on the device during setup time together with the corresponding identity certificate. Then the certificate is stored in corresponding infield and maintenance ICSes. Similarly, IoT services, running somewhere in the Internet or in an intranet, have keys and certificates generated during setup time.

Depending on the purpose of the IoT system, access to the infield and maintenance ICSes can be opened to any Internet query, or can be protected. The first case suits those IoT systems that offer data for public usage directly from IoT devices, guaranteeing their source. In the second case, the IoT system is a private one, and is accessible via the Internet using VPNs or it offers only pre-processed data via network services.

Many IoT systems are closed solutions for private usages. However, there are some publicly accessible IoT devices or services to find in the Internet. There are concepts, and may be already deployments, of federated systems (where services of one owner can access devices belonging to another one). There are also concepts of mobile devices that can move from one gateway to another. The more open is a system the more important is the identity checking. Moreover, trust information about IoT devices and services becomes desired. The ICSes help to quickly find and check identity of entities in question. Moreover, a relevant party may sign trust label of a selected device or service adding that signature to the identity certificate in its ICS. A certification body that approves legality of medical devices or services could be such a party. The certification body can maintain its own ICS to enable easy access to certified by the body identities of approved devices/services.

The described ICSes should be able to communicate with each other, to download required certificates or to update their status. The ICSes belong to different parties and can run on various computer platforms. To facilitate the communication a common protocol has to be chosen and standard data structures for certificate representation has to be applied.

Key servers widely used today could be adapted to the schema described above. A common infrastructure for carrying signed identities of components would be efficient, even if some of them do not have attached public keys (i.e. code identities). LDAP/LDAPS could be applied for message exchange between ISCes. OpenPGP can be used and may be extended as a data structure standard for IoT identity certificates. An attempt of such extension has been already proposed in an Internet-draft (Atkins 2015). This draft defines a set of notations that may be used when signing an IoT device certificate. However the draft does not cover code identities, neither above-mentioned attributes of IoT devices or code status.

5 CONCLUSIONS

The techniques for digital certification available nowadays can be used in IoT systems to improve security of gathered data and to provide better trust for the systems. However they do not allow building such infrastructure of cooperating ICS as described in the previous chapter. The proposed infrastructure could help in more efficient certificate distribution between the parties concerned. It is expected that IoT system will be very numerous. Therefore, we could deploy such ISC infrastructures along with them without scalability problems. Moreover, the infrastructure provides two important features: identification of responsible parties thanks to their certificates, and cryptographically secured states of IoT components (which can change over time). The features could strengthen security of IoT systems, and ease their software design.

We are going to check out the presented above concept building an experimental network of cooperating ICSes. The network will serve a set of sensors attached to a gateway. Stored certificates will be grouped in domains related to responsibilities of ICSes’ owners. Access rights for reading and updating of the domains will be defined in policy descriptors. Automatic certificate synchronization could be performed according to the policies. Moreover, we are going to propose a structure of identity descriptor with signatures attached. Definition of such structure is not obvious, because it would be desirable to support existing standards, which do not favour needed extensions.

Scalability of the presented Certificate Management Infrastructure for IoT results from the fact that new ICSes are installed with new IoT system deployments, and an ICS stores only certificates related to the deployment, or to the responsibility of a given enterprise. The certificates support network services which provide functionalities for end users, as well as services intended for administrators of the IoT systems. The software installed on IoT devices, IoT gateways, and Internet/Intranet servers can apply the certificates to secure communication.

An owner or a user of IoT system may prefer to rely on the system integrator or maintenance company than on an unknown certification authority selling certificates with weak validation level. Vetting of and vouching for identity of IoT devices and services could be more trustable when more parties sign identity certificates and the parties are bound by mutual contacts and contracts.

The aim of this article is to reveal the advantages of commonly accepted Certificate Management Infrastructure for IoT. Such systems would help to build more secure and more trustable IoT solutions. However, the communicating protocol and data structures for the system are yet to be worked out and to be exercised in experimental deployments.

REFERENCES

Atkins, D., 2015. OpenPGP Extensions for Device Certificates. Available at: https://tools.ietf.org/html/draft-atkins-openpgp-device-certificates-03.

Garcia-Morchon, O. & Sethi, M., 2017. State-of-the-Art and Challenges for the Internet of Things Security. Available at: https://tools.ietf.org/html/draft-irtf-t2trg-iot-seccons-07.

Guermazi, A. et al., 2017. KMMR: An Efficient and scalable Key Management Protocol to Secure Multi-Hop Communications in large scale Wireless Sensor Networks. *KSII Transactions on Internet and Information Systems*, 11(2), pp.901–923. Available at: http://itiis.org/digital-library/manuscript/1606.

Keoh, S.L., Kumar, S.S. & Tschofenig, H., 2014. Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal*, 1(3), pp.265–275.

Mahalle, P.N. & Railkar, P.N., 2015. Identity Management for Internet of Things, *River Publishers*.

Nguyen, K.T., Laurent, M. & Oualha, N., 2015. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks*, (32), pp.17–31.

Sicari, S. et al., 2015. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 76, pp.146–164. Available at: http://dx.doi.org/10.1016/j.comnet.2014.11.008.

Wang, Y., Attebury, G. & Ramamurthy, B., 2006. A Survey of Security Issues In Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 8(2), pp.1–23.