Poster Abstract: Security Comes First, A Public-key Cryptography Framework for the Internet of Things

Hossein Shafagh
Department of Computer Science
ETH Zurich, Switzerland
Email: shafagh@inf.ethz.ch

Anwar Hithnawi
Department of Computer Science
ETH Zurich, Switzerland
Email: hithnawi@inf.ethz.ch

***Abstract*—Novel Internet services are emerging around an increasing number of sensors and actuators in our surroundings, commonly referred to as smart devices. Smart devices, which form the backbone of the *Internet of Things* (IoT), enable alternative forms of user experience by means of automation, convenience, and efficiency. At the same time new security and safety issues arise, given the Internet-connectivity and the interaction possibility of smart devices with human's proximate living space. Hence, security is a fundamental requirement of the IoT design. In order to remain interoperable with the existing infrastructure, we postulate a security framework compatible to standard IP-based security solutions, yet optimized to meet the constraints of the IoT ecosystem. In this ongoing work, we first identify necessary components of an interoperable secure End-to-End communication while incorporating *Public-key Cryptography* (PKC). To this end, we tackle involved computational and communication overheads. The required components on the hardware side are the affordable hardware acceleration engines for cryptographic operations and on the software side header compression and long-lasting secure sessions. In future work, we focus on integration of these components into a framework and the evaluation of an early prototype of this framework.**

I. Introduction

The emerging smart devices in our immediate environment perform sensing and actuating tasks which enable unique forms of experience with our surroundings. Given the integrity of smart devices in our proximate living space, they deal with sensitive data of our daily lives and can perform malicious actuating commands with critical safety implications on our lives. For example, cyber criminals could launch an attack to turn on heatings of countless households at the same time to trigger a collapse of the power grid, or misuse smart devices in form of botnets[1]. Hence, in order to prevent leakage of sensitive data and prevention of malicious actuating tasks, security is vital for the IoT ecosystem. More importantly, since the end-user can not easily modify smart devices, security must be already designed into the system. Security in the IoT must ensure secrecy and integrity of communication, as well as the authenticity of messages.

Smart devices are inherently resource constrained with regards to processing power, memory, communication bandwidth, and energy. We consider smart devices that are equipped with only a few MHz of computation power, several KBytes of RAM and several tens of KBytes of ROM. The main energy sources are typically batteries, energy harvesting, or a combination of both. Since smart devices should operate for long times, energy efficiency is another important factor besides computational and memory efficiency. Currently, the integrated hardware support for symmetric-key cryptography (e.g., AES) provides fast and cheap en/decryption on smart devices. However, an efficient and scalable key management scheme, which is required for symmetric-key cryptography, is still missing. *Public-key Cryptography* (PKC) is a popular method for key establishment, but the high computational overheads are the main boundary for constrained devices.

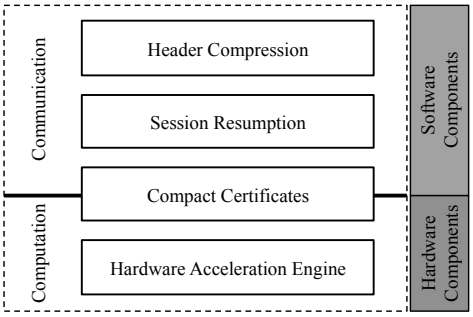

Fig. 1. Main components of a our lightweight PKC-based framework for the IoT, with the affected sides on constrained devices.

In this ongoing work, we *(i)* identify necessary components of a PKC-based security framework for the IoT, as depicted in Fig. 1. These components tackle the computational and communication overheads involved in a PKC-based security protocol. The integration of these components and the herewith related challenges are still unexplored. More importantly, the seamless integration of the hardware acceleration engine into IP-based security protocols, such as DTLS, makes security more affordable on constrained devices. Furthermore, we plan to *(ii)* perform thorough evaluations of our PKC-based framework on constrained devices, with a focus on scalability and efficiency, in terms of computation and communication overheads.

II. Framework Design

Given the resource constraints of smart devices, we focus our work on four different components. The first component deals with reducing the constant header overheads, that are caused by security protocols. Furthermore, we look at possibilities of reducing the occurrences of the most expensive part

[1]"Proofpoint Uncovers *Internet of Things* (IoT) Cyberattack", January 16, 2014: http://www.proofpoint.com/about-us/press-releases/01162014.php

of a secure communication, namely the session establishment. The third component, is the efficient use of certificates for secure authentication and key agreement. Finally, potential usage of hardware acceleration engines for PKC forms the fourth component. In the following, we elaborate on each of these components.

Compression. Security protocols come along with large header sizes. These headers are not designed with the limitations of the constrained devices in mind. For instance, a typical handshake packet of the DTLS protocol has header lengths of 25 Byte and each data packet is concatenated with a header of length 13 Byte. In [1], we show how the transmission overhead of DTLS in constrained networks can be reduced to as few Bytes as 3 for the handshake and 5 Byte for data packets.

Session Resumption. Protocol handshakes for establishing a secure session incur transmission overheads of at least several hundred Bytes. In the IoT paradigm, however, communication becomes more sparse. Hence, handshakes are more often required. Session resumption is a feature used in the Internet, to allow a secure session to last for a longer time. To this end, the security context is not removed after the session teardown and can be used for possible subsequent sessions. In order to mitigate DoS attacks, todays servers offload the corresponding security context toward clients. In [2], we introduce a constrained-device-friendly session resumption method that allows for offloading the security context towards the more powerful end-point.

Certificates. Certificates are widely used in the Web. They bind a public-key to a defined identity. This information is verified via the signature of a *Certification Authority* (CA) as a trusted third party. PKC enables an efficient and scalable key management, since the end-points do not require to share a secret prior to communication. More importantly, with mutual certificate-based authentication no human interaction is required during the authentication, which makes certificates suitable for *Machine-to-Machine* (M2M) communication. The disadvantages of certificates with regards to constrained environments are their large size and the involved PKC computations. To tackle the large size of a certificate, it is important to first of all employ *Elliptic Curve Cryptography* (ECC), instead of RSA. This reduces the key size from 3072 to 256 bits with the same level of security. Furthermore, certificates can become arbitrarily large. In order to avoid this, we create a list of recommendations on creating compact certificates. For instance, avoiding unnecessary extensions goes hand-in-hand with smaller certificate sizes. Moreover, it is important to keep the length of certificate-chains as low as possible. According to Google's pilot log traces for Certificate Transparency[2] 2.4 % of logged certificates are directly signed by a root CA, 67.7 % by one intermediary CA and 28.8 % by two intermediaries. For smart devices, it is important to have certificates which are directly signed by a root CA, in order to keep the transmission overhead as low as possible.

Hardware Acceleration Engine. Traditionally, PKC has been considered not feasible for constrained devices [3]. Existing ECC libraries for constrained devices require few KBytes of static RAM and in order of 10 KBytes of ROM. Given that PKC is used only at session establishments, dedicating memory space during the whole life cycle of a smart device for PKC means less memory space for applications. Recent System-on-Chip (SoC)[3] developments not only offer hardware encryption engines for AES and SHA, but as well acceleration engines for ECC or RSA for a secure key exchange. With an additional cost of 6 % ($\sim$ 0.2 \$), hardware acceleration engines would enable efficient PKC on smart devices, making large PKC libraries dispensable [4], [5].

[2]Certificate Transparency (RFC 6962), http://www.certificate-transparency.org, log trace with 2.8 million entries retrieved at October 17, 2013

III. Framework Integration

The introduced components of our framework can be further optimized in their own domain, however, we are additionally interested in the integration of these components. The header compression is transparent to application layer protocols and is applied to communication within a constrained network. Hence, incoming and outgoing messages are de/compressed at border routers which interconnect a constrained network with the Internet. The session resumption feature is, similar to header compression, transparent to the application layer. This feature, allows the smart device, to offload the encrypted security context of a session to the more powerful end-point. The hardware acceleration engines require drivers to provide an interface for the crypto functionality. We employ the AES interface for the protection of communication. The ECC interface is used during the handshake phase, namely the verification of certificates and key agreement. Regarding the certificates, the engagement of the developer is required. The developer should provide certificates that comply with our recommendations regarding cipher suites and used fields. This allows to have certificates which are compact in size.

IV. Future Work

In this work, we define mechanisms to achieve secure E2E communication with constrained environments involving *Public-key Cryptography* (PKC). To this end, we are considering an early prototype integrating the four components of header compression, session resumption, compact certificates and hardware acceleration engines for crypto. Moreover, we aim at gaining insight about efficiency, usability, sustainability of our framework through extensive evaluations. In our evaluation, we consider the feasibility, scalability and convenient of deployments and maintenance.

References

[1] S. Raza, H. Shafagh, K. Hewage, R. Hummen, T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," in *IEEE Sensors Journal (Volume:13, Issue:10)*, 2013.

[2] R. Hummen, H. Shafagh, S. Raza, T. Voigt, and K. Wehrle, "Delegation-based Authentication and Authorization for the IP-based Internet of Things," in *IEEE SECON*, 2014.

[3] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM (Volume:47, Issue:6)*, 2004.

[4] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Networks (Volume:11, Issue:8)*, 2013.

[5] W. Hu, P. Corke, W. C. Shih, and L. Overs, "secFleck: A Public Key Technology Platform for Wireless Sensor Networks," in *EWSN*, 2009.

[3]TI CC2538 System-On-Chip with ECC hardware acceleration for IEEE 802.15.4 networks, http://www.ti.com/product/cc2538