Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 13, 2014

C. Schmitt
B. Stiller
University of Zurich
T. Kothmayr
TU Muenchen
W. Hu
CSIRO ICT Centre

October 10, 2013

DTLS-based Security with two-way Authentication for IoT

<draft-schmitt-two-way-authentication-for-iot-01>

Abstract

In this draft the first key idea for a full two-way authentication security scheme for the Internet of Things (IoT) based on existing Internet standards, specifically the Datagram Transport Layer Security (DTLS) protocol, is introduced. By relying on an established standard, existing implementations, engineering techniques, and security infrastructure can be reused, which enables an easy security uptake. The proposed security scheme is, therefore, based on RSA, the most widely used public key cryptography algorithm. It is designed to work over standard communication stacks that offer UDP/IPv6 networking for Low power Wireless Personal Area Networks (6LoWPANs).

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Today, there is a multitude of envisioned and implemented use cases for the Internet of Things (IoT) and wireless sensor networks (WSNs).In many of these scenarios it is intended to make the collected data globally accessible to authorized users and data processing units through the Internet. Most of these data collected in such scenarios is of sensitive nature due to the relation to location and personal information or IDs. Even seemingly inconspicuous data, such as the energy consumption measured by a smart meter, can lead to potential infringements in the users' privacy, e.g., by allowing an eavesdropper to conclude whether or not a user is currently at home. From an industry perspective, there is also a pressing need for security solutions based on standards as pointed out by the market research firm Gartner Inc. [1]. Regarding the infrastructure, security risks are aggravated by the trend toward a separation of sensor network infrastructure and applications. Therefore, a true end-to-end security solution is required to achieve an adequate level of security for IoT. Protecting the data once it leaves the scope of the local network is not sufficient.

A similar scenario in the traditional computing world would be a user browsing the Internet over an unsecured WLAN. Assuming attackers in physical proximity of the user it can happen that the attacker can capture the traffic between the user and a Web server. Countermeasures against such attacks include the establishment of a secured connection to the Web server via HTTPS, the use of a VPN tunnel to securely connect to a trusted VPN endpoint, and using wireless network security such as WPA.

These solutions are comparable to security approaches in the IoT area. Using WPA is similar to the traditional use of link layer encryption. The VPN solution is equivalent to creating a secure connection between a sensor node and a security end-point, which may or may not be the final destination of the sensor data. Establishing a HTTPS connection with the server is comparable to the approach described in this draft: The use of the DTLS protocol in an end-to-end security architecture for IoT is investigated, where DTLS is an adaption of the widespread TLS protocol, used to secure HTTPS, for unreliable datagram transport.

1.1. Document Structure

Section 2 mentions conventions used in this draft. Afterwards the assumed high level design requirements are briefly mentioned in Section 3. Section 4 describes idea of bringing DTLS to wireless sensor networks. In this section a brief description of the

standard
DTLS protocol based on RFC 6347 is given, as well as the description

of the proposed solution for a standard based end-to-end security architecture. The assumed use-case with its requirements and architecture is described in Section 5. Section 6 defines the hardware requirements, followed by security considerations. The draft is concluded in Section 8.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

A publisher represents any kind of device that makes its data public available in a network using WLAN or LAN connection.

A subscriber represents any kind of device that wants to access data.

An Access Control server (AC) is an entity in the network that regulates the access of data and issues an access ticket for subscribers based on legal and regulative implications.

3. High Level Design Requirements

Due to the usage of DTLS for establishing an end-to-end security architecture for IoT three high-level design decisions MUST be made.

3.1. Implementation of A Standards Based Design

Standardization has helped the widespread uptake of technologies. Radio chips can rely on IEEE 802.15.4 for the physical and the MAC layer. Routing functionality is provided by the so-called 'IPv6 Routing Protocol for Low power and Lossy Networks' (RPL) [RFC6550] or 6LoWPAN [RFC4944]. COAP [2] defines the application layer. So far, no such efforts have addressed security in a wider context of IoT.

3.2. Focus on Application Layer and End-to-End Security

An end-to-end protocol provides security even if the underlying network infrastructure is only partially under the user's control. As the infrastructure for Machine-to-Machine (M2M) communication is getting increasingly commoditized, this scenario becomes more likely:
The European Telecommunications Standard Institute (ETSI) plans to standardize the transport of local device data to a remote data center. For stationary installations security functionality could be provided by the gateway to the higher-level network. However, such gateways MAY present a high-value target for an attacker. If the

devices are mobile, as it is possible within a logistic application, there may be no gateway to a provider's network that is under the user's control, similar to how users of smart phones connect directly to their carrier's network. Another example that favours end-to-end security is a multi-tenancy office building being equipped with a common infrastructure for metering and climate-control purposes. Tenants share the infrastructure but are still able to keep their devices' data private from other members of the network.

DTLS is located between the transport and the application layer. Thus, it is not necessary that providers of the infrastructure support security mechanisms. It is purely in the hands of the two communicating applications to establish security. If the security is provided by a network layer protocol (e.g., IPsec) the same is true to a lower degree, because network stacks of both devices MUST support the same security protocols.

3.3. Support for Unreliable Transport Protocols

Reliable transport protocols like TCP incur an overhead over simpler, unreliable protocols such as UDP. Especially for energy starved, battery powered devices this overhead is often too costly and TCP has been shown to perform poorly in low-bandwidth scenarios [3]. This is reflected in the design of the emerging standard COAP, which uses UDP transport and defines a binding to DTLS for security [2]. By using DTLS in conjunction with UDP this draft does not force the application developer to use reliable transport - as it would be the application developer to use reliable transport - as it would be the case if TLS would be used. It is still possible to use DTLS over transport protocols like TCP, since DTLS only assumes unreliable transport.

This is a weaker property than the reliability provided by TCP. However, adaptations of DTLS for unreliable transport introduce additional overhead when compared to TLS. There MAY be a benefit in using TCP during the handshake phase but the DTLS reliability mechanism SHOULD be adapted to the special requirements of constraint networks.

4. DTLS Protocol for Wireless Sensor Networks

4.1. DTLS Standard - RFC 6347

The Datagram Transport Layer Security (DTLS) protocol in version 1.2

was standardized under the RFC 6347 [RFC6347]. All messages sent via DTLS are prepended with a 13 Byte long DTLS record header. This header specifies the content of the message (e.g. application data or handshake data), the version of the protocol employed, as well as the 64 bit sequence number and the record length. The top two bytes of the sequence number are used to specify the epoch of the message, which changes once new encryption parameters have been negotiated between client and server.

If no security has been negotiated yet, the DTLS record header is followed by the plaintext, otherwise by the DTLS block cipher. If a block cipher is used, the plaintext is prepended by a random Initialization Vector (IV), which has the size of the cipher block length. This approach protects against attacks where attackers can adaptively choose plaintext. The plaintext is followed by a Hash-based Message Authentication Code (HMAC), which allows the receiver to detect if the DTLS record has been altered. Finally, the message is padded to a multiple of the cipher block length. Unlike TLS, DTLS does not allow for stream ciphers, because they are sensitive to message loss and recording. Instead DTLS uses block ciphers in the Cipher-Block Chaining (CBC) mode of operation.

The key material and cipher suite, consisting of a block cipher and a hash algorithm, are negotiated between the client and the server during the handshake phase, which commences before any application data can be transferred. Three types of handshake exist: unauthenticated, server authenticated, and fully authenticated handshakes. During an unauthenticated handshake neither party authenticated with the other. In contrast, in a server-authenticated handshake only the server proves its identity to the client. In a fully authenticated handshake the client has to authenticate itself to the server as well.

4.2. A Standard Based End-to-End Security Architecture

The proposed system architecture in this draft is following the IoT model. It is assumed that IPv6 connects the Internet and parts of it run 6LoWPAN. The transport layer in 6LoWPAN is UDP, which can be considered unreliable; the routing layer is RPL or Hydro [[3]. Both routing protocols are similar enough and, therefore, a change SHOULD have negligible impact on the results. IEEE 802.15.4 is used for the physical and MAC layer. Based on this protocol stack DTLS was selected as the security protocol and placed in the application layer on top of the UDP transportation layer. Figure 1 shows the network stack used in this draft [6], while BLIP is a special 6LoWPAN implementation including several IP protocols [7].

```
+---------------------------------------+
|  Application Layer: COAP, XML, ....   |
|                                       |
|   +-------------------------+         |
|   | Security Layer: DTLS    |         |
|   +-------------------------+         |
+---------------------------------------+
|  Transport Layer: UDP--               |
|                        |-->BLIP,RPL   |
|  Network Layer: IPv6---               |
+---------------------------------------+
|  Medium Access Layer: IEEE 802.15.4   |
+---------------------------------------+
|  Physical Layer: IEEE 802.15.4        |
+---------------------------------------+
```

Figure 1: Assumed Network Stack

In order to support end-to-end communication security the need for proper authentication of data publishing devices and access control throughout the network is required. Thus, an Access Control (AC) server is integrated in the assumed system architecture. The AC is a trusted entity and a resource-rich server, on which access rights for the publisher (= sensor nodes) of the secured network are stored. The identity of a default subscriber is usually preconfigured on a publisher before it is deployed. If any additional subscribers want to initialize a connection with the publisher, they first have to obtain an access ticket from the AC. The AC verifies that the subscriber has the right to access the information available from the publisher. In the next step the publisher only has to evaluate the identity of the subscriber and has to verify the ticket it has received from the AC. This requires a unique identity for a publisher in the network. In the Internet, identities are usually established via public key cryptography (PKC) and identifiers are provided through X.509 certificates. An X.509 certificate contains, among other information, the public key of an entity and its common name. A trusted third party, called the Certificate Authority (CA), signs the certificate. The CA serves two purposes: Firstly, the signature allows the receiver to detect modifications to the certificate. Secondly, it also states that the CA has verified the identity of the entity that requested the certificate.

5. Use-Case Description

As briefly mentioned in Section 1 collected data is connected to

sensitive information and can lead to potential infringements in the users' privacy. This fact becomes a security risk if the data is transmitted over long distances, perhaps several hops, to a specified global sink [10]. Depending on the setting it might happen that the data is also transmitted via the Internet and might be cached in between. The latter case is inspired by the project FLAMINGO, which deals with access regulations based on legal and regulative implications in IP networks [9]. By definition of the Internet of Things it can be assumed that IP communication is supported by all devices in wireless sensor networks, which allows the adaptation of standards in IP networks to constraint networks.

5.1. Architecture Requirements

In order to show the applicability of the proposed solution throughout the above sections a common network structure consisting of a global sink and several sensor nodes is assumed. Additionally, an Access Control server (AC) is integrated into the network. The AC is a trusted entity and a more resource-rich server, on which the access rights for the publishers (= sensor nodes) of the secured network are stored. Therefore, it is required that each publisher in the networks has a unique identity.

As mentioned in the beginning of the draft the ideas of the concept Internet of Things are the basis, which include also the basic understanding of the Internet. Thus, it is assumed that identities are usually established via public key cryptography and the identifiers provided through X.509 certificates [RFC5280]. In general, X.509 certificate contains the public key of an entity and its common name. A trusted third party - Certificate Authority (CA) - signs that certificate. This signing allows the receiver to detect modifications to the certificate and that the identity of the entity, who requested the certificate, has been verified by the CA. The CA can be run by the administrator of the network or an established Internet certificate authority can be used.

Furthermore, it is assumed that the identity of a default subscriber is usually preconfigured on a publisher before it is deployed.

5.2. Data Access Procedure

Based on the FLAMINGO project the following use-case is assumed [9]: A sensor node has published its data, which is transmitted in direction to the global sink. In between the data can be cashed in order to make it accessible more quicker to subscribers. In this case the cashed entity functions as a publisher. Assuming the new subscriber wants to access the data, it must initialize a connection with the publisher. Therefore, the subscriber MUST obtain an access ticket from the AC before. The functionality of the AC is to verify that the subscriber has the right to access the data available from the publisher. Those rights are influenced by legal and regulative implications (e.g., rights connected to an ISP region, where the subscriber belongs to). If the subscriber received a valid access ticket, it is presented to the publisher. The publisher must evaluate the identity of the subscriber and verify the ticket it has received from the AC. If the validation was successful the subscriber can access the data. Before every kind of data exchange, where sensitive information is involved, takes place the proposed two-way authentication handshake is performed in order to establish a highly secured communication channel between the entities.

6. Hardware Requirements

Hu et al. showed that RSA, the most commonly used public key algorithm in the Internet, can be used in sensor networks with the assistance of a Trusted Platform Module (TPM), which costs less than 5% of a common sensor node [4]. A TPM is an embedded chip that provides tamper proof generation and storage of RSA keys as well as hardware support for the RSA algorithm. The certificate of a TPM equipped publisher and the certificate of a trusted CA MUST be stored on the publisher prior to deployment.

For publishers that are not equipped with TPM chips the authentication can be proposed via the DTLS pre-shared key cipher-suite, which requires a small number of random bytes, from which the actual key is derived, to be preloaded to the publisher before deployment. This secret MUST also be made available to the AC server, which will disclose the key to devices with sufficient authorization.

7. Security Considerations

The following security goals are addressed by the key idea presented in this draft:

Authenticity

Recipients of a message can identify their communication partners and can detect if the sender information has been forged.

Integrity

Communication partners can detect changes to a message during transmission.

Confidentiality

Attackers cannot gain knowledge about the content of a secured message.

By choosing DTLS as the security protocol those goals can be achieved. DTLS is a modification of TLS for the unreliable UDP and inherits its security properties [5].

8. Conclusion

In this draft the key idea of a standard-based security architecture with two-way authentication for the IoT was introduced. During a fully authenticated DTLS handshake authentication can be performed, while the handshake is based on an exchange of X.509 certificates containing RSA keys. The proposed architecture provides message integrity, confidentiality, and authenticity with affordable energy, end-to-end latency, and memory overhead [6]. Thus, it can be assumed that DTLS is a feasible security solution for the emerging IoT. A fully authenticated handshake with strong security through 2048 bit RSA keys is considered as feasible for sensor nodes equipped with a TPM chip, since a fully authenticated, RSA-based handshake consumes as little as 488 mJ [6]. These additional memory requirements of fewer than 20 kB RAM are well below the 48 kB of memory offered by the sensor node used [6].

Sensor nodes without a TPM chip forego protection against physical tampering, but can still perform a DTLS handshake based on Elliptic Curve Cryptography (ECC), which could be performed on the same platform with little more than 100 mJ of energy usage [6].

For the future it MAY be possible to apply these techniques to DTLS together with an Authenticated Encryption with Associated Data (AEAD) mode of operation. Another focus MAY be the inclusion of more constrained nodes without a TPM in the proposed architecture, for which a variant of the DTLS pre-shared key cipher suites SHALL be used.

9. Acknowledgement

This work was supported partially by the SmartenIT [8] and the FLAMINGO [9] projects, funded by the EU FP7 Program under Contract No. FP7-2012-ICT-317846 and No. FP7-2012-ICT-318488, respectively.

10. Formal Syntax

6LoWPAN - IPv6 over Low power Wireless Personal Area Network (RFC 4944)

AC - Access Control Server

BLIP - Berkeley Low-power IP stack

CA - Certificate Authority

CBC - Cipher-Block Chaining

COAP - Constrained Application Protocol

DTLS - Datagram Transport Layer Security protocol (RFC 6347)

ECC - Elliptic Curve Cryptography

ETSI - European Telecommunications Standard Institute

HMAC - Hash-based Message Authentication Code

IoT - Internet of Things

IV - Initialization Vector

PKC - Public Key Cryptography

RPL - Routing Protocol for Low power and Lossy Networks (RFC 6550)

TCP - Transmission Control Protocol (RFC 793)

TLS - The Transport Layer Security (TLS) Protocol Version 1.2 (RFC 5246)

TPM - Trusted Platform Module

UDP - User Datagram Protocol (RFC 768)

WSN - Wireless Sensor Network

11. References

11.1. Norminative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[2] Shelby et al., Z., "Constrained Application Protocol (CoAP), http://tools.ietf.org/html/draft-ietf-core-coap-14", March 2013.

[3] Dawson-Haggerty et al., S., "Hydro: A Hybrid Routing Protocol for Low-power and Lossy Networks", In Proceedings of the 1st IEEE International Conference on Smart Grid Communications, SmartGridComm, Gaithersburg, Maryland, U.S.A.. , 2010.

[5] Modadugu et al., N., "The Design and Implementation of Datagram TLS", In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, California, U.S.A.. , 2004.

11.2. Informative References

[1] LeHong, H., "Hype Cycle for the Internet of Things", Tech. rep., Gartner Inc. , 2012.

[4] Hu, W., "Toward Trusted Wireless Sensor Networks", ACM Transactions on Sensor Networks, Vol. 7, No.5. , 2010.

[6] Kothmayr et al., T., "DTLS Based Security and Two-Way Authentication for the Internet of Things", Elsevier, Journal Ad Hoc Networks , 2013.

[7] Dawson-Haggerty, S. and D. Culler, "Berkeley IP Information, Berkeley WEBS Wireless Embedded Systems, http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip", 2010.

[8] SmartenIT Consortium, "Socially-aware Management of New Overlay Application Traffic combined with Energy Efficiency in the Internet (SmartenIT), http://www.smartenit.eu/", 20103.

[9] Flamingo Consortium, "FLAMINGO - Management of the Future Internet, http://www.fp7-flamingo.eu/", 2013.

[10] Schmitt, C., "Secure Data Transmission in Wireless Sensor Networks, Dissertation", Network Architectures and Services (NET), ISBN: 3-937201-36-X, ISSN: 1868-2634 (print), DOI: 10.2313/NET-2013-07-2 , 2013.

Authors' Addresses

Corinna Schmitt
Univerity of Zurich
Department for Informatics
Communication Systems Group
Binzmuehlestrasse 14
Zurich 8050
Switzerland

Email: schmitt@ifi.uzh.ch

Burkhard Stiller
Univerity of Zurich
Department for Informatics
Communication Systems Group
Binzmuehlestrasse 14
Zurich 8050
Switzerland

Email: stiller@ifi.uzh.ch

Thomas Kothmayr
Technische Universitaet Muenchen
Department of Informatics
Datenbanksysteme (Informatik 3)
Boltzmannstr. 3
Garching 85748
Germany

Email: kothmayr@in.tum.de

Wen Hu
CSIRO ICT Centre
1 Technology Court
Pullenvale QLD 4069
Australia

Email: Wen.Hu@csiro.au