Securing and Integrating the IoT with a Smart Home Router

Fergus W. Leahy

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Masters project proposal

16/12/2013

Contents

1 Introduction

> *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."*
>
> – Mark Weiser, *The Computer for the Twenty-First Century, 1991*

The modern home is becoming increasingly filled with a variety of *connected* devices (laptops, tablets, phones, set-top boxes etc.), providing a myriad of different services to users within the home. On top of this, with the advent use of smart phones and introduction of wearable devices, we too are starting to carry around our own personal network of devices everywhere we go, brushing past many others in our daily lives at home, work and on the street. Although all connected to the Internet, these devices are often encapsulated within their own environment and ecosystem, unable to interconnect, creating a fractured and often complex user experience.

Making matters more interesting, the Internet of Things paradigm is once again becoming a field of great interest due to the advent of cheap, low power wireless embedded devices [1]. However, not much consideration has been made for how these Things should be integrated into the existing home network, with many approaches opting to simply bridge the device to the cloud ([3], [4], [15]), with obvious concerns for security, privacy and up-time.

As these devices enter our homes and pockets, bringing with them their own ecosystems, the user is faced with the increasingly difficult burden of managing all of them and the ecosystems [7], [6]. Due to the sheer number and diversity of these devices, many of which will provide overlapping services and functionality, problems arise in how to ensure these devices not only play nicely together but also ensuring the user's network and information stays secure against new and unanticipated threats.

In order for these multiple layered networks of devices to truly fade away into the fabric of our everyday lives, a platform and relevant protocols need to be engineered to not only support this heterogeneous network securely, but also aid the user in managing both the network and the privacy of their information.

The Homework home router platform was created to these issues. Rather than assume every user is a network administrator, the project investigated the needs and abilities of the average user in order to propose the future of home networking, re-inventing the protocols, models and architectures to truly suit the home environment. This re-invention of the home router allows a user to easily install, manage and use their home network, without the need of a Cisco qualification.

In regards to the Internet of Things development, previous work demonstrated that it was in need a suitable protocol in order to meet the specific needs of a network of Things [12]. Thus, a new protocol was designed and implemented, which could not only run on even the most constrained battery-powered devices (8MHz), but it could also efficiently scale to support hundreds of Things within the same network.

2 Statement of Problem

The Internet of Things protocol created in [12] proved to be a successful proof-of-concept; However, in order for it to be considered for deployment and integration into existing homes, several issues need to first be addressed.

Security: Due to time constraints the initial design of the IoT protocol didn't consider security concerns. However, the IoT protocol needs to be sufficiently secured to prevent eavesdropping of the transmitted data and injection of false events by perpetrators masquerading as sanctioned participants in the network.

Integration: The current implementation exists as a standalone component with several demo applications. Integration of the IoT protocol into a user-friendly platform is necessary to harness the full power of the Thing's network. The integrated platform would then be able to search and connect to available Things, receive events from the sensors and using user customised rules, use automata to detect if the received events match and then perform actions by pushing commands to actuators in the network.

2.1 Intranet of Things vs Internet of Things

As described earlier in section 1, many previous deployments of Internet of Things networks have taken a cloud first approach, see [3, 4]. Whilst this yields certain benefits, such as easy external access and integration with other services [2, 5], it also poses several questions regards data security, privacy and up-time. For this project, the focus will be on developing a home first platform, in which all Things communication will be kept local, with no cloud processing involved; Thus a more suitable name, the Intranet of Things, will be used.

2.2 Project Outcomes

Outcomes of the project:

- An extended IoT protocol with sufficient security to prevent eavesdropping and unsanctioned devices.
- An Extended home information platform (Homework), with the IoT controller role implemented, enabling capturing of IoT events from sensors and creation of commands for actuators.
- Use of Homework's automata to implement closed loop control of Things in the home, subscribing to sensor events, processing rules and publishing to actuators to perform actions.

3 Literature Review

This section discusses previous work on Internet of Things protocols, WSN security and home networking which provides the motivation and possible aid in solving the previously mentioned issues.

3.1 State-of-the-art IoT Protocols

3.2 Homework - Smart Home Router

3.3 Symmetric Security - TinySec, MiniSec, ContikiSec

3.3.1 TinySec

TinySec is a fully functional symmetric security link layer component created for the wireless sensor network operating system, TinyOS. It was the first fully implemented solution for WSNs and was created to address the security worries of running a WSN and transmitting private sensor data in the clear. Unlike conventional security protocol implementations which can afford significant time and space overheads, such as 16-32 bytes for security per packet, WSN typically run on extremely constrained devices with packet sizes of just 30 bytes, making those implementations impossible/extremely expensive to run.

To resolve this, TinySec took a balanced approach making a compromise between the level of security, packet overhead and resource requirements. The end result proved that it's possible to secure a WSN efficiently entirely in software, without the need for additional hardware.

Communication between nodes, not just nodes-to-base-station, in WSNs is often quite important, allowing nodes to not only redirect other's traffic along routes but also consolidate duplicate packets from multiple nodes about the same event, saving the overall network from wasting power receiving and transmitting the extra packets; TinySec chose to engineer in security at the link layer approach allowing these mechanisms to perform without alteration. The security goals of TinySec aimed to enable access control, whereby only authorised participants may participate in the network, with unauthorised messages easy to spot and reject; ensure message integrity, so that authorised messages can't be illegally altered by a man-in-the-middle without the receiver noticing; and ensure confidentiality, to ensure information is kept secret from unauthorised eavesdroppers.

The TinySec implementation uses Cipher Block Chaining with an initialisation vector (IV), together these achieve semantic security, therefore ensuring that encrypting a plain text twice returns a different cipher text each time. So that the receiving end knows how to begin decryption of the data, the IV must be sent along in the clear with the encrypted data. When using an IV, it's length needs to be taken into consideration because repeats

will occur when the number wraps, causing a security vulnerability. On unconstrained devices an IV is usually 8 or 16 bytes, however due to the packet size limitations of the wireless sensors used, a 4 byte IV was chosen.

For ensuring authenticity and integrity of messages, TinySec uses Cipher Block Chaining Message Authentication Codes (CBC-MAC) of 4 bytes in length. Similar to a CRC, CBC-MAC runs over the data and produces a 4 byte MAC which is appended to the packet. If a message was to be altered, the attacker has a 1 in 2^{32} of blindly forging a valid MAC. In a WSN with a limited send rate of 19.2Kb/s it would take over 20 months to send enough packets to possibly succeed in forging a MAC. In the case of attack, a receive heuristic could be used to detect multiple failed MAC transmissions at a nearby node, triggering an alert to the rest of the network.

TODO: problems (key distribution problem, capture)

3.3.2 MiniSec

MiniSec was created to tackle several problems apparent in the then current WSN security protocols, TinySec and Zigbee. The pre-cursor to MiniSec, TinySec, received much attention and use due to its power and resource efficient security implementation, but because of a lack of authentication and replay prevention the overall security of it is insufficient to protect a WSN. An commercial alternative, Zigbee, has significantly higher security, but does so at the cost of high energy consumption. MiniSec was designed to find the middle-ground between the two, increasing security whilst remaining energy efficient.

TODO: Discuss improved security, improved E+AUTH with OCB, use of authentication (OCB) and replay prevention mechanism

3.3.3 ContikiSec

[10, 14, 8]

3.4 Key Distribution Problem

Tried to solve using Faraday cages...[11] Propose to use Asymmetric Security aka Public Key crypto

3.5 Asymmetric Security - TinyECC and Certificates

3.5.1 TinyECC

[13] A public key crypto implementation of Elliptic Curve cryptography, supporting higher effective security with smaller keys (than RSA).

3.5.2 Certificates and Public Key Infrastructures

Discuss the use of certificates and certificate authority to pass trust up the chain.

3.6 Other Works

3.6.1 MQTT

3.6.2 IETF Work

[9, 15]

4 Proposed Approach

4.1 Security Architecture

4.1.1 Symmetric Key Cryptography

4.1.2 Asymmetric Key Cryptography

4.2 Implementation of IoT Protocol on TinyOS

4.3 Integration of IoT with Smart Home Router

5 Work Plan

- Secure IoT Protocol
- Implement Secure IoT Protocol on TinyOS
- Port Secure IoT Protocol to Homework Automata

References

[1] 2013: The year of the Internet of Things. http://www.technologyreview.com/view/509546/2013-the-year-of-the-internet-of-things/. Accessed: 21/03/2013.

[2] If This Then That service. https://ifttt.com/. Accessed: 21/03/2013.

[3] Smart Things IoT platform. http://smartthings.com/. Accessed: 21/03/2013.

[4] Twine "Internet of Things" Thing. http://supermechanical.com/. Accessed: 21/03/2013.

[5] Xively, Internet of Things public cloud. https://xively.com/. Accessed: 21/03/2013.

[6] Anthony Brown, Richard Mortier, and Tom Rodden. Multinet: Reducing interaction overhead in domestic wireless networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1569–1578, New York, NY, USA, 2013. ACM.

[7] Patrick Brundell, Andrew Crabtre, Richard Mortier, Tom Rodden, Paul Tennent, and Peter Tolmie. W-must'11 best papers-the network from above and below. *SIGCOMM-Computer Communication Review*, 41(4):519, 2011.

[8] Lander Casado and Philippas Tsigas. Contikisec: A secure network layer for wireless sensor networks under the contiki operating system. In Audun Jsang, Torleiv Maseng, and SveinJohan Knapskog, editors, *Identity and Privacy in the Internet Age*, volume 5838 of *Lecture Notes in Computer Science*, pages 133–147. Springer Berlin Heidelberg, 2009.

[9] Castellani. CoAP to HTTP mapping. https://datatracker.ietf.org/doc/draft-castellani-core-http-mapping/. Accessed: 21/03/2013.

[10] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 162–175, New York, NY, USA, 2004. ACM.

[11] Cynthia Kuo, Mark Luk, Rohit Negi, and Adrian Perrig. Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 233–246, New York, NY, USA, 2007. ACM.

[12] F.W. Leahy. A lightweight protocol for constrained devices for use in the Internet of Things paradigm. Technical report, University of Glasgow, 2013. 4th Year Dissertation.

[13] An Liu and Peng Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 245–256, 2008.

[14] Mark Luk, Ghita Mezzour, Adrian Perrig, and Virgil Gligor. Minisec: a secure sensor network communication architecture. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 479–488. IEEE, 2007.

[15] Z. Shelby. IETF, Constrained RESTful Environments - Resource Discovery, 2012. RFC6690, 1.2.1.