This is a pre-print. The final version will appear in the proceedings of the 19th IEEE International Sym
Mobile and Multimedia Networks (WoWMoM 2018) as part of the The Internet of Things: Smart Objects an
Workshop that will be held from 12-15 June of 2018 in Chania, Greece. 978-1-5386-4725-7

The road to privacy in IoT: beyond encryption and signatures, towards unobservable communication

Ralf C. Staudemeyer*, Henrich C. Pöhls†, and Marcin Wójcik‡
*Faculty of Computer Science, Schmalkalden University of Applied Sciences, Schmalkalden, Germany
†Institute of IT-Security and Security Law, University of Passau, Passau, Germany
‡Computer Laboratory, University of Cambridge, Cambridge, UK
Emails: *r.staudemeyer@hs-sm.de †hp@sec.uni-passau.de ‡marcin.wojcik@cl.cam.ac.uk

***Abstract*—Privacy requires more than just encryption of data before and during transmission. Privacy would actually demand hiding the sheer fact that communication takes place. This requires to protect meta-data from observation. We think that this feature is in particular useful and interesting for the Internet-of-Things. However, it requires strong cryptographic security mechanisms, like encryption of communication, to be in place. We motivate the need for strong privacy protection by highlighting privacy issues in a smart home use-case. We advance beyond encryption and discuss which existing techniques can be used to achieve unobservable communication. Then we describe the architecture needed to provide strong protection in this particular use-case. Lastly, we present the building blocks of the architecture we implemented so far on the Re-Mote sensor nodes running Contiki OS and sketch the computational and network overheads imposed by these techniques.**

***Keywords*—Internet of Things, privacy, network security, cryptography, dc-net, mix, contiki, IoT, DTLS, ECDSA, PAN, WSN.**

I. Introduction

Privacy in the Internet-of-Things (IoT) needs more than encrypted end-to-end communication. As privacy is understood as a human right, the threat arising from meta-data collection and analysis must be countered with strong security features. The strongest security feature suitably and cryptographically realisable for local environments is unobservable communication. At least in the legal regime of the European Union (EU) the principle of privacy-by-design [1], [2] and data protection laws play an important role for any information technology enhanced system. From the legal perspective the IoT is not an exception, but it is an exceptional case in its ubiquity and thus in its potential to intrude peoples' private lives. The EUs report on Privacy in the IoT released 2014 [3] shows the increased sensitivity of the topic. The need and awareness within the EU that strong protection is required is also highlighted by the European Commission's support for projects like PANORAMIX[1] that aims to build a mixed networking framework similar to onion routing of Tor. Thus, once you want to offer an IoT service or product in the EU, the data protection rights of the data subject need to be respected as they are guaranteed by EU law. However, also outside the regulatory domain of the EU, we strongly believe that it is a human right to have strong protection of personal data.

Having privacy means that you gain the ability to prohibit the leakage of information to unauthorised third-parties. As a first step, encrypted and authenticated channels technically ensure that only authorised parties are able to read a message's payload during transmission. They found their way into standards in the IoT-domain, e.g., Datagram Transport Layer Security (DTLS) [4], [5] (see Section V-B for more details). However even assuming DTLS or the like is enabled, and HP's recent report on IoT security finds just the opposite, i.e. that "70 percent of devices used unencrypted network service" [6], meta-data still leaks details about the communication. It is very hard to estimate to what extent meta-data can be gathered and utilised by network traffic analysis. Among other things, meta-data includes information like communication endpoints, message timing and location details of the communication. When combined with a-priori knowledge, and processed by machine learning algorithms, extracted information can be so rich that end-to-end encryption can be bypassed. For example it might not be necessary to decrypt the payload at all, because its content can solely be derived from network traffic. This strongly demands an additional layer of privacy protection to prevent the leakage of sensitive information from meta-data.

To counter traffic analysis we need to minimise any kind of information leakage due to communication meta-data and content-data to the same extent. Therefore the network communication property we aim for is unobservability. This property ensures that messages and random noise are indistinguishable from each other. In terms of network nodes it ensures that their activity goes unnoticeable and that messages cannot be correlated. It is a very powerful property ensuring unlinkability, unidentifiability, and a continuous flow of dummy traffic.

"Truly smart gadgets should have built-in intelligence". [7] In this work we show how to use these smart devices to build networks with very strong security and privacy properties. Our contribution is twofold, first we show the applicability of ideas and existing mechanisms for unobservable communication to improve privacy in the IoT, second we present an estimation of the overhead that a truly private IoT would induce. In the following, we discuss the related works and motivate the need for privacy within the SmartHome use-case (Section. III). Then, an analysis of existing solutions and their suitability for unobservable communication in the IoT (Section IV). We finally present our estimations of the overhead needed (Section V).

II. State-of-the-Art

Bandyopadhyay and Sen [8] identify security and privacy as key technologies that will enable IoT. The authors point

[1] panoramix-project.eu (accessed 11 March 2018)

out that there is a lack of privacy preserving technologies available for IoT environments. They identify anonymity networks as a potential basis to implement privacy in IoT. But anonymity networks still require significant resources in terms of computing power and bandwidth. The work published by Miorandi, Sicari et al. [9] provides a survey on the key issues related to the development of IoT services and applications. The authors identify data confidentiality, privacy and trust as key challenges in IoT security and suggest addressing privacy issues in the system design phase, but admit the lack of a privacy framework tailored for IoT. An analysis of security challenges in distributed IoT environments is provided by Roman, Zhou et al. [10]. The authors argue that potential threats and attackers need to be modelled first. Security challenges identified and discussed are identity and authentication, access control, protocol and network security, privacy, trust management, governance, and fault tolerance. The authors conclude that the heterogeneous nature of IoT increases the complexity of most security mechanisms.

This need for privacy(-by-design) is acknowledged by the European Union (EU) [1]. The EU Article 29 Working Party released a list of recommendations to increase privacy of IoT deployments [3]. Among other things they recommend that "Device manufacturers should limit as much as possible the amount of data leaving devices" [3]. The latter is referring to the minimisation of information inside the payload, and they suggest aggregation of data. The EU-funded project RERUM[2] that developed a framework will allow IoT applications to consider security and privacy mechanisms early in their design phase. This works towards a configurable balance between reliability (requiring secure, trustworthy and precise data) and privacy (requiring data minimisation for private information, like location) [11], [12], [13]. All EU-funded projects working on security and privacy in the IoT (IERC AC5) reported jointly in January 2015 that there is currently no research project that tackles anonymising the traffic in networks used for IoT applications [14, p.70].

III. Use-Case and Privacy Problems

In the following, we are going to describe a smart home scenario. The reasons for the choice of this scenario are manifold. First, it is an IoT scenario that is in its effects directly visible in our daily lives. Second, it is most likely to happen on a broad scale in the near future. Many products, e.g. "smart" light bulbs, are already sold today. Development frameworks are maturing, for instance the operating system Contiki[3] or Android Things[4] or visions have been expressed[5][6]. Third, it has also been selected as a scenario by research projects in the smart city domain, e.g. RERUM or CITY PULSE[7]. Last but not least, it allows us to highlight privacy problems in a comprehensible manner, e.g. the real life implications are clearly visible when assuming a burglar is the attacker.

[2]ict-rerum.eu (accessed 11 March 2018)

[3]contiki-os.org (acc. 11 March 2018)

[4]developer.android.com/things/index.html (acc. 11 March 2018)

[5]Corning's Day Made of Glass youtube.com/watch?v=6Cf7IL_eZ38 (acc. 11 March 2018)

[6]Panasonic's Wonder Life-BOX 2020 panasonic.com/global/corporate/center/tokyo/floor/lifebox2020.html (acc. 11 March 2018)

[7]ict-citypulse.eu/scenarios (acc. 11 March 2018)

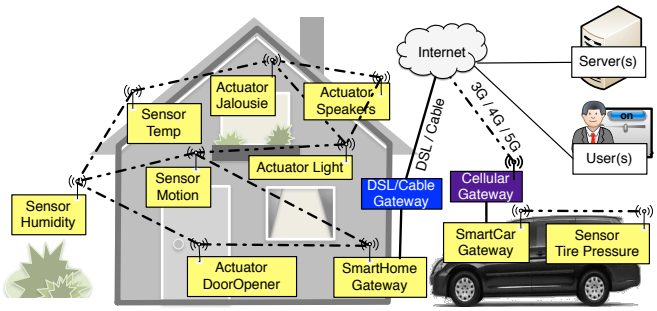

Fig. 1. Example devices in a smart home

Imagine a smart home, which consists of a multitude of sensors and actuators. The latter allow us to influence the physical world (e.g. a door-opener or a light), while the former observe the physical surroundings (e.g. a wearable heartbeat sensor or a motion detector). Most of them interact wirelessly. Some sensor-actor relationships can be tolerant of greater delays, like the communication with the garden's humidity sensor for the lawn. Not all functionality is provided locally only. Access networks like digital subscriber line (DSL) or the mobile cellular network (e.g. 3G) are used to connect via the Internet to servers and users with their applications.

It is obvious that IoT devices will gather information about the home's inhabitants. To protect confidentiality, for instance, a sensor could encrypt readings with the designated server's public key, such that the data can only be decrypted by this selected server; assuming the data owner gave his/her consent to share this information to that server. Our motion detector could encrypt the "detected_motion" message for the smart home gateway, which in turn would encrypt and sign the command "turn_on" for the light. However, encryption of the payload does not prevent activity within the house from being observable. For instance the detection of movement of the homeowner is observable for an attacker eavesdropping on the communication. To monitor messages within a house or vehicle the attacker needs to eavesdrop on a user-controlled network of sensors, actuators, and the gateway. This assumes that messages are not traversing non-private access networks, like DSL, 3G or the public Internet in general. The attacker needs to access the local network. In Fig. 1, the attacker would either need to be near the house to eavesdrop the wireless transmission, or the attack could be carried out by a trojan device (e.g. an attacker that controls a malicious sensor).

However, if "smartness" requires data to be sent to the Internet, for example to request the weather forecast, then the attacker could eavesdrop at many locations. In our smart home example this would be requesting the weather forecast. Point of observation could be not only the sensors, the local network, or the access network, but also the Internet, or the servers providing the requested service. Observation of communication flows, as an attack on privacy, is called traffic analysis. The event driven nature of the communication flows in the IoT also lends those messages to be a good basis for attacks that facilitate pattern analysis. For example, the knowledge that the lights might receive their messages (containing commands) from different sources (switches and movement detectors)

make them become recognised as actuators. The messages that relate to "turn lights on", "get weather forecast" and "translate to speech" always follow in the same order, and occur most likely within a specific time window each day. All this makes the IoT messages, even if their content is encrypted, a very rich hunting ground for meta-data analysis. Next we will discuss some countermeasures.

IV. Private communications

In this section we introduce concepts to counter passive attacks ([15], [16]) based on eavesdropping and traffic analysis into the context of Internet-of-Things (IoT), Private Area Networks (PAN), Wireless Sensor Networks (WSN), and our use-case described in Section III.

A. Existing concepts

A very limited degree of anonymity can be achieved by using a proxy or a Virtual Private Network (VPN). An observer with access to traffic entering and leaving the proxy over extended periods of time can reveal the communication relation. Therefore these solutions fail against a global observer. Fortunately the situation improves by using specific proxy chains. These tunnel encrypt traffic through a number of low-latency proxies.

There are a few basic concepts that offer adaptability to perfect protection against a global observer. One must distinguish between sender, receiver and mutual protection. In this context mutual protection guarantees that both parties of a communication remain anonymous to each other and to any third party. For example, to broadcast or multicast a message would allow recipient anonymity. A summary of anonymous communication systems is provided by [17]. The basic concepts are MIXing [18] and DC-Net [19]. MIXing generates a high degree of anonymity and unlinkability of sender and receiver. It takes a branch of messages and scrambles, delays and re-encodes them in a way an attacker can no longer easily match incoming with outgoing messages. Several adaptations of the MIX-concept were introduced, they add new functions and correct security problems. To obtain the unobservability property, an adaptation of the MIX-concept was introduced in [20]. It includes constant Dummy Traffic and Time Slices [20] preventing an attacker to obtain useful information from packets travelling through the network.

David Chaum introduced in [19] the Dining Cryptographers Net (DC-Net) – a communication protocol that provides unconditional secure unobservable communication. DC-Net is a broadcast-round-based protocol where members of the round can unobservably publish a one bit message per round. This is called "superposed sending" and is very secure but prone to Denial-of-Service attacks. By "superposed receiving" [21] DC-Net was extended to support anonymous receiving of messages. Protections against disrupting nodes were proposed in [21] and [22]. Further, it is possible to categorise concepts into re-routing-based and non-re-routing based concepts [23]. DC-Net and Broadcast (or Multicast) are the only non-re-routing based systems.

In practise we need to consider the attack model and distinguish concepts to protect communication in two different environments: Privacy-Area-Networks (PAN) and Wide-Area-Networks (WAN). The attacker model for this scenario is a global attacker in a local environment, who is computationally bound, but can listen to every communication and can insert arbitrary messages, which is very realistic in a locally limited, wireless scenario. However Denial of Service attacks, such as through radio jamming, are not considered. Given an attacker with powerful enough radio equipment, who is jamming radio frequencies used by the sensor nodes, there is very little chance of preventing this kind of attack.

The protection of the sender gets more difficult when the attacker controls most of the network. A message must get into the network in order to be delivered to the recipient at some point. The concept of "superposed sending" was developed to aid this process. With superposed sending and receiving it is possible to build the DC-Net ([19],[21]), solution for the dining cryptographers problem [19]. The DC-Net protocol is unconditionally secure and permits participants to anonymously broadcast one bit messages. The protocol implicitly assumes the exchange of a one-time pad over a secure channel (sharing unbiased coins) and a reliable broadcast medium. In reality those guarantees are hard to achieve but different optimisations, like tackling the problem of reliable broadcasts, successfully addressed them and were presented in [21], [24]. Other optimisations make DC-Net traffic unobservable by continuous communication of all participants using dummy traffic achieving authenticity, integrity and confidentiality with public key cryptography instead of one time pads. The basic DC-Net-Algorithm is described in [15]. We are aware of only two DC-Net implementations, Herbivore [25] and Dissent [26], [27], probably due to DC-nets sensitivity to disruption. However none of them is aimed towards the IoT and constrained, and embedded devices.

B. Overhead

Communication systems that provide strong security properties like anonymity and ideally unobservability suffer from a high computational and communications overhead. Already the initial key distribution problem, which requires keys to be exchanged with all potential recipients, makes it very expensive. In MIX-networks this concerns MIX-nodes only, in DC-net this concerns all network participants. The computational and bandwidth overhead in MIX-networks is related to the thread model. For increased protection the number of chained mixes and amount of dummy traffic needs to be increased accordingly. A chain length of three running under different entities should make the correlation of sender and receiver sufficiently challenging. The computational overhead is therefore approximately $3 * k$, with k being the number a MIXes per chain. In terms of network traffic, the additional header for every nested message causes on overhead of $8 + 8 + 4 = 20$ bytes per message. The dummy messages are necessary to guarantee continuous flow of traffic, whereas the network benefits from a large number of users.

In contrast to MIX-networks, DC-networks can offer perfect sender and receiver unobservability using one time pads, offering the most protection.[8] Here every participant needs to generate one output (XOR-operation) for every message bit

[8] note: computationally secure with public key encryption

to traverse the network (superposed sending). This makes the overhead proportional to the number of network participants, since each output needs to be delivered to all other participants (using a reliable broadcast operation). This results in $n(n-1)$ bits to be processed per 1 bit message, with n being the number of network participants. For a high number of participants this gets very expensive, and gets worse depending on packet size and network collisions. To worsen the situation both networks require self-organisation and benefit from dummy traffic to ensure a continues flow of traffic. In contrast to MIXing, in DC-net an increase in the number of messages will also cause an increase in collisions. Thus, taking care of collision handling adds to the overhead for DC-net.

V. Towards implementing the Use-Case with 6LoWPANs and Contiki

In the following, we describe the underlying state-of-the-art technologies for embedded devices. This is essential to build the architecture for our use-case. We present all the security mechanisms needed to reduce information leakage and aspire towards unobservable communication, and finally present our feasibility study. In [13], [28] we document two higher layer views of the security and privacy issues to be addressed in the IoT in a SmartCity context, which we take into consideration. Essential building blocks are available open source [29], [30] and we proved them portable to the Re-Mote [28], [31], [32], [33]. Missing system modules were implemented by us in the EU-funded project RERUM[9] [11], [34] by ([32], [33], [35]). Within RERUM the performance gains of ECC signatures and DTLS were accessed, both qualitatively and quantitatively, to identify potential issues in the software and hardware modules of the Re-Mote [31].

A. Architecture

For our use-case scenario we use the 802.15.4 for low-power wireless communications [36]. For those networks, the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [37] is the de-facto standard for routing, whereas the Constrained Application Protocol (CoAP) [38] is the IETF's protocol recommendation in order to realise the RESTful architecture for constrained environments.

We use IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) [39], RPL [37] and CoAP [38] in our studies. This selection implies that devices are capable of handling the firmware with all above-mentioned stacks and protocols. A suitable firmware is the Contiki open source embedded Operating System (OS)[10] [40], [41]. Contiki is a lightweight operating system designed for Internet-of-Things, and the restrictions and needs of constrained devices in mind. It officially supports numerous available wireless sensor platforms, including our target platform, the Re-Mote[11] [42] device.

The Re-Mote houses a CC2538 System-on-Chip (SoC) from Texas Instruments. The platform can run Contiki with the 6LoWPAN/RPL/CoAP software stacks and protocols. In our architecture [11], [34] the gateway provides access to the outside network (e.g. Internet). At the same time, it is the main component of the wireless mesh formed among Re-Mote devices and the gateway. Sensors will also use this wireless mesh to transmit their measurements using CoAP over IPv6.

In order for our use-case to be implemented using Contiki and those networking technologies mentioned just now, the following needs to be done:

- All embedded devices and sensors must be equipped with an 802.15.4 low-power radio interface.
- All embedded devices need to be powered by Contiki, with 6LoWPAN/RPL networking enabled.
- The gateway (Fig. 1) will also get a 802.15.4 interface.
- The gateway creates a 6LoWPAN/RPL network and advertises its presence over the 802.15.4 interface.
- Embedded devices will join this network and, as a result, an 802.15.4 wireless mesh will be formed among those devices and the gateway.
- Sensors use this wireless mesh to transmit their measurements using CoAP over IPv6.

Our selected candidate for the implementation of such a firmware is the Contiki open source embedded OS. Contiki officially supports numerous commercially-available wireless sensor platforms, but many more are also supported unofficially. It features a standards-compliant embedded TCP/IP implementation and supports a number of specifications aiming to optimise the use of TCP/IP networking in embedded devices. This list includes all standards and specifications mentioned above, including 802.15.4, 6LoWPAN, RPL and CoAP.

B. Security considerations

In this section, we present an overview of some of the security mechanisms applicable to the smart home networks outlined in the previous section. This includes a discussion of physical security, hop-by-hop security between neighbouring devices, security of routing control messages, end-to-end security, ensuring authenticity and integrity with signatures, and privacy-enhancing technologies using overlay networks.

Physical layer considerations: The original 802.15.4 standard specifies that wireless meshes will use the 2.4 GHz frequency band at a 250 Kbps bit-rate. However, the 802.15.4g amendment [43] defines alternative physical layers and provisions for wireless operation in different frequency bands, such as at the 863-870 MHz band for Europe. This means that operating at a lower frequency, two devices can now communicate at much greater distances (magnitude of a few kilometres). The amendment also defines various bit rates, ranging from 2.4-500 kbps and as well longer frame sizes. The extended distance, if not bounded by devices, means a greater risk of eavesdropping and interception by an external adversary. But then, new rates and longer frame sizes also give an extra flexibility to build networks with the required size and properties.

[9] ict-rerum.eu (last acc. 11 March 2018)
[10] contiki-os.org (last acc. 11 March 2018)
[11] zolertia.io/product/re-mote-professional-pack/ (last acc. 11 March 2018)

Hop-by-Hop security: The aforementioned 802.15.4 standard specifies security services, which aim to protect the communication between wireless devices on TCP/IP Model Layer 2. To that end, the standard specifies that all the security services use the Advanced Encryption Standard (AES) algorithm with 128-bit keys. The standard permits group keys, i.e., a common key used by a group of nodes (devices) mainly for multi-casting and broadcasting. As such a shared group key provides protection against outsider nodes, but not against malicious insider nodes sharing the same key. The standard defines eight different security suites, which can be used to provide various combinations of confidentiality, integrity and origin authentication. Securing communication on Layer 2 comes with significant performance cost, unless the key is shared by the group. Because then each intermediate node has to perform re-encryption of every single frame using the next-hop node key.

Security of routing control messages: For 6LoWPANs, the de-facto standard routing protocol is RPL. It is a distance vector protocol, which perceives the 6LoWPAN network as a tree-like structure called a Destination Oriented Directed Acyclic Graph (DODAG). Data traffic in an RPL network can flow upwards in the tree (from a node towards the root), while support for downward flow of data traffic is optional. For the protection of routing control messages, RPL uses AES-128 CCM as its underlying cryptography algorithm and MAC values can be either 32- or 64-bit long. The RPL specification also discusses support for signed messages, using a scheme based on RSASSA-PSS [44] with 2048- or 3072-bit moduli. Mechanisms defined as part of the RPL specification can only be used for the protection of RPL control packets, but not for application data.

End-to-End security: One option to ensure the end-to-end security in IoT is to use the DTLS protocol ([4], [5]). DTLS is the modified version of a well-known and widely-deployed cryptographic protocol called Transport Layer Security (TLS). In contrast to TLS, which requires a reliable TCP channel to establish and carry on a secure communication, DTLS can be used over unreliable protocols, such as the User Datagram Protocol (UDP). The latter feature, i.e. a possibility of deployment over UDP, makes DTLS a good candidate for secure communications between resource-constrained devices. Furthermore, since CoAP is designed to operate over UDP, it is possible to deploy CoAP over DTLS.

We developed a research prototype for Contiki and the Re-Mote platform. Our devices are enabled to establish integrity and confidentiality at the transport layer level, including origin authentication [31], [45].

Authentication and integrity protection: The use of per device public keys for confidentiality protection by encryption allows the sender to encrypt data using the intended recipient's public key. The recipient of such a message has no way of verifying the sender's identity. Digital signatures can be facilitated to provide strong entity authentication. Once the data is signed the recipient of such a signed message can verify who signed the message and can use the information as a basis for its access control. In Fig.1 the signed message could include the command 'turn_on', which is now verifiable under the SmartHome Gateway's public signature verification key. Recently we presented a proof-of-concept implementation of ECDSA digital signatures using MicroECC [30] for Contiki on the Re-Mote platform for data integrity and data origin authentication (which supports non-reputation)([32], [33]). Our prototypical implementation on the Re-Mote device is based on NIST curve P160 ECDSA signatures of JSON encoded sensor data[12]. This allows protecting the integrity of data flowing from, to or between constrained devices. It furthermore allows identifying the origin of data by means of public keys. This works on any application level data, allowing a broad use. The protocol and concept of on-device-signatures designs are flexible and allow the use of different cryptographic signature mechanisms [32].

Dining Cryptographers Net: Based on 6LoWPAN and RPL it is possible to create different overlay networks such as VPN and proxy networks, or aforementioned DC-net/Mixing networks. Security considerations in such networks strongly depend on the number of communicating nodes. We consider DC-net [19] as a preferred local option, and using Mix-Networks [18] over the Internet.

Chaum [19] proposed a ring topology, which was extended by [21], where every node receives the message from its neighbours and in order to reduce bandwidth adds its local output directly to the message received. For one DC-net communication round, the message has to travel twice through the ring: the first time is used to get local outputs from neighbouring nodes (sending) and the second one for broadcasting the final, global message to all nodes (receiving). The DC-net implementation Dissent uses a client-server architecture [27], [26], where communication always happens among client-server and server-server, but never among two clients directly. Wireless networks, however, have one big advantage in terms of topology: broadcasts can be done naturally over the physical medium and therefore are cheaper in terms of communication overhead. This assumes that a broadcast message reaches all other nodes without further interaction such as forwarding. The decoupling between the global message and the clients (as per Dissent), cannot be realized in a purely decentralized, fully connected network graph. For example if a single node does not broadcast his local message in a given round, the local output cannot be calculated. This is due to all other nodes in the network depending on it to calculate the global sum.

The first aspect is important to determine the load a constrained node has to cope with. The second aspect indicates the bandwidth and latency requirements that a low-power and lossy network like a wireless sensor network has to offer. We recently compared these two approaches and implemented a prototype [46]. Summarising the costs for individual clients, the star topology proved favourable. However with the down-side that every single node has to cope with one round to build the centre of the star. Nodes in the fully connected network have a constant sending rate of one message per round, but must process $n-1$ messages per round. The receiving costs are thus increasing linearly with number of clients. The total number of messages exchanged in the fully connected network and the ring are linear, whereas the costs are quadratic in the star [46].

[12] github.com/ict-rerum

C. Security features implemented

We now present the security features developed, implemented, and/or ported to the Contiki Operating system running on the Re-Mote.

Contiki: As of October 2014, Contiki provides off-the-shelf support for 802.15.4 security services [47]. Contiki's source tree includes a software implementation of the AES algorithm, but it also provides drivers for some AES acceleration hardware, such as the CC2538 used in the Re-Mote platform.

DTLS encryption: In the RERUM project we discussed the use of DTLS in [34], [11], [45], [28]. Based on [29] we build a research prototype for Contiki and the Re-Mote platform[13]. The experimental results are documented in [31]. We investigated TinyDTLS in great detail with the focus on security and energy consumption in [31]. Our provisional results show that in case of pre-placed keys the total time of handshake execution is around 1.3 seconds, whereas for the raw-public key option is around 137 seconds. The latter case shows significant time overhead to perform the handshake, and thus lower the overall performance of DTLS.

ECDSA signatures: Regarding digital signatures, several implementations for Elliptic Curve Digital Signature Algorithm (ECDSA) exist. From the existing sets of cryptographic libraries, we selected those, which were suitable without significant underlying changes for both: (i) running under Contiki and (ii) running on the ARM Cortex-M3 core. We investigated TweetNaCl (Curve25519 and Ed25519) [48], Piñol (Secp256r1) [49] and MicroECC (Secp160r1, Secp192r1, Secp224r1, Secp256k1, Secp256r1) [30].

In order to run the aforementioned cryptographic libraries on the Re-Mote we ported them to Contiki and adjusted the code whenever necessary [33]. As a container we developed JSON Sensor Signatures (JSS), a JSON format to transport the ECDSA signature over JavaScript Object Notation (JSON) data alongside [35]. We specifically designed JSS for running on constrained devices and we implemented it in [33], [32]. We as well implemented Ed25519 [50] to proof its usability in Contiki([33], [32]). We evaluated the total performance and power consumption loss of sending ECC signed messages on IoT devices to be about 300 % [33]. However, there is still room for optimisation for both the IoT hardware and for Contiki.

DC-Net: With regard to overlay networks providing unobservability discussed in this paper, we bring the DC-Net protocol [19] to constrained node devices. We presented a proof-of-concept implementation and discussed the feasibility of it in great detail in [46]. Here we also integrated some of the improvements suggested in [21], [24]. In contrast to the original protocol we implemented two important optimisations for the message exchange:

Instead of sending 1 bit messages in every message round, we exchange messages with an increased payload of 7 bytes. This is done by carrying multiple message rounds in only one broadcast packet in the "sending vector". The prototype currently uses four message rounds per packet, which is a rather arbitrary number that needs to be optimised in future. Another optimisation is that we exchange multiple messages within a single 802.15.4 frame, in order to utilise the network more efficiently. In total this significant decreases effort for the recipient to reassemble the individual messages from the other member in the network.

In the time of writing our prototype does not utilise a TCP or a UDP stack. Instead it builds upon the RIME [51] stack, which is a network stack of the Contiki OS [40] with very low overhead. RIME itself offers different layers with certain services, but our prototype uses solely the "anonymous broadcast" layer, which offers the required broadcast service. This RIME layer builds directly upon 802.15.4, which permits a node to send 127 bytes at most. Therefore it increases the efficiency in terms of energy consumption and overhead by utilising the packets as well as possible.

Each new node that enters the network has to establish secrets with all other nodes. This can be done using the Diffie-Hellman key exchange. Such key exchange based on elliptic curve cryptography making use of ECDH can be done efficiently with the MicroECC [30] cryptographic library.

VI. Conclusions

Privacy cannot be retrofitted, and here this rule holds even more than in security. Thus, if we want to support sensitive services in the Internet-of-Things, we must act now to put enabling technology in place. So far we implemented most of the building blocks required, this includes DTLS encryption, ECDSA signatures, and DC-Net.

In particular our proof-of-concept of DC-net on the Re-Mote is a novel contribution. Many parameters, like the optimal message size, an efficient key-exchange or the influence of disturbers have to be further considered.The implementation currently relies on reliable broadcast assumption, which will not hold in real world use-cases considering bigger networks like Smart Cities. However, our proof-of-concept shows that strong, privacy enhancing technologies can be adapted to use in the IoT.

VII. Future Work

We see the following steps after our analysis towards a truly private IoT communication algorithm:

Increase the ROM footprint of devices: All the security mechanisms discussed in Section V should be in place, or allow over the air provisioning. But even with the latter combining all of them in the same firmware results in a binary image too large and therefore some needed security mechanisms may get excluded in a real deployment.

Enable layer 2 hop-by-hop encryption within the wireless network: This step prohibits the attacker from reconstructing the network's topology using a very simple wireless sniffer on the RPL routing control packets transmitted in the clear.

Support hardware acceleration: The obvious benefit is that encryption and decryption operations are potentially faster but it can also reduce the size of the firmware image. However, besides the runtime, other aspects like the problem of side

[13] github.com/ict-rerum

channel attacks that MicroECC addresses have to be considered in greater depth.

Carefully consider the use of devices that allow long range communication, e.g., sub-GHz frequency band: It has the disadvantage that the attacker does not need to be so close to eavesdrop. On the other hand it will allow to broadcast messages to a wider audience, which can significant reduce the overhead of the DC-Net protocol.

ACKNOWLEDGMENT

We would like to thank J. Bauer, G. Oikonomou, B. Petschkuhn, M. Moessinger, M. Danish, and A. Moore for their invaluable help during the course of this work. H. C. Pöhls and R. C. Staudemeyer were partially funded by the European Union's FP7 grant n^{o}609094 (RERUM) and the EU's H2020 grant n^{o}644962 (PRISMACLOUD). M. Wójcik was partially funded by the EU H2020 grant n^{o}644866 (SSICLOPS). This paper reflects only the authors' views.

REFERENCES

[1] G. Danezis, J. Domingo-Ferrer *et al.*, "Privacy and Data Protection by Design - from policy to engineering," European Union Agency for Network and Information Security, Tech. Rep. dec, 2014.

[2] A. Cavoukian, "Privacy by Design - The 7 foundational principles - Implementation and mapping of fair information practices," p. 5, 2009.

[3] EU Article 29 Data Protection Working Party (WP 223), "Opinion 8/2014 on the Recent Developments on the Internet of Things," pp. 1–24, sep 2014.

[4] N. Modadugu and E. Rescorla, "The Design and Implementation of Datagram TLS," in *Proc. of the 11th Ann. Network and Distributed System Security Symp. (ISOC NDSS'04)*, 2004.

[5] D. McGrew and E. Rescorla, "RFC5764 – Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)," RFC Editor, Tech. Rep., may 2010.

[6] Hewlett Packard Enterprise, "Internet of Things research study," HP, Tech. Rep. jul, 2015.

[7] M. Vella, "Nest CEO Tony Fadell on The Future of the Smart Home," *TIMES Magazine*, 2014.

[8] D. Bandyopadhyay and J. Sen, "Internet of Things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, may 2011.

[9] D. Miorandi, S. Sicari *et al.*, "Internet of things: Vision, applications and research challenges," *Ad Hoc Netw*, vol. 10(7), pp. 1497–1516, 2012.

[10] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, jul 2013.

[11] H. C. Pöhls, V. Angelakis *et al.*, "RERUM: Building a reliable IoT upon privacy- and security- enabled smart objects," in *Wireless Communications and Networking Conf. Workshop on IoT Communications and Technologies (WCNC '14)*. IEEE, apr 2014, pp. 122–127.

[12] E. Z. Tragos, V. Angelakis *et al.*, "Enabling reliable and secure IoT-based smart city applications," in *Proc. of the Int. Conf. on Pervasive Computing and Communication Workshops (PERCOM'14)*. IEEE, mar 2014, pp. 111–116.

[13] R. C. Staudemeyer, H. C. Pöhls, and B. W. Watson, "Security & Privacy for the Internet-of-Things communication in the SmartCity," in *Designing, Developing, and Facilitating Smart Cities: Urban Design to IoT Solutions*. Springer, 2017, ch. 7, pp. 109–137.

[14] G. Baldini, T. Peirce *et al.*, "Internet of Things: IoT Governance, Privacy and Security Issues," IERC - European Research Cluster on the Internet of Things, Position Paper Activity Chain 05, jan 2015.

[15] J.-F. Raymond, "Traffic analysis: protocols, attacks, design issues, and open problems," in *Designing Privacy Enhancing Technologies*, ser. LNCS, H. Federrath, Ed. Springer, jul 2001, pp. 10–29.

[16] R. C. Staudemeyer, D. Umuhoza, and C. W. Omlin, "Attacker models, traffic analysis and privacy threats in IP networks," in *Proc. of the 12th Int. Conf. on Telecommunications (ICT'05)*, 2005.

[17] D. Kelly, R. Raines *et al.*, "Exploring extant and emerging issues in anonymous networks: A taxonomy and survey of protocols and metrics," *IEEE Commun Surv Tut*, vol. 14(2), pp. 579–606, 2012.

[18] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Comm. of the ACM*, vol. 24, no. 2, pp. 84–90, feb 1981.

[19] ——, "The Dining Cryptographers problem: unconditional sender and recipient untraceability," *J. of Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[20] A. Pfitzmann, B. Pfitzmann, and M. Waidner, "ISDN-mixes: untraceable communication with very small bandwidth overhead," in *GI/ITG-Conference "Kommunikation in verteilten Systemen" (Communication in Distributed Systems)*, 1991, pp. 451–463.

[21] M. Waidner and B. Pfitzmann, "The Dining Cryptographers in the disco: unconditional sender and recipient untraceability with computationally secure serviceability," *Proc. of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT '89)*, vol. 89, p. 690, 1990.

[22] P. Golle and A. Juels, "Dining Cryptographers revisited," in *Proc. of Adv. in Cryptology (EUROCRYPT '04)*, vol. 2729, 2004, pp. 456–473.

[23] Y. Guan, X. Fu *et al.*, "An optimal strategy for anonymous communication protocols," in *Proc. of the 22nd Int. Conf. on Distributed Computing Systems (ICDCS'02)*. IEEE, 2002, pp. 257–266.

[24] M. Waidner, "Unconditional sender and recipient untraceability in spite of active attacks," in *Proc. of Advances in Cryptology (EUROCRYPT'89)*. Springer, Jun 1989, pp. 302–319.

[25] S. Goel, M. Robson *et al.*, "Herbivore: a scalable and efficient protocol for anonymous communication," Cornell University, Tech. Rep., 2003.

[26] D. I. Wolinsky, H. Corrigan-Gibbs *et al.*, "Dissent in numbers: Making strong anonymity scale," in *Proc. of the 10th USENIX Conf. on Operating Systems Design and Implementation*, ser. OSDI'12. USENIX Association, 2012, pp. 179–192.

[27] H. Corrigan-Gibbs and B. Ford, "Dissent: Accountable anonymous group messaging," in *Proc. of the 17th ACM conf. on Computer and communications security (CCS'10)*. ACM, 2010, pp. 340–350.

[28] R. C. Staudemeyer, H. C. Pöhls *et al.*, "Privacy enhancing techniques in Smart City applications," University of Passau, Tech. Rep., 2015.

[29] O. Bergmann, "TinyDTLS: A DTLS open source stack," 2015.

[30] K. MacKay, "micro-ecc," 2016.

[31] G. Papadopoulos, R. C. Staudemeyer *et al.*, "The RERUM Laboratory Evaluation Results," University of Passau, Tech. Rep., 2016.

[32] J. Bauer, R. C. Staudemeyer *et al.*, "ECDSA on things: IoT integrity protection in practice," in *Proc. of the 18th Int. Conf. on Information and Communications Security (ICICS'16)*. Springer, 2016, pp. 1–15.

[33] M. Mossinger, B. Petschkuhn *et al.*, "Towards quantifying the cost of a secure IoT: overhead and energy consumption of ECC signatures on an ARM-based device," in *17th Int. Symp. on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, jun 2016, p. 6.

[34] V. Angelakis, J. Cuellar *et al.*, "The RERUM system architecture," University of Passau, Tech. Rep., sep 2014.

[35] H. C. Pöhls, "JSON Sensor Signatures (JSS): end-to-end integrity protection from constrained device to IoT application," in *Proc. of the Workshop on Extending Seamlessly to the Internet of Things (esIoT'15)*. IEEE, jul 2015, pp. 306–312.

[36] IEEE Standards Association, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," 2006.

[37] A. Brandt, J. Hui *et al.*, "RFC6550 – RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," pp. 1–164, mar 2012.

[38] Z. Shelby, K. Hartke, and C. Bormann, "RFC7252 – The Constrained Application Protocol (CoAP)," p. 112, jun 2014.

[39] G. Montenegro, N. Kushalnagar *et al.*, "RFC4944 – Transmission of IPv6 Packets over IEEE 802.15.4 networks," sep 2007.

[40] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *29th Ann. Int. Conf. on Local Computer Networks (LCN'04)*, 2004, pp. 455–462.

[41] Contiki, "The Open Source OS for the Internet of Things," 2017.

[42] Zolertia, "RE-Mote datasheet," p. 2, Dec 2015.

[43] IEEE Standards Association, *Part 15.4g: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks IEEE Computer*, 2012.

[44] J. Jonsson and B. Kaliski, "RFC3447 – Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," Internet Engineering Task Force, Tech. Rep. 3447, feb 2003.

[45] D. Ruiz, M. Wójcik *et al.*, "Enhancing the autonomous smart objects and the overall system security of IoT based smart cities," University of Passau, Tech. Rep., 2015.

[46] J. Bauer and R. C. Staudemeyer, "From Dining Cryptographers to Dining Things: Unobservable Communication in the IoT in practice," in *Proc. of the Int. Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD'17)*, 2017, p. 9.

[47] H. Soroush, M. Salajegheh, and T. Dimitriou, "Providing transparent security services to sensor networks," in *Proc. of the Int. Conf. on Communications*, 2007, pp. 3431–3436.

[48] D. J. Bernstein, B. van Gastel *et al.*, "TweetNaCl: A Crypto Library in 100 Tweets," in *Proc. of the Int. Conf. on Cryptology and Information Security in Latin America (LATINCRYPT'14)*, vol. 8895, 2014.

[49] O. Piñol Piñol, "Implementation and evaluation of BSD Elliptic Curve Cryptography," Master thesis (pre-Bologna period), Universitat Politècnica de Catalunya, nov 2014.

[50] S. Josefsson and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)," RFC 8032, Jan. 2017.

[51] A. Dunkels, "RIME - a lightweight layered communication stack for sensor networks," in *Proc. of the European Conf. on Wireless Sensor Networks (EWSN'07), Poster Abstract*, 2007, p. 2.