Security Update and Incident Handling for IoT-devices
A Privacy-Aware Approach

Geir M. Køien

Institute of ICT
University of Agder, Norway
Email: geir.koien@uia.no

***Abstract*—There is a fast-growing number of relatively capable Internet-of-Things (IoT) devices out there. These devices are generally unattended and also relatively vulnerable. The current practice of deploying, and then leaving the devices unattended and unmanaged is not future proof. There is an urgent need for well-defined security update and incident management procedures for these devices. Sensible and secure default settings, as well as built-in privacy must also be included. This is particularly important since the devices are managed by non-professionals. This paper presents an outline of a model to cater to these needs.**

Keywords–Internet-of-Things; Smart home; Incident reporting; Security maintenance; Privacy; Security management.

I. INTRODUCTION

A. Background and Motivation

A central assumption in this paper is that IoT devices, whether owned and operated privately or by an organization/company, will be management. The management in question should encompass security functions and sensible privacy setting, and the default settings should be both secure and privacy-respecting. We furthermore assume that the majority of the IoT device owners will be unable to adequately manage and respond to security and privacy requirements.

Web browsers today will quite likely silently install security updates, most operating system will routinely go through security updating procedures as will the smartphone operating system. Some of these require user interaction and consent, while others are fully automated. We postulate that IoT devices will need similar capabilities, and we also believe that these capabilities must be fully autonomous (no user intervention required) since we believe that the largely unattended IoT devices generally cannot rely on user intervention and response.

B. Outline

In this paper, we briefly investigate a class of IoT devices and how these may have full security update management and a minimal security incident and anomaly reporting service. The proposal is still at the modelling stage and the devised model is still work-in-progress. The IoT devices that adhere to our suggested architecture, will feature three information planes:

- User Services Plane (USP)
- User Management Plane (UMP)
- Security Management Plane (SMP)

The services will be realized by a two-tier architecture, separating global and local components, with clear division of authority and assumed trust between them.

The USP and UMP service planes may have cloud-based components, but whatever the case, these planes will have "local" termination with respect to the IoT device. The SMP service will be centralized and "global" in scope.

Privacy is a required property, and our design aim to adhere to the Privacy-by-Design (PbD) [1] tenets. We have therefore taken steps to make the model privacy-aware and privacy respecting, by introducing separation of duties and being particular at what kind of trust is placed in which architectural component/layer.

C. Related Work and Relevant Standards

The field is not yet settled, and the number of papers and proposed standards, of all types, is large and growing. We expect security and privacy to become even more important for IoT in the future

1) Related Work: A few examples.

The survey paper "Security, privacy and trust in Internet of Things: The road ahead" [2] contains a broad overview over the challenges to IoT security. It emphasises that the IoT vision is characterized by heterogeneity, in terms of technologies, usages and application domains. It is also a fast phased and dynamic environment. Traditional security measures still play a large role, but the paper highlights that these are not always complete, sufficient or even appropriate. The authors also point out that scalability and flexibility is essential in this domain.

Another paper which also highlights open issues more than solutions is found in [3]. Also, the authors discusses these and related issues, like vulnerability, threats, intruders and attacks, in [4]. Both papers take a relatively high-level perspective.

In [5], the authors claim that "And as IoT contains three layers: perception layer, transportation layer and application layer, this paper will analyze the security problems of each layer separately and try to find new problems and solutions.". In the end, the authors conclude that IoT devices are more exposed and less capable than other network elements, and that therefore the challenges are both different and more urgent. Trust for IoT devices, both on software and hardware, is discussed in [6].

2) Relevant Standards: There is no shortage of formal standards and industrial standards concerning IoT and security for IoT. The following is an incomplete selected set of standards. There is a bias in the selection towards wireless and cellular communications standards, but this may be well justified as a very large proportion of the IoT devices will have WLAN and/or cellular capabilities built-in.

 ISBN: 978-1-61208-493-0

– 3GPP TS 33.401: 4G Security Architecture

This standard is about the 3GPP 4G security architecture and it encompasses security for the eNodeB (eNB) base (tranceiver) stations (chapter 5.3 in [7]). In a 4G network, to achieve sufficient spatial ($[bit/s]/m^2$) capacity, one needs a densely distributed network of eNB's. There will therefore be a large number of eNB's, and the scenario may be somewhat reminiscent of a managed IoT network. Security for updating and managing the highly distritbuted base stations may be different from many IoT scenarios, but we believe there are many similarities and lessons to be learned here.

– 3GPP TS 33.310: Authentication Framework

This standard [7] specifies, amongst others, roll-out of digital certificates to the 3GPP eNB base stations, using the Certificate Management Protocol (CMP) [8]. This part is highly relevant for IoT devices too, since many of them will indeed be capable of handling asymmetric crypto and digital certificates. Indeed, even the humble SIM card (smart card) is able to do so, and we therefore postulate that this capacity is fully feasible for any IoT device that needs to handle security sensitive data and/or privacy sensitive data. Moore's law also implies that this capacity will only be cheaper over time, and so we fully expect that such capabilities will be commonplace.

– 3GPP TS 33.187: Machine-Type Communications

This standard [9] encompasses security for the so-called Machine-Type Communications (MTC). The standard defines how to allow IoT and machine-to-machine (m2m) devices be connected to a Service Capability Exposure Function (SCEF). Specifically, TS 33.187 requires "integrity protection, replay protection, confidentiality protection and privacy protection for communication between the SCEF and 3GPP Network Entity shall be supported" (Chapter 4.1 in [9]). These aspects are important for all IoT devices and this standard may serve as design input for non-3GPP cases too.

– GSMA CLP.11: IoT Security Guidelines Overview

This document [10] by the GSM Association is a non-binding guidelines document, and is as such not a normative standards document. It may still be quite influential since the GSM Association does have great reach within the community of cellular operators and vendors. The document identifies a set of grand challenges for IoT, and the then proceeds to propose possible solutions. The challenges listed are:

A) Availability
B) Identity
C) Privacy
D) Security

Provisioning of scalable and flexible identifier structures is at the heart of the problem. Similarly, availability and security normally presupposes that the entities (the IoT devices) can be identified. Privacy then adds to this, but presupposing strong security [1] *and* requiring that the long-term identifiers are never exposed in clear (amongst others).

The document pays considerable attention to life-cycle aspects issues. The document also includes a chapter on risk assessment, an aspect which is all too often neglected in standards documents. Would-be IoT system designers are well advised to take this document into consideration. The document seems inspired by the "assumptions must be stated" idea, in a similar vein to the "Prudent Engineering Practice for Cryptographic Protocols" [11] paper. We strongly approve of the need for being explicit about assumptions and conditions.

– NIST: Draft Cyber-Physical Systems (CPS) Framework

The NIST "Framework for Cyber-Physical Systems" document, current in public review draft status, is an ambitious document which is expected to have considerable influence over future products [12].

D. Paper Layout

In Section 2, we provide a high-level problem description. This includes the main aspects and high-level requirements. In Section 3, we continue our investigation with a focus on underlying assumptions and premises concerning the devices and the detailed security service needs. In Section 4, we provide an outline of the proposed security management plane model. Here we outline the logical planes, network components and interfaces. In Section 5, we discuss the achievements and in Section 6 we round off with a Summary and Conclusion.

II. HIGH-LEVEL PROBLEM DESCRIPTION

A. Device Capabilities

A typical mid-level IoT platform these days would be based on the ARM Cortex family of processors. Here we have the relatively powerful ARM M4 processor (w/floating point and DSP functionality), being both very affordable and surprisingly power efficient [13]. These devices typically sport 32-256KB SRAM memory and up to 1GB flash memory. We assume a device of roughly this capability in our design.

B. Lightweight, Minimality and Modularity

The core IoT architecture should be lightweight, including the base protocols. Correctness and efficiency is likely to benefit from this. Basic security and privacy functionality must be included in the core architecture.

Extensibility and additional features will be needed, and this must be designed to be modular. Restraint in adding features is necessary, but is clear that any successful architecture will over time grow more complex and encompass new areas [14]. We advocate a design reminiscent of the microkernel approach to operating systems design [15], in which only a minimal set of functional are at the core, running in supervisor mode, and where other component may be added and where strict rules concerning use of well-defined interfaces and protocols are adhered to. This will, amongst others, facilitate security hardening and it will enable the systems to be deployed on less capable devices.

C. Connectivity and Exposure

Commonly the devices will have bluetooth low energy connectivity, WLAN connectivity or even fixed LAN or cellular access. That is, they are reachable over the internet. This also exposes the devices to a whole range of threats, and whenever a device, or a class of devices, gains popularity they are prone to become a target. It is therefore prudent to assume that our IoT devices will, sooner-or-later, become targets.

 ISBN: 978-1-61208-493-0

D. Scalability

Needless to say, any solution that must be able to cope with a large, and fast growing number of devices, must be scalable. That is, the cost model for adding devices/users must be linear and with a low constant factor. The upper limit on the number of devices must be very high as to not prohibit future growth.

E. Explicitness

As a rule, all requirements, including the security and privacy requirements must be explicit. Also, all conditions and premises must be made explicit. Explicitness is also a main lesson from [11] (being essential to Principles 1, 2, 4, 6, 10 and 11 in that paper).

F. Security and Privacy Requirements

Due to the exposure, the devices will need security protection, security supervision and security updating to remove, reduce and mitigate the risks. The devices will need basic capabilities regarding device integrity assurance, and for handling entity authentication, data confidentiality and data integrity.

It is quite likely that the devices will capture, store and transmit privacy sensitive data. We therefore require that a PbD regime should be adhered to [1]. As noted in [16], [17], PbD does not come about all by itself, and considered and careful design, implementation and maintenance is required to create credible privacy solutions.

G. Automation and Autonomy

We cannot expect that the end-users will provide security management for the devices. In fact, the end-user may increasingly be unaware of the presence of the IoT-devices. Effective security management of unattended and highly distributed devices will necessarily have to be automated and autonomous.

H. Challenges

As already mentioned, the GSM Association has recognized four main challenges created by IoT: *availability*, *identity*, *privacy* and *security* [10]. An autonomous security update and incident management system will need to address all these aspects, and provide at least a partial solution to the security aspect. We note that strong security is effectively a prerequisite for availability and privacy.

Trust and trustworthiness are essential elements and even prerequisites for widespread IoT adoption. Trust is a complex matter [6], but suffice to say that credible security management should instill confidence and thereby trust. Trustworthiness is hard to prove, but good security management should provide a measure of assurance.

I. Scope

The proposal made in this paper is an architectural proposal concerning security updating and incident and anomaly reporting. The proposal is, however, not a proposal for a fully fledged architecture, but rather for an architectural component. The proposal may therefore be compatible with other IoT architectures, but may of course also overlap with them or even be at odds with them.

III. Assumptions and Premises

This paper makes a few assumptions about the IoT devices.

A. Internet Connectivity

We assume that the device is connected to the Internet.

B. Hardened OS

The OS is assumed to be hardened. Hardening is also assumed to be carried out when the OS is compiled and built with the program, as is often the case for embedded devices. Unnecessary protocols and services must removed or disabled, and only a minimal set of software be present. A local IPtables firewall may be deployed.

C. Security Capabilities

The devices are assumed to have a trusted platform module (TPM), with basic crypto processing support and secure storage. Preferably, they adhere to standards such as ISO/IEC 11889-1:2015 [18]. A vendor issued device certificate is assumed to be available, or some similar identification that may be used for bootstrapping the CMPv2 protocol [8].

In late 2015, ARM released the ARMv8-M architecture, which is the new baseline Cortex-M architecture [19]. It introduces support for ARM's TrustZone TPM for the Cortex-M processors, and is as such an important step towards credible security for IoT devices. As of yet, there are no commercially available designs, but it is expected that there soon be a plethora of available processors targeted for the security sensitive IoT markets.

D. Power, Processing and Memory Capabilities

The device may have limited capabilities, but we shall assume that the device is not too restricted. That is, we assume it to be roughly at least as powerful as the lower end of the ARM Cortex M3/M4 processor families.

E. Secure Bootloading and Software/Firmware Attestation

A secure bootloader is necessary, and it will likely be using TPM functionality. All software, including firmware and patches, must be signed. All software packages shall have version numbers, and this includes firmware and patches. A TPM may facilitate attestation, but alternatives exists [20].

F. Device Recovery

The device shall feature a secure loader, which facilitates a basic boot strap procedure that can securely rebuild the device software. We expect this to be part of the TPM functionality.

G. Device Identifier

The device must have a unique device identifier. This identifier is assumed to be used in the device certificate, but we shall otherwise be agnostic about the nature of the identifier. The device may also have, or use, higher-layer identifiers, but this is considered outside the scope of this contribution. An example would be a dropbox account identifier.

The device may also have network addresses and cellular identifiers. These *may* uniquely identify the device, but we do not in general consider these to be appropriate for identifying the device (observe the *explicitness* rule).

 ISBN: 978-1-61208-493-0

H. Identifiers and Privacy

A fundamental part of privacy is that there is sensitive data that is linked to a person. If one can break the linkage between the person and the sensitive data, then leakage of the data would not necessarily be (privacy) critical.

We must assume that an intruder will be able to link plaintext device identifiers with the person(s) associated with the device. This capability is after all the core business for enterprises like Google. Consequently, we must assume that the intruder will be able to correlate unprotected data.

It is thus necessary to conceal the permanent device identifier such that no outsider will be able to associate the device identifer with the device or the user/owner. There are several ways to do this, including those described in [21], [22]. The functional split between the global and local services are very much reminiscent of split found in the cellular networks, where the local component necessarily must know the location and where the central component must necessarily know the permanent identity. Here, it has been shown that with proper setup one may achieve both location- and identity privacy [23]. In this paper, we shall ignore the specifics, but we do require that identifier and location privacy is part of the design.

IV. Outline of the Security Management Plane Model

Figure 1 depicts an outline of the Security Management Plane (SMP) model. We have already introduced the logical planes, but shall now take a closer look at how they are arranged. We shall primarily investigate the SMP plane and the associated services.

A. Trust Assumptions and Trust Relationships

We have the following principal entities in our model:

- **USER:** The user and/or owner of the IoT-device.
- **LOCAL:** The local SMP component.
- **GLOBAL** The global (centralized) SMP component.

We assume that the USER is an entity entitled to privacy protection according to the local laws. The GLOBAL entity is assumed to be operated by the IoT device manufacturer or some entity operating on behalf of the device manufacturer. It may also be operated by the software manufacturer. This would be similar to patch update services operated by Microsoft, Google and others.

The LOCAL entity is assumed to be operated by a local entity, perhaps a local branch of the IoT manufacturer or some authority which is legally responsibly, warranties etc., for the IoT devices. It is required that the LOCAL and GLOBAL entities strictly observe the SMP model with regard to information exchange. We have observed that in the post-Snowden era, local authorities have increasingly required critical services to be hosted locally. We therefore have reason to believe that similar requirements may surface for IoT-devices too, or that such services are seen as commercially important to reassure the end-users (building confidence and perceived trustworthiness). We have the following trust assumptions:

- **USER vs. LOCAL**

  The USER trust LOCAL with respect to provided services. This is an asymmetric dependence trust.

- **LOCAL**

  The LOCAL entity must have security trust in the GLOBAL entity. The LOCAL entity shall not trust the GLOBAL entity with respect to USER privacy. The LOCAL entity cannot fully trust the USER. The LOCAL entity trust the incident- and anomaly reports, but do not place high significance in individual reports.

- **GLOBAL**

  The GLOBAL entity trust the LOCAL entity with respect to security, but not blindly so. The GLOBAL entity trust the incident- and anomaly reports, mediated by the LOCAL entity, but need not trust any single report and/or report from any single device.

B. The Logical Planes

1) The User Services Plane (USP): USP consists of the data associated with services provided by the IoT-device. We shall not be further concerned with the USP in this paper.

2) The User Management Plane (UMP): UMP consists of the device setup and configuration services provided by the IoT-device. The UMP is specifically about setting up the device end-user functionality. It does not cover basic security or privacy related setup or configuration. The data *may* be privacy sensitive, and the design must reflect this. We shall not be further concerned with the UMP in this paper.

3) The Security Management Plane: The *security management plane* (SMP) is the crux of this paper. It consists of:

- Security setup and configuration
- Security update functionality
- Security incident and anomaly reporting, including local aggregation
- Secure restore functionality
- Identity- and Location Privacy handling

There will be a division of labor:

- Local SMP handling
- Centralized SMP handling

This will facilitate privacy and provide geo-distributed services. Localized processing may easier satisfy national regulatory requirements, while centralized analysis and handling of incidents will provide scalability and efficiency benefits.

C. The Network Components

The division or labor implies a LOCAL component and a centralized GLOBAL component. We observe that the local component will need to have provisions for geographical assurance. Implementation-wise, it will be a matter of policy if there is a need to comply with jurisdictional and regulatory requirements that dictate location of the local SMP handling.

1) The Central/Global SMP Component: The central security update and incident management control function will facilitate both security update production and distribution, and security incident and anomaly analysis.

This function does not need to know the device identifiers, nor does it need to know the associated IoT-device owner or user(s). It may need to know the software version status and any report on incidents and security anomalies associated with

 ISBN: 978-1-61208-493-0

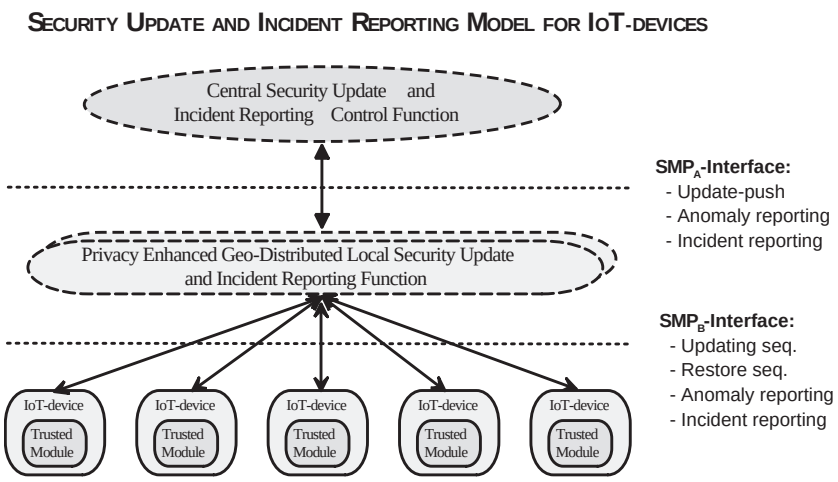

Figure 1. Outline of the Security Management Plane Model.

the devices. For the purpose of the incident analysis, we restrict this function to know the device class and the identity of the local SMP handling component. The true device identifier must never be divulged to the central SMP component.

2) The Local SMP Component: This function handles interactions with the IoT-devices within its geographical coverage area. We expect this area to coincide with regulatory or jurisdictional borders.

The IoT-devices will communicate with the local SMP component. The local component will therefore know both the IP-address and the device identifier. The IP-address may be concealed if one uses Tor services [24], but the device identifier must be known to the local SMP component.

The local SMP component will communicate with the central SMP component, and it will receive protected security patches and software packages from the central SMP component. The local SMP component will aggregate and anonymize incident- and security anomaly reports from the IoT-devices before forwarding them to the central SMP component. The local SMP component may use temporary synthetic alias identifiers for a device, if there is a need for device references. This identifier must never be allowed to become an emergent identifier, and it must be fully de-correlated from the true device identifier. The de-correlation must be complete with respect to the full context given by the message exchange.

D. The SMP-Interfaces

1) The SMP_A-interface: This is a fully authenticated and security protected interface between the local SMP component and the central SMP component, as depicted in Figure 1.

2) The SMP_B-interface: This is a fully authenticated and security protected interface between the IoT-device and the local SMP component, as depicted in Figure 1.

3) Realization: The abstract SMP protocols should be agnostic about the underlying security transport protocol. Suffice to say, that strong security and credible privacy must be assured. The ENISA recommendations for cryptographic protocols, algorithms and key lengths provides good advice in this respect [25], [26]. ENISA is an EU agency, and the recommendation therefore carry some significance.

E. The SMP Services

1) Security Update – Local provisioning: One can have both push and pull mechanisms for security updates, but for IoT devices we do not generally recommend push solutions since it probably require more resources from the device. Push solutions may of course be appropriate for zero-day vulnerabilities, but scheduled pull solutions would likely suffice for patches that are less urgent and less critical. The scheduled pull frequency should reflect the security policy for the particular device class and according to usage, availability, etc.

In either case, signed security updates will be received by the IoT device. All updates must be numbered, and the device will log the date/time and update number before implementing it. The local SMP shall not maintain logs about device status unless required to do so by the IoT device.

2) Security Update – Central provisioning: Whenever a security update patch is produced, the central SMP component will distribute the security update to the local SMP components. We recommend update frequencies to reflect the common vulnerability scoring system (CVSS) [27], although the CVSS system has been criticized for not properly reflect IoT devices [28]. The normal "serious vulnerability" score of 7 may therefore not properly reflect IoT concerns.

3) Incident- and Anomaly Reporting: Security incidents and anomalies are detected and reported by the TPM. This information is used by the SMP components to uncover large scale attacks and emerging attack trends. The ENISA publication [29] provides valuable guidance as to EU regulatory input on incident reporting.

4) Local Incident and Anomaly Reporting: This service will include software status, including patch levels etc. The device identifier is part of the security context, but should not be part of the incident/event report itself. A synthetic referential identifier may be provided by the local SMP.

It may, subject to authorization, be beneficial to store the incident history of the devices at the local SMP. This may

 ISBN: 978-1-61208-493-0

allow the local SMP to detect if certain devices are specifically targeted. If so, one may speculate that the IoT device is an advanced persistent threat (APT) target. This in turn may trigger increased supervision and alarms.

5) Central Incident and Anomaly Reporting: The local SMP component will forward incident reports to the central SMP component. The local SMP component shall take steps to replace identifiers, if any, such that the central component never learns the true device identifier behind a reported incident. The local component *may* aggregate certain events and may delay reports to provide further de-correlations.

6) Device Attestation: The IoT device may request attestation services from the local SMP component. This service will need to be based on TPM functionality and permitting the local SMP component to survey the state of the IoT device. It may be part of a forensics service or a device recovery service.

7) Device Recovery: The IoT device may subscribe to recovery services at the local SMP component. As a minimum the local SMP should provide services to restore the device to a pristine condition, with all recent security update patches being implemented.

8) Device Backup: The local SMP component may provide a secure backup procedure, covering all or selected data data elements. The device backup data should be encrypted and protected by the TPM, using unique device specific keys. Only the TPM should be able to restore the backup data.

9) Device Decommissioning: Life cycle considerations implies that one will need an explicit way of clearing all information on the target device. This will in effect clear all data and restore initial factory settings. This procedure must be resilient enough to withstand efforts from ordinary forensic tools to restore the information. The procedure may be triggered by a request via the local SMP component. The TPM should be responsible for carrying out the task.

V. Discussion

This paper describes an outline of an architectural component. Quite a few of the characteristics described below cannot be fully judged on the basis of the outline.

A. Lightweight, Minimality and Modularity

Our architectural component outline is both lightweight and relatively minimal. It is also modular, in the sense that it will build upon basic identifier structures and cryptographic capabilities, and delivers higher-level services.

B. Explicitness

This is related to requirements and conditions, including preconditions and postcondition. Essentially we have a "Mean what you say and say what you mean" situation. Use of formal methods may help verifying that captured requirements are adhered to, but these tools cannot in general help out with the "capturing" part. Explicitness must be enforced in any further development of the architectural component and in any implementation.

C. Scalability and Exposure

The division into a local-global split will facilitate scalability, as well as improving error resilience and thereby improving availability. Exposure is a necessary evil, but conscious design and appropriate use of cryptographic protocols can significantly reduce the unwanted effects of exposure.

D. Security and Privacy

The concrete security mechanisms is not specified in our proposal. Hence, more work is needed here for a concrete realization. However, there is no grand challenge here, only work that must be done precisely and consistently. Identity privacy and unlinkablity is mainly addressed through the local-global functional split. Data privacy is primarily by means of encryption. The requirements for the split is important, and schemes and measures that enforce the split must be encouraged. It would seem prudent to have this as a contractual requirement, and local regulatory requirements may also be an instrument in enforcing the functional split. Still, in the end, there must also be an economical incentive to manage and run both the local and the global infrastructure.

How credible is the privacy?

Clearly, it depends on the split between the local and global component being fully respected. There exists other solutions that would avoid this. These would be *privacy-preserving* and tend to be based on secure-multiparty computation and/or homomorphic cryptography. However, as argued in [30], strong irrevocable encryption may in the end provide less security and privacy. Governments are claimed to act a long the lines of "If we cannot break the crypto for a specific criminal on demand, we will preemptively break it for everybody." [30]. So, privacy must be balanced and possibly revoked, and this is achieved in our proposal.

E. Challenges: Availability, Identity, Privacy and Security

"Identity" is the only aspect that has not been addressed by our proposal. That is, we have identified this as a building block that our proposal depends upon.

F. Scope and Completeness

The scope is limited to a high-level model. Within the scope the proposal is reasonably complete, but there are many parts to be resolved, and the details have not yet been fully worked out.

G. Further Work

The model presented is an architectural component of a security architecture. Further work is needed to fit this component into a complete architecture. In particular, the concrete implementation of the security requirements should be aligned to the use in other areas. This is particularly relevant for identifiers and for basic services such as entity authentication, and integrity and confidentiality services.

Key agreement and key distribution must also be addressed and aligned to the overall security architecture. Preferably, one also wants to have a well-defined, effective and efficient security protocol to be the backbone of the services. As of today, one is often advised to use the Transport Layer Security (TLS) protocol [31] or the IPsec security protocols [32]. However, these are poor choices for IoT, and TLS are also broken [33], [34], and should probably be phased out. That is, a dedicated, effective and efficient privacy-aware security protocol will probably have to be designed for this purpose. This will be a difficult task, but following advise from [11], [25] and applying state of the art tools, it is also clearly doable. Privacy, if it is to be credible, must be strongly aligned and be consistent over the full architecture to avoid leakage of sensitive data.

 ISBN: 978-1-61208-493-0

Smart metering or remote home monitoring would be examples of IoT systems that could benefit from the capabilities of the model. As such they would make good candidates for a pilot implementation to feature the model architecture.

VI. Summary and Conclusions

In this paper, we have identified the need for autonomous security update and incident/anomaly reporting for IoT-devices. In particular, we have addressed relatively capable IoT devices that ordinarily will be unattended devices, very much in line with a significant segment of the smart home devices.

This paper has provided a rough outline of a model in which IoT security update and incident handling is separated from normal user functionality, including user functionality setup and configuration. We believe that this is necessary since security management is becoming too complex to handle for end-users, and that the consequence of not managing security will be too severe. The current deploy-and-forget regime does not play out well for security functionality.

We have also provided a model in which there is a clear distinction between the centralized function and the local function. The main benefits of this arrangement is that one can more easily adhere to local regulatory requirements and one can provide identity- and location privacy solutions. This facilitates unlinkability, which is essential for credible privacy. It also enables scalability, which is ever so important for the IoT domain.

This paper represents an initial investigation of a new model for security update and incident handling for IoT devices. The model is by no means complete as it stands, but we believe that it has great promise for both better and more flexible security update and incident handling, in addition to catering to local regulatory requirements and being able to provide much needed privacy by design features.

References

[1] A. Cavoukian, "Privacy by design; the 7 foundational principles," [retrieved: 06-2016] www.ipc.on.ca/images/Resources/7foundationalprinciples.pdf, 01 2011.

[2] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," Computer Networks, vol. 76, 2015, pp. 146–164.

[3] M. Abomhara and G. M. Køien, "Security and privacy in the internet of things: Current status and open issues," in Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on. IEEE, 2014, pp. 1–8.

[4] ——, "Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks," Journal of Cyber Security, vol. 4, 2015, pp. 65–88.

[5] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: Perspectives and challenges," Wireless Networks, vol. 20, no. 8, 2014, pp. 2481–2501.

[6] G. M. Køien, "Reflections on trust in devices: an informal survey of human trust in an internet-of-things context," Wireless Personal Communications, vol. 61, no. 3, 2011, pp. 495–510.

[7] 3GPP TSG SA3, "3GPP System Architecture Evolution (SAE); Security architecture (Release 13)," 3GPP, TS 33.401, 03 2016.

[8] T. Kause and M. Peylo, "Internet X.509 Public Key Infrastructure – HTTP Transfer for the Certificate Management Protocol (CMP)," IETF, RFC 6712, 09 2012.

[9] 3GPP TSG SA3, "Security aspects of Machine-Type Communications (MTC) and other mobile data applications communications enhancements (Release 13)," 3GPP, TS 33.187, 01 2016.

[10] GSM Association, "IoT Security Guidelines Overview Document; CLP.11, Ver.1," [retrieved: 06-2016] www.gsma.com/connectedliving/wp-content/uploads/2016/02/CLP.11-v1.1.pdf, 02 2016.

[11] M. Abadi and R. Needham, "Prudent engineering practice for cryptographic protocols," IEEE transactions on Software Engineering, vol. 22, no. 1, 1996, pp. 6–15.

[12] Cyber Physical Systems Public Working Group, "DRAFT: Framework for Cyber-Physical Systems," NIST, USA, Public Review Draft v0.8 Framework, 09 2015.

[13] ARM Ltd., "Cortex-M4 Processor," [retrieved: 06-2016] www.arm.com/products/processors/cortex-m/cortex-m4-processor.php, 2016.

[14] G. M. Køien, "Reflections on evolving large-scale security architectures," International Journal on Advances in Security Volume 8, Number 1 & 2, 2015, 2015, pp. 60–78.

[15] A. S. Tanenbaum, "Lessons learned from 30 years of minix," Communications of the ACM, vol. 59, no. 3, 2016, pp. 70–78.

[16] S. Spiekermann, "The challenges of privacy by design," Communications of the ACM, vol. 55, no. 7, 2012, pp. 38–40.

[17] D. Le Métayer, "Privacy by design: a formal framework for the analysis of architectural choices," in Proceedings of the third ACM conference on Data and application security and privacy. ACM, 2013, pp. 95–104.

[18] ISO/IEC, "ISO/IEC 11889-1:2015," ISO, Geneva, Switzerland, Standard 11889-1:2015, 08 2015.

[19] ARM Connected Community., "Whitepaper - ARMv8-M Architecture Technical Overview," [retrieved: 06-2016] https://community.arm.com/docs/DOC-10896, 2015.

[20] F. Armknecht, A.-R. Sadeghi, S. Schulz, and C. Wachsmann, "A security framework for the analysis and design of software attestation," in Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 1–12.

[21] G. M. Køien and V. A. Oleshchuk, Aspects of Personal Privacy in Communications-Problems, Technology and Solutions. River Publishers, 2013.

[22] G. M. Køien, "A privacy enhanced device access protocol for an iot context," Security and Communication Networks, vol. 9, no. 5, 03 2016, pp. 440–450.

[23] ——, "Privacy enhanced cellular access security," in Proceedings of the 4th ACM Workshop on Wireless Security, ser. WiSe '05. New York, NY, USA: ACM, 2005, pp. 57–66.

[24] "The Tor Project," [retrieved: 06-2016] www.torproject.org, 2016.

[25] N. P. Smart, V. Rijmen, M. Stam, B. Warinschi, and G. Watson, "Study on cryptographic protocols," ENISA, Report TP-06-14-085-EN-N, 11 2014.

[26] N. P. Smart et al., "Algorithms, key size and parameters report 2014," ENISA, Report TP-05-14-084-EN-N, 11 2014.

[27] First, "Common vulnerability scoring system, v3," [retrieved: 06-2016] https://www.first.org/cvss, 06 2015.

[28] D. J. Klinedinst, "CVSS and the Internet of Things," SEI Insights, [retrieved: 06-2016] insights.sei.cmu.edu/cert/, 09 2015.

[29] M. Dekker and C. Karsberg, "Technical guidance on the incident reporting in article 13a (ver.2.1)," ENISA, Report, 10 2014.

[30] P.-H. Kamp, "More encryption means less privacy," Commuications of the ACM, vol. 59, no. 4, 04 2016, pp. 40–42.

[31] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol; Version 1.2," IETF, RFC 5246, 08 2008.

[32] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," IETF, RFC 4301, 12 2005.

[33] H. Krawczyk, K. G. Paterson, and H. Wee, "On the security of the tls protocol: A systematic analysis," in Advances in Cryptology–CRYPTO 2013. Springer, 2013, pp. 429–448.

[34] C. Hlauschek, M. Gruber, F. Fankhauser, and C. Schanes, "Prying open pandora's box: Kci attacks against tls," in 9th USENIX Workshop on Offensive Technologies (WOOT 15), 2015, pp. 1–15.

 ISBN: 978-1-61208-493-0