LUA-IoT: Let's Usably Authenticate the IoT

Markus Dahlmanns, Jan Pennekamp, Robin Decker, and Klaus Wehrle

Communication and Distributed Systems, RWTH Aachen University, Germany
{dahlmanns, pennekamp, decker, wehrle}@comsys.rwth-aachen.de

Abstract. Following the advent of the Internet of Things (IoT), users and their devices transmit sensitive data over the Internet. For the Web, Let's Encrypt offers a usable foundation to safeguard such data by straightforwardly issuing certificates. However, its approach is not directly applicable to the IoT as deployments lack a (dedicated) domain or miss essentials to prove domain ownership required for Let's Encrypt. Thus, a usable approach to secure IoT deployments by properly authenticating IoT devices is missing. To close this research gap, we propose LUA-IoT, our framework to **L**et's **U**sably **A**uthenticate the **IoT**. LUA-IoT enables autonomous certificate enrollment by orienting at the success story of Let's Encrypt, seamlessly integrating in the setup process of modern IoT devices, and relying on process steps that users already know from other domains. In the end, LUA-IoT binds the authenticity of IoT deployments to a globally valid user identifier, e.g., an email address, that is included in certificates directly issued to the IoT deployments. We exemplarily implement LUA-IoT to show that it is realizable on commodity IoT hardware and conduct a small user study indicating that LUA-IoT indeed nudges users to safeguard their devices and data (transmissions).

1 Introduction

The Internet of Things (IoT) offers several amenities on a broad scale [29, 35, 44]. However, to realize these benefits, IoT deployments, i.e., IoT devices and backend servers, have to communicate sensitive data and control commands [14, 58, 60, 66]. In this regard, threats evolve around eavesdropping on sensitive information or injecting malicious commands [58, 60, 66]. Thus, communicating entities should not only encrypt and integrity-protect sent data but authenticate each other.

On the Internet, Transport Layer Security (TLS) [22] is the predominant protocol for confidential, integrity-protected, and authenticated communication. However, even today, its secure operation still challenges IT administrators leading to various insecure deployments [33, 34, 39]. This situation aggravates in the IoT since IoT deployments requiring manual configuration, i.e., deployments that do not only communicate with the manufacturer's cloud, have an even higher risk of insecure configuration: Many deployments are operated by less security-experienced users or novices, e.g., in smart homes or industry. Indeed, several studies stress the prevalence of misconfigured and insecurely communicating IoT deployments [20, 45, 61]. Their findings include entirely disabled communication security and configurations relying on deprecated primitives, e.g., `MD5`.

On the Web, Let's Encrypt caused a significant increase in secure traffic by reducing the barrier for secure communication [1]. To this end, Let's Encrypt provides a usable method to obtain and retrieve a globally trusted certificate (for authentication). In this context, it requires a domain to be included in the certificate and the end device to prove ownership of the domain, e.g., by serving a token via HTTP or including it in a DNS record. However, IoT deployments and their owners frequently lack a domain and by far not all IoT deployments offer webpages to the Internet or have access to the DNS configuration. Hence, the IoT cannot widely benefit from the simplicity of Let's Encrypt.

Moreover, other IoT-oriented authentication schemes still do not address all requirements that are fundamental for any practical approach. For example, approaches relying on pre-installed secrets [4, 23, 30–32, 57, 63, 64] increase the probability of key leakage while simultaneously decreasing the usability by often requiring users to handle secrets. Likewise, delegation approaches [5, 42, 47, 56, 72] usually only allow devices within a single network to authenticate each other.

In this paper, we address the pressing gap of a missing usable and practical authentication scheme for the IoT. To this end, we present LUA-IoT—our approach to **L**et's **U**sably **A**uthenticate the **IoT**. LUA-IoT allows users to register themselves at a Certificate Authority (CA), validate an identifier, e.g., an email address, and add IoT devices to this account, which, in turn, can periodically request up-to-date certificates. These certificates include the user's identifier and a device identifier. Thus, with these certificates, IoT devices can appropriately authenticate themselves. Overall, LUA-IoT orients along (i) the success story of Let's Encrypt regarding the simple enrollment of up-to-date certificates, (ii) the conventional setup process of IoT devices for seamless integration, and (iii) steps users already know from other procedures, e.g., email address validation.

Contributions: Our main contributions are as follows.

- To boost secured and authenticated communication in today's IoT, we propose LUA-IoT—our scheme to **L**et's **U**sably **A**uthenticate the **IoT**.
- We show that LUA-IoT can be efficiently realized on commodity IoT device hardware, e.g., an ESP8266 and ESP32.
- We conduct a small user study (n=23) indicating that LUA-IoT is easy to deploy by being significantly more usable than self-generated certificates.
- We open-source our implementation of LUA-IoT to enable the reproducibility of our results and decrease the hurdle of integrating LUA-IoT into devices [17].

2 The IoT and its Security Challenges

Figure 1 shows an example Internet of Things (IoT) scenario including a variety of devices and protocols to communicate highly sensitive data introducing different attack vectors. Notably, we focus on scenarios where devices do not (only) communicate with any manufacturer's cloud, e.g., for privacy reasons. Especially in these cases, users have a high burden in configuring all devices and services securely. While we exemplarily focus on smart homes, LUA-IoT also supports other IoT subordinates, e.g., the Industrial IoT.

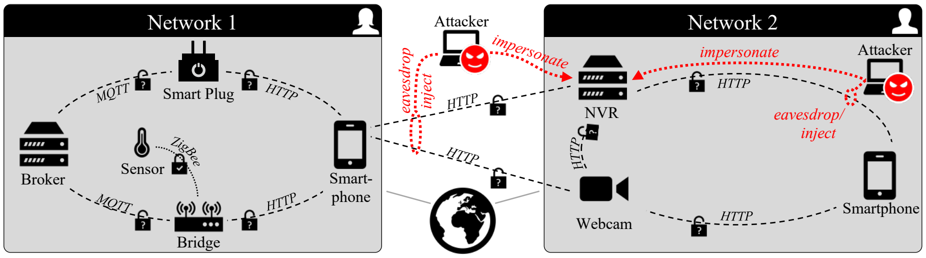


Fig. 1. Two IoT networks, their included devices, as well as their secure-by-default (🔒) and optionally-secure (🔓) communication. When users do not configure their deployments securely, attackers easily can access sensitive data or issue control commands.

2.1 Devices and Components

Figure 1 shows the different device classes the IoT encloses [37, 40]. There are (i) sensing devices, e.g., a sensor or a webcam, and actuators, e.g., a smart plug, that have direct access to sensitive user data and interplay with the physical world, (ii) bridges that connect low-cost environmental sensors to the local network, (iii) smartphones that allow users to observe their homes and issue commands, as well as (iv) servers like brokers or Network Video Recorders (NVRs) that receive, store, and send data or commands. While most sensors and actuators have in common that they are produced as cheaply as possible, their resource constraints and allowed energy consumption depend on their use-case.

2.2 Communication

IoT devices exchange very sensitive information [9, 36, 41]. For example, the smart plug in Network 1 (Figure 1) sends its state as well as measured power consumption and receives control commands, and the webcam in Network 2 provides a live stream. Depending on where the two communicating devices are, the communication scenario differs, i.e., the devices communicate locally within the same network or communicate via the Internet.

Local Communication: Local communication of devices within the same network allows the direct exchange of data. Thereby, resource-constrained sensors use energy-saving communication protocols, e.g., ZigBee [73] or Z-Wave [71], to communicate with a more powerful bridge relaying the data to other devices [37]. Hence, these devices are not directly reachable and the used protocols for such device-to-bridge communication are usually secure-by-default.

Contrarily, the bridges as well as less power-saving sensors and actuators usually communicate via Wi-Fi and offer their services to other devices directly. To this end, the IoT encompasses different communication patterns and protocols, e.g., many-to-many communication via MQTT [10] as well as traditional client-server communication via HTTP or CoAP [27, 59]. For easier identification, the devices usually get a DNS name with a pseudo Top Level Domain, e.g., `.local` or `.lan` [16]. However, these DNS names are only locally valid and thus cannot be used for global identification via the Internet.

Internet Communication: Many amenities also require IoT deployments to exchange data via the Internet, e.g., when remotely accessing the video stream of a webcam at home. Additionally, IoT devices might be mobile and regularly change their network or communicate via LTE or 5G, e.g., when used for tracking purposes. Thus, various IoT devices are directly accessible from the Internet [20, 70] or communicate with (self-hosted) servers via the Internet [45, 54, 61]. Still, unlike web services, by far not all IoT deployments and devices have domain names or fixed IP addresses that allow for their globally unique identification as, e.g., customer ISPs typically rely on dynamic IP addresses [48, 68].

2.3 Attack Vectors and Threat Model

In this setting, we have to deal with both local attackers and attackers on the Internet who target potentially privacy-sensitive information, e.g., videos or location data, and commands allowing to control physical devices paving the way for several attacks [14, 20, 45]. Specifically, they can (i) *eavesdrop* on the data communication, (ii) *inject* control commands into ongoing communication, e.g., by performing Man-in-the-Middle (MitM) attacks, and (iii) *impersonate* legitimate users or devices and try to log into services, as we illustrate in Figure 1. While our attackers are powerful network-level adversaries, they cannot compromise the IoT devices through other means to get access to the communication.

2.4 Theoretical Security and the Reality

The main requirements that prevent such attacks are confidential, integrity-protected, and authenticated communication. Thus, attackers are not able to eavesdrop on sensitive data, inject altered and malicious messages, or impersonate legitimate communication partners. To ensure this security, many IoT protocols, e.g., MQTT or CoAP, rely on Transport Layer Security (TLS) [22] or Datagram TLS (DTLS) [53] which usually authenticate communication partners via certificates and additionally ensure confidentiality and integrity.

TLS and DTLS: TLS and its counterpart for UDP-based connections, DTLS, realize secure communication [22, 53] (we use (D)TLS as an abbreviation for DTLS or TLS in the remainder of this paper). To this end, (D)TLS relies on cryptographic primitives, e.g., a Diffie-Hellman key exchange, that enable the server and client to agree on key material for encryption and integrity-protection. However, without a proper process for communication partners to identify and authenticate each other, (D)TLS cannot protect against the presented attacks.

Certificates and Public Key Infrastructures: To allow communication partners to prove their identity, (D)TLS supports the usage of certificates [22, 53]. Specifically, certificates include a public key of which the matching private key is only known by the certificate-owning device. For identification, the device that delivers the certificate proves knowing the private key during the handshake.

In this regard, two certificate types exist. Self-signed certificates force users to (manually) pin all trusted certificates on all communicating devices [12]. Thus, they do not require any unique identifier, e.g., a domain, to be bound at but are inflexible to use. Contrarily, a Public Key Infrastructure (PKI) allows more flexibility but requires a unique and verifiable identifier. Upon request, Certificate Authorities (CAs) issue certificates that bind a public key to this identifier after verifying that the identifier indeed belongs to the requester [18]. When communicating and upon certificate retrieval, the communication partners check whether the identifier matches, the opponent has the corresponding private key, and a trusted CA validly signed the certificate. For the latter, different operating systems, browsers, and IoT devices include root certificates of well-known CAs [43,49]. While IoT devices can easily check the authenticity of (cloud) backbone services by their domain, they usually do not have a domain or any other attribute usable as a verifiable identifier on their own.

Additionally, the security of the authentication bases on cryptographic primitives used to create certificates, i.e., the hash function to generate the signature and the scheme the included key material relies on [3]. Hence, to keep installations secure and prevent impersonation attacks, operators need to replace certificates relying on deprecated primitives. To enforce operators for regular replacements to update used security primitives and account for possibly leaked cryptographic secrets, certificates contain an expiry date [15].

Real-World Insights: Indeed, research found various Internet-reachable IoT deployments of which a high share put the confidentiality of sensitive data and the user's control over their devices at risk [19–21,45,61]. Most likely the majority of deployments does not implement any security mechanisms or configure them insecurely due to the complex configuration of TLS [39]. Yet, whenever users rely on external CAs issuing their certificates, the security is higher [20].
Takeaway: *Introducing various amenities, the IoT transmits highly sensitive data which attracts attackers. Thus, the encompassing heterogeneous devices must communicate securely. However, many deployments fail to implement secure communication, e.g., due to high burdens in configuration for proper authentication.*

3 Requirements for IoT Authenticity

Preventing MitM and impersonation attacks requires authentication mechanisms. We thus derive five requirements from related work [3,29,35,37–39] that approaches must address to practically enable authentication in the IoT.

R1: IoT Heterogeneity: To implement authentication in the entire IoT landscape, approaches must be compatible with the various device types and deployment scenarios [37]. Thus, any approach must, on the one hand, cope with different levels of computation power. On the other hand, it needs to support deployment scenarios relying on different communication protocols, performing local or Internet communication, but must not require additional special resources and means, e.g., no domain and no access from outside the own network.

R2: Secret Security: The essence of authentication is that only the device can prove a given identity to which it belongs [3]. Thus, all required secrets to prove a specific identity need to stay on the single device.

R3: Usability: The configuration of authentic communication already challenges experienced IT administrators [39]. Hence, it challenges operators of IoT devices and services with less experience in security even more. Contrarily, Let's Encrypt showed that good usability of security tools and concepts leads to their application and thus more secure deployments [1, 65] although still leaving room for improvements [46]. Thus, approaches allowing authentication in the IoT must be simple to set up and must not require any profound security knowledge.

R4: Scalability: Already today the IoT encompasses billions of devices and even single deployments like smart homes comprise an increasing number of IoT devices [29, 35]. Thus, approaches for authentication must be scalable to cope with this rising number of devices.

R5: Global Trust: The IoT allows users to communicate with devices of other owners, e.g., in a smart home when visiting friends or in the industry when companies track parcels sent by others [28, 69]. Additionally, paradigms like the Social IoT envision that devices communicate without any action of a human [38]. Hence, authentication in the IoT must rely on a global root of trust, i.e., if required all devices and services should be able to authenticate each other.
Takeaway: *To practically enable authentic communication in the IoT to secure its operation, approaches must consider five distinct design requirements.*

4 Related Work

Various approaches tried to improve authentication in the IoT and beyond. In this section, we analyze these approaches and show if they address the identified requirements. To the best of our knowledge, all IoT-specific approaches overlooked the importance of usability, i.e., none of the authentication approaches consider usability as an essential requirement. Table 1 summarizes our results.

Pre-Installed Secrets: To address the main problem of authentication, i.e., assigning an identity key to a specific device, numerous approaches and protocols, e.g., [4, 13, 23, 30–32, 57, 63, 64], require the user or the device manufacturer to pre-install secrets on the device. This approach in general allows for authentic communication in the heterogeneous IoT (R1) and usually scales with its growing size (R4). However, relying the device identity on pre-installed secrets requires trust in the device manufacturer as it has access to the keys during production (R2), or has a massive impact on usability when users are required to manually install secrets (R3; note that we show the impact of the limited usability in our user study (cf. Section 8)). Additionally, other approaches, e.g., JEDI [40], rely on IBE-based secrets not easily allowing a global trust infrastructure due to the lack of a globally trusted, key-generating entity (R5).

Hardware Secrets: Instead of assigning an individual secret to each of their devices, manufacturers also can equip them with (non-duplicatable) hardware elements undoubtedly identifying a device, e.g., physically uncloneable functions [2, 6]. While some authentication approaches prevent impersonation despite knowing how the hardware element will react in some cases (modeling attacks [55]) (R2), most approaches require an out-of-band exchange between

Approach	**Heterogeneity** (R1)	**Security** (R2)	**Usability** (R3)	**Scalability** (R4)	**Trust** (R5)
Pre-Installed Secret [4, 13, 23, 30–32, 40, 57, 63, 64]	◕	○	◑	◕	◑
Hardware Secret [2, 6]	○	●	◑	○	○
Delegation [5, 42, 47, 56, 72]	◑	◑	◑	●	◔
Lets's Encrypt [1, 11]	○	●	◔	●	●

Table 1. No approach fully covers all needs (5×●) for practical IoT authentication.

the device and all possible communication partners. While this factor massively impacts usability (R3), it is also impossible to pair an IoT device with all other devices it might communicate with during its lifetime (R4). Additionally, all manufacturers would have to equip their devices with such elements contradicting the low-cost manufacturing of most devices (R1).

Delegation: To overcome the issue of installing a globally trusted secret on each device, various approaches rely on delegating the authentication process to a local gateway [5,47,56,72]. While delegation usually helps to relieve computation from constrained IoT devices (R1), the majority of approaches only build a local trust infrastructure, i.e., the gateway can only prove authenticity between devices in the same network, or authenticates a device to a cloud server [8] (R5). To mitigate this issue, other approaches rely on blockchains as a global trust anchor [42]. However, while the recipient IoT device relies on the gateway performing complex cryptographic operations, the communication partner itself has to perform it. Thus, these approaches do not allow constrained devices to authentically communicate with each other and break with the IoT's heterogeneity (R1).

Let's Encrypt: Mitigating the requirement of a secret installed on the device and without any additional hardware required, Let's Encrypt [1, 11] allows Web servers to retrieve a globally trusted certificate for a specific Internet domain (R5). To this end, the server proves that it controls a domain, e.g., by offering a negotiated string via HTTP or in a DNS record of the domain. While this approach scales well with an increasing number of servers and helped to massively increase the share of TLS connections on the Web [1], the impact on IoT security is limited so far. In the IoT not all devices are accessible via HTTP from the Internet and their owners usually do not control a domain (R1; cf. Figure 1) or do not want all devices to change the DNS configuration (R3).
Takeaway: *Current approaches for authentication fail to satisfy all requirements that are essential in the IoT at once, i.e., they do not sufficiently consider the security of key material or scalability in combination with usability.*

5 Authentication with LUA-IoT

Even though properly authenticated communication is essential in the IoT, none of the available authentication mechanisms allows inexperienced users to usably configure their deployments. We thus develop LUA-IoT, our framework to **L**et's **U**sably **A**uthenticate the **IoT**.

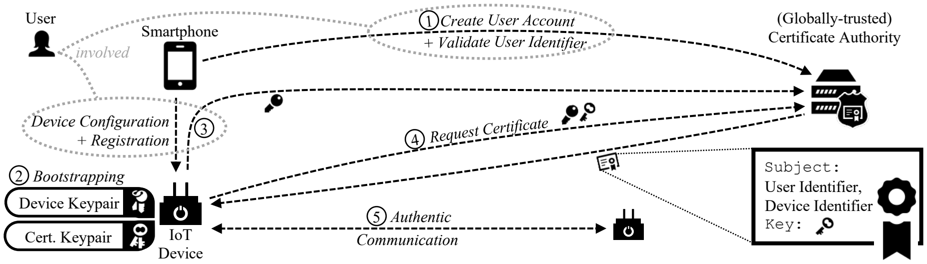


Fig. 2. LUA-IoT requires up to four steps during the IoT device setup until the IoT device is able to continuously request new certificates from the CA.

The core idea of LUA-IoT is to introduce a usable setup process that builds the foundation for authentic communication in the IoT. To this end, LUA-IoT aligns all steps users have to traverse with actions known from other domains, ensuring its overall usability. From a technical perspective, LUA-IoT binds the authenticity of a device[1] to a device keypair, a user-defined device identifier, as well as a static but verifiable user identifier, e.g., the user's email address. When using LUA-IoT, IoT devices request certificates from globally trusted CAs after users registered and added their devices to their account. Later, the devices can share their certificate with their communication partners. Such a certificate unequivocally contains both identifiers in the subject field.

5.1 LUA-IoT's Framework

Figure 2 illustrates the setup process when utilizing LUA-IoT. Once the user creates an account at a CA and proves the ownership of the chosen identifier, e.g., via a confirmation link in an email (Step ①, Section 5.1). When users add a new device to their IoT deployment, the device first generates a device key pair for identification (Step ②, Section 5.1). Then, the user adds the device to the CA account during the device's setup process, i.e., the process automatically submits the device's public key to the CA (Step ③, Section 5.1). After the setup, the device can continuously request new certificates that include the user and device identifier (Step ④, Section 5.1). Finally, when authenticating each other (Step ⑤, Section 5.1), devices can validate and trust certificates that include (i) a specific user identifier, i.e., all devices of a user, *or* (ii) a specific device of a user and communicate securely sensitive data in a local network and over the Internet.

User Account Creation To enable the issuance of certificates without relying on specialized identifiers, e.g., a domain, LUA-IoT requires users to register at a participating CA of their choice, e.g., at a non-profit CA like Let's Encrypt. To ensure good usability (R3), the registration process aligns with processes users already perform frequently on the Internet, e.g., for online shopping.

When creating the account, the user selects an identifier that the CA will include in all certificates issued for devices of this specific user. The only re-

[1] LUA-IoT is also compatible with standard (IoT-backbone) servers.

quirement for the identifier is that the CA can verify the respective ownership. For example, the CA can test the ownership of an email address by sending a verification link or the ownership of a phone number by making a test call. Still, a globally valid and easy-to-remember identifier is important as it allows users to easily share their identity and thus enables users to trust the devices of other users they know. All in all, LUA-IoT will bind the device's identity to a globally unique identifier of the user which users can exchange to trust devices of others.

Device Bootstrapping To enable the CA to identify a device when it requests a new certificate, LUA-IoT requires all devices to have their own device keypair. Hence, we recommend that IoT devices generate a fresh keypair on their first boot to guarantee secret security (R2). To further enable devices to renew their certificates with fresh key material, LUA-IoT permits the use of a different certificate keypair that the device solely uses for authentication to other devices.

For a device changing ownership, e.g., device reselling, devices should wipe their keypairs during a factory reset and generate new pairs during the next boot. This way, the new owner cannot impersonate the device and thus cannot request new certificates on their behalf to, e.g., impersonate it for communication with other devices of the original owner (R2). Optionally, to still support very constrained devices, not capable of a key generation, device keys could also be pregenerated by the device manufacturer (R1) which, however, negatively impacts secret security (R2). We consider these aspects orthogonal to the LUA-IoT framework and thus refrain from discussing them in more detail.

Finally, to enable commodity hardware communicating authentically using LUA-IoT, e.g., a home server acting as an MQTT broker, all these bootstrapping steps can be accomplished by an application similar to CertBot [1] (R1).

Device Configuration To add the IoT device to the user's CA account, LUA-IoT requires slight changes to the device configuration process users already know to (i) get the device public key and register it at the CA and (ii) configure the device with information on the chosen CA.

More specifically, during the conventional device configuration process, LUA-IoT leverages the communication between the IoT device and the device used for configuration, e.g., a user-controlled smartphone. During this process, the IoT device transmits the device public key to the smartphone. In turn, the smartphone relays the device key to the CA once the user logged into the CA account. Additionally, the smartphone sends information required to request certificates, e.g., the CAs endpoint, back to the device.

This information exchange integrates nicely into today's setup processes of IoT devices. For example, Wi-Fi-based IoT devices usually open their own access point waiting for the user to connect and configure the device via a web interface or app, e.g., by inserting credentials of the user's Wi-Fi network. Hence, besides logging into the user's CA account, which could also be cached on the user's device, from the user's perspective LUA-IoT requires no changes to the default setup process of the IoT device in question (R3).

Certificate Request After the device configuration, i.e., after the device received all information about the chosen CA, it requests its initial certificate.

Secure Communication: To ensure an authentic certificate retrieval, e.g., to prevent attackers from injecting invalid certificates or request certificates on behalf of others, the IoT device connects via (D)TLS to the CA (R2). Upon connection establishment, the CA authenticates to the IoT device via a standard certificate, i.e., the IoT device validates the received certificate using a pre-installed root store. For the other direction, LUA-IoT leverages the Raw Public Key (RPK) extension [67] allowing the CA to identify and authenticate the IoT device during the (D)TLS handshake based on the device public key. If this public key is unknown to the CA or the IoT device cannot prove the possession of the corresponding private key, the CA aborts the process.

Certificate Enrollment: After successfully completing the handshake and establishing a secure and authentic connection, the IoT device can request its certificate. To this end, LUA-IoT relies on the Enrollment over Secure Transport (EST (TLS)) [50] or EST-coaps (DTLS) [62] protocol. First, the IoT device generates a new certificate keypair (can be pre-generated). Subsequently, it sends a certification request to the CA which includes the certificate public key. In turn, the CA responds with a certificate that includes the user's identifier, the device's certificate public key, and the device identifier. Optionally, the CA can also generate a certificate key pair and send the private key together with the certificate to the client, e.g., to support devices not capable of key generation (R1).

As periodic replacement of certificates is recommended to ensure secure operation, e.g., to replace cryptographic primitives that lost their promises, the CAs should adhere to state-of-the-art recommendations, e.g., issuing certificates with a validity of at most 398 days [15] or less (e.g., 90 days as done by Let's Encrypt). To renew the certificate the IoT device can repeat this step during its entire lifetime. This way, LUA-IoT ensures that the devices are equipped with valid, up-to-date certificates that rely on secure primitives. If an IoT device does not support any secure primitives anymore, the CA can notify the user about the security problem, so that the user can update the device's software or replace it completely to continue maintaining a secure operation.

Device Authentication Once an IoT device received its certificate, it can authenticate on incoming and outgoing connections locally and over the Internet, e.g., during a (D)TLS handshake. After the communication partner received the IoT device's certificate and verified that (i) the device indeed possesses the matching private key and (ii) the CA signature is valid, it checks the content of the certificate. Thereby, the level of trust remains configurable in LUA-IoT.

Trusting Own Devices: LUA-IoT's default setting is that devices of the same user trust each other (secure-by-default). Here, the communication partners simply validate that the received certificate includes the same user identifier which also scales in large deployments and the IoT as a whole (R4).

Trusting Devices of Others: To extend the trust boundary and enable IoT devices to trust devices owned by others e.g., when tracking parcels [28, 69], users can configure their devices to accept certificates that include other specific user identifiers. As devices usually communicate in clusters, i.e., devices of a single user communicate with devices of a few other owners, this practice is maintainable and scalable through the complete IoT as well (R4).

Authenticating Specific Devices: When validating certificates, until now, only the ownership of a device to a specific user can be confirmed. While this practice is the strength of LUA-IoT and secure on its own, as users need to trust the owners of devices as much as the devices themselves, LUA-IoT also allows to validate the device's identity. To this end, LUA-IoT allows users to further add device identifiers to their configurations. Still, this practice might include numerous identifiers and introduces additional configuration overhead (R3, R4).
Takeaway: *LUA-IoT binds the authenticity of each device to a globally unique and verifiable identifier of its owner, e.g., an email address, that is included in certificates IoT deployments autonomously and periodically renew at globally trusted CAs. By relying on these device certificates, LUA-IoT allows for confidential, integrity-protected, and authentic communication.*

5.2 LUA-IoT's Usability Approach

To implement good usability, LUA-IoT aligns its steps that involve users with actions they already know from other applications. LUA-IoT's user account creation (Step ① in Figure 2) constitutes a one-time action and is identical to widespread registration processes including the verification of an email address. Second, configuring the IoT device for LUA-IoT can simply be integrated into the conventional setup process of the device (Step ③). As part of this step, users only have to log in to the user account created in Step ① and LUA-IoT transparently transfers any necessary information from the IoT device. Optionally, when the device should not only trust other devices of the same user, the user can also define identifiers of others that the IoT device can also trust.

Steps ②, ④, and ⑤ do not involve the user and occur automatically. More specifically, generating the device keypair (Step ②) is a one-time operation and can be performed when the user turns on the IoT device for the first time. Additionally, the device autonomously triggers the certificate enrollment process (Step ④) after the configuration and periodically renews its certificate. Likewise, establishing the authenticated communication between IoT devices (Step ⑤) is transparent to the user, i.e., the user is not involved during regular operation.

For exceptional steps, e.g., revocation, LUA-IoT can exploit established procedures known from the traditional Web. While these actions are not part of a regular operation and unlikely to occur frequently, their results might influence the operation of IoT services, e.g., when connections are declined due to invalid certificates. In this case, the implementations must provide comprehensible error messages and advice on how (novice) users can resolve them.
Takeaway: *All of LUA-IoT's steps that directly involve the user align with actions users already know from other domains, e.g., validating their email address when creating online accounts. Hence, LUA-IoT ensures being easily usable without special knowledge in IT security.*

6 Security Discussion

Given that security is paramount for the success of any usable scheme in the IoT, we now discuss how LUA-IoT realizes authentic communication without relying on any *specialized* resources, e.g., own domains. Specifically, we state how LUA-IoT meets the requirements for secret security (R2) and global trust (R5).

In the IoT setting, attackers could resort to impersonating legitimate devices, performing MitM attacks, or eavesdropping on sensitive data (cf. Section 2.3). To prevent such attacks, users and devices can use a secure transport layer protocol, e.g., (D)TLS, that (i) provides integrity protection, (ii) offers confidentiality through encryption, and (iii) supports authentication, while not relying on deprecated primitives. While IoT deployments frequently miss essentials for authentication, e.g., a domain or fixed IP address, LUA-IoT enables users to create an account at participating CAs, verify the control over any public identifier, e.g., an email address, and associate IoT devices during their setup process to their account using the device key pair. The CA then issues certificates including the user's identifier, a device identifier and the certificate public key directly to IoT devices after their authentication using the device private key.

Key Security: To ensure that attackers cannot impersonate the IoT device, attackers may not have access to the device private key or the certificate private key. Otherwise, attackers could impersonate the device to its communication partners or toward the CA to request new valid certificates before. By storing the private keys only on the IoT device itself, LUA-IoT prevents attackers from getting access, as long as the IoT device is not compromised (R2). Given that the secure device configuration is out of scope of the LUA-IoT scheme, the attack vector of extracting private keys from the IoT devices is as small as possible.

Account Security: Another attack vector follows from an illegitimate communication with the CA to receive a valid certificate, i.e., adding a malicious device key pair to a victim's user account and subsequently requesting a certificate with the victim's user identifier. LUA-IoT secures the user account conventionally, i.e., it features a login prompt. Since this mechanism is not bound to the scheme of LUA-IoT, this login mechanism may go beyond a simple user-password prompt. For example, the CA may implement two-factor authentication [51, 52], which is already commonly used in practice, e.g., to realize financial services on the Web. Moreover, the CA may inform the user via email about new (suspicious) logins or added devices allowing checking for malicious changes in the account and subsequently removing malicious device entries. We refer to established and usable best practices to address corresponding threats in LUA-IoT.

Trust: To ensure authentic communication in the IoT, LUA-IoT requires (i) users to protect their account and devices to ensure that attackers cannot request valid certificates, as well as (ii) the involved CAs to only issue genuine certificates. Hence, LUA-IoT builds on the inherent interest of users not purposely harming their devices' authenticity and the same trust that is shown to CAs on the Web, i.e., not issuing faulty certificates. While LUA-IoT can support users in account security, e.g., with two-factor authentication, the supervision of CAs using Certificate Transparency Logs (CTLs) would help to check for their

integrity. Overall, these aspects allow LUA-IoT to authenticate all IoT devices with each other based on the user and optionally the device identifier (R5).
Takeaway: *LUA-IoT adds barely additional attack surface to the IoT since its design ensures to keep private keys secret and user accounts safe. Simultaneously it allows IoT devices to trust each other helping to appropriately secure the IoT.*

7 Performance Evaluation

After attesting the security of LUA-IoT, we now have to substantiate that LUA-IoT is compatible to heterogeneous hardware prevalent in the variety of IoT devices that communicate via TCP or UDP (R1). With such compatibility, we ensure that LUA-IoT is indeed a prime candidate to establish a solid foundation for authentic and secure communication in the IoT.

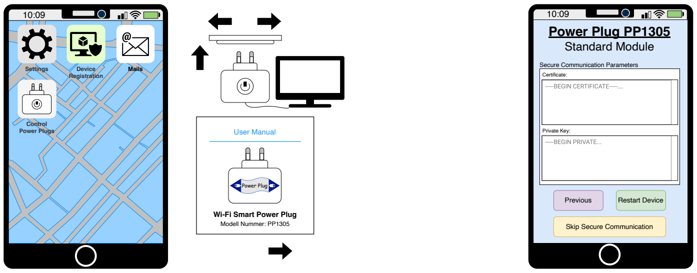
Experimental Setup: To conduct our evaluation, we prototype a LUA-IoT-enabled CA and a client library. For evaluation of a lower bound, we run our library on constrained ESP8266 and ESP32 controllers which are widely used in IoT devices [25,26], set their frequency to the lowest possible, i.e., 80 MHz, rely on LUA-IoT's TLS variant, and execute the CA on commodity hardware (Quad-Core, 2.6 GHz). Using this testbed, we evaluate the performance of LUA-IoT's certificate retrieval that IoT deployments have to accomplish and report on the time required (arith. mean of 200 runs ± 99 % CI).

Secure Communication: The first step to retrieve a certificate is the device establishing a secure connection to the CA. In our setup this step takes 3.9788 s ± 0.019 s on the ESP32 and 2.8816 s ± 0.019 s on the ESP8266 including all communication latency. While this duration seems long, the connection to the CA is only required during the setup and when the certificate expires. Thus, the overhead is manageable for the renewal process and shows that a TLS handshake is generally feasible on these controllers. For connections between IoT deployments, the TLS handshake duration is also of little importance as connections in the IoT are rather long and measures exist to reduce the overhead of subsequent handshakes, e.g., session resumption.

Certificate Enrollment: The certificate enrollment process is comparably short. To generate fresh ECC key material, create the enrollment request, and store the certificate in total the ESP8266 requires 1798 ms ± 1.67 ms and the ESP32 needs 1570 ms ± 5.48 ms. Hence, the enrollment is also feasible.
Takeaway: *LUA-IoT adds a small computational overhead. As this overhead is manageable for many IoT deployments it is no obstacle for LUA-IoT to increase the share of secure communication in the IoT.*

8 User Study

We designed LUA-IoT specifically with usability in mind. Thus, in addition to LUA-IoT's security and performance, we also evaluate whether its workflow is straightforward and has the potential to increase the share of secured IoT communication, we conducted a user study.



(a) (b)

Fig. 3. The web interface of our study includes smartphone and smart plug mockups as well as a user manual for the smart plug *(a)*. The control group had to insert a certificate and a private key to secure communication *(b)*.

8.1 Study Design

To capture (i) how users interact with LUA-IoT and (ii) what their perception is, we split our study into two parts. First, a task where users configure an IoT device in our custom online tool, and second, a subsequent survey.

Online Configuration Task To reduce the burden on our study participants and ensure our study's scalability, we evaluated the usability via an online study that is as close to reality as possible.

User Interface and Tutorial: Figure 3(a) details the web interface of our study tool from a participant's perspective. The interface displays (i) a smartphone which is required to set up the (ii) smart plug which is connected to a TV, and (iii) a user manual giving hints on the plug's configuration. While the participants can intuitively interact with the smartphone by clicking on the screen as well as with the smart plug's button, they can trigger physical changes via the shown arrows. Using these arrows, users can rotate the smart plug (e.g., to discover a sticker with setup credentials), (un)plug it to/from the wall and turn the manual's pages. We prepended a tutorial (cf. Appendix A.1) to allow the participants to familiarize themselves with these controls.

General Task: We communicated to the participants that they should configure the smart plug as they would on their own, before switching on the TV via the smartphone to complete the study. Orienting the general setup process of the plug to workflows of standard IoT devices, the smart plug opens a Wi-Fi access point when connected to power where the participants have to connect the smartphone to. Afterward, the plug's interface guides through the further configuration, i.e., mainly to select the study's "home" Wi-Fi and insert the corresponding credentials which are always visible below the study's interface. During the configuration, the participants can enable secure communication.

Security and Participant Groups: We implemented separate studies for two groups: a *control* group and a *LUA-IoT*-enabled study. Figure 3(b) shows the installation guide to allow participants of the *control* group to insert a certificate and a private key to the IoT device to enable authentic communication. While the

Group		**Total**	**LUA-IoT**	**Control**
n		23	12	11
Demographics				
Gender	Female	8.7 %	0 %	18 %
	Male	87 %	100 %	73 %
	Undisclosed	4.3 %	0 %	9.1 %
Age	Mean	24	24	25
	StdDev	2.48	2.48	2.48
Country	Germany	100 %	100 %	100 %
Student		96 %	92 %	100 %
C.S. Degree	Bachelor's level	22 %	8.3 %	36 %
	Yes	78 %	92 %	64 %
C.S. Employed		83 %	92 %	73 %
Experience				
Smart Home	Setup (myself)	57 %	58 %	55 %
	Setup (others)	35 %	42 %	27 %
	Used	61 %	58 %	64 %
Devices Set Up	5+	30 %	42 %	18 %
	3-4	13 %	17 %	9.1 %
	1-2	13 %	0 %	27 %
Certificates	CA-signed	74 %	83 %	64 %
	Self-signed	65 %	67 %	64 %

Table 2. Overview of the demographics and experiences of our 23 study participants. While most of them are students the experience regarding smart home devices and certificates slightly shifts to the LUA-IoT group.

barrier on a real smartphone is even higher, i.e., users barely can generate, copy, and paste a certificate to the web interface, it is easier in our PC-oriented study. The *LUA-IoT*-enabled workflow guides the participants through the complete framework, i.e., it (i) requires to set a device identity, (ii) nudges to register at a CA using a predefined email address, and (iii) verify the "ownership" of the address. In both groups, the participants can skip the security configuration.

Survey After performing the task, we ask our participants a few questions on their demographics, experience with smart home devices, and their perception of the task (cf. Appendix A.2). For the latter, we focus on the security configuration and their judgment of how sensitive the communication with a smart plug is. Based on this information, we are able to classify the decisions participants made.

8.2 Participants

Since our study does not require any prerequisites from participants, we recruit them from our lab's student mailing list with 124 entries encompassing student workers as well as thesis students (Bachelor's and Master's level) promising participants with different experience levels. Table 2 gives an overview of the demographics and experience with certificates as well as smart home deployments of our 23 participants. Furthermore, it shows how our alternating assignment split participants with different experience levels into the two study groups.

Demographics: The gender distribution of our participants fairly coincides with the distribution in the computer science degrees at our institution. Furthermore, all participants reside in the country where our lab is located in. The distribution of age, studentship, and degree in computer science are as expected since we have undergraduates, B.Sc., and a few M.Sc. graduates on our mailing list. Although we targeted an equal distribution, a larger share of participants in the LUA-IoT group are graduates and thus might have a better understanding for computer science and especially security.

Experience: The experience regarding certificates and setting up smart home devices shows a similar shift: A larger share of participants in the LUA-IoT group has set up (more) smart home devices for themselves and others. Additionally, while the participants in the LUA-IoT group have slightly more experience in handling certificates, in both groups the majority acknowledges their experience. While the experience level is slightly higher than expected from operators of smart home deployments, our study will still provide meaningful insights on how the usability of LUA-IoT compares to a traditional security configuration.

8.3 Ethical Considerations

Our research warrants ethical consideration as it involves human subjects. We thus, before the study, got informed consent of our participants and briefed them that their actions do not have any influence on the real world, that they can abort the study at any time, as well as that they must not enter any sensitive information, e.g., personal passwords. We also primed them about the intended length of our study (short; only 20 min) to convince them to participate despite the lack of (monetary) compensation. Still, according to the current legislation in our country, our study does not require an IRB approval.

For educational purposes, we concluded the task with an explanation of the potential impact of (not) enabling secure communication. Moreover, we will send this publication to our student's mailing list.

8.4 Study Results

Based on our study and survey, we can derive if the increased usability of LUA-IoT nudges users to secure their IoT deployments and analyze their perception.

Security Configuration Figure 4 shows how the participants acted over time until they finished the study by turning on the smart plug via their smartphone. Especially it outlines whether they were able to prepare secure communication by correctly configuring LUA-IoT or a certificate with a private key (control group).

Notably, 92 % of the participants in the LUA-IoT group and only 9.1 % in the control group prepared secure communication. Hence, our results underpin that LUA-IoT can nudge users to more secure configurations: With LUA-IoT, an additional 83 % of users configure IoT deployments securely, i.e., the devices are able to authenticate themselves; preventing MitM or impersonation attacks.

The configuration of LUA-IoT-enabled devices took on average 5.268 min (StdDev: 1.217 min) which (i) is bearable in smart home environments with devices that are configured via a web interface, and (ii) includes the one-time invest

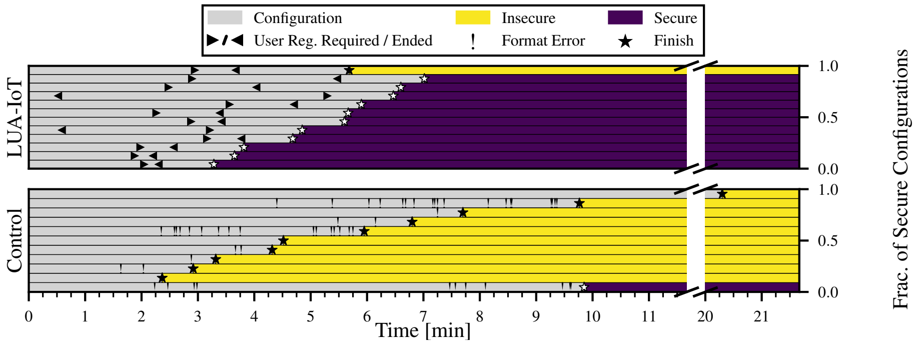


Fig. 4. Only one participant in the control group but all but one participant in the LUA-IoT group successfully configured the IoT device for authentic communication.

required to register at a CA (56 s on average (StdDev: 42 s)). Two participants even registered at the CA before the IoT device setup required it. While this behavior is traceable to participants testing out the possibilities in our study setup, it also suggests that the registration process is straightforward and doable without any background on how to register. One participant registered at the CA but still finished the setup with an insecurely configured device. Upon manual examination, we discovered a deadlock in our tool, which prevented a secure configuration for this specific sequence of user interactions; i.e., it is an outlier.

In the control group only one participant prepared the IoT device for authentic communication, although 8 participants tried to enable it by inserting certificates or private keys. However, the participants either included other cryptographic material, e.g., PGP keys, or in some cases only random strings. This result shows that, although our participants have an increased share of background in computer science, the generation of a certificate and private key is comparatively difficult and often skipped at the expense of insecure communication. Even the participant successfully preparing the device for secure communication revealed the use of a tutorial website for certificate generation to complete the task.

Reasons To (Not) Enable Security The result that all but one participant from the control group did not prepare their IoT device for secure communication is also reflected in their survey answers (cf. Question 26B in Appendix A.2); all but three study participants attest smart home devices to have access to sensitive data (cf. Question 9), e.g., usage patterns (cf. Question 10). Thereby, two participants claim to not invest as much time in the study as they would in reality when configuring an IoT device, e.g.,

> *I did not want to invest time in researching how to generate a certificate/private key pair. If this was a real device, I would have invested the time.*

While this approach might work out for some users, it still underpins how easy the configuration with LUA-IoT is, i.e., the participants did not have to look up anything. Fittingly, most participants in the LUA-IoT group acknowledge the

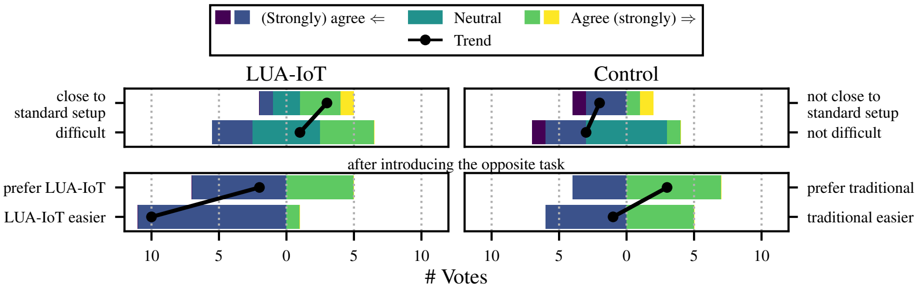


Fig. 5. The participant's perception point out the simplicity of LUA-IoT's configuration.

insignificant overhead of configuring LUA-IoT to increase the security of their IoT deployments (Question 26A), which is exemplified through:

Here is a better question for you: Why sacrifice security when it only costs a few clicks?

Participants' Perception Figure 5 summarizes how users perceived our study, i.e., how close the configuration process was to conventional IoT setup processes and how difficult it was. Additionally, after detailing the workflow of the opposite group, the participants gave insight on which one they perceive as easier and which they would prefer for their setups.

While the participants in the control group attest the setup process in our study to be rather close to conventional processes, the majority of participants in the LUA-IoT group disagreed (cf. Question 16 in Appendix A.2). This result suggests that, on the one hand, from the perspective of the participants, our control study was designed as we intended, i.e., close to standard configuration processes of known IoT devices. On the other hand, it underpins that, while using well-known principles from other domains, e.g., email address verification, LUA-IoT adds steps to an IoT device setup.

Looking at the perception of how difficult the configuration was (cf. Question 17), participants from the control group attest a slightly higher difficulty than participants from the LUA-IoT group. The fact that all but one participant from the control group did not and all but one participant from the LUA-IoT group prepared their devices securely amplifies this perception. Specifically, participants from the control group mostly bypassed inserting certificate and private key after some attempts. Contrarily, participants from the LUA-IoT group finished the entire process, including CA registration and email verification, which, in practice, constitutes a one-time task. Consequently, LUA-IoT does not lead to more complicated processes despite paving the way for secure communication.

After getting an explanation of what the setup process of the other group looks like (Figure 5 (bottom)), the majority of participants in the control group would still prefer the traditional setup (cf. Question 20). Additionally, an equal number of participants think that LUA-IoT and the traditional approach are easier, i.e., a clear trend in the dispersion is missing. Contrarily, the majority of participants in the LUA-IoT group vote for LUA-IoT being easier to use than traditionally generating and setting up certificates (cf. Question 22). Still, nearly

an equal number of participants would prefer the traditional approach. Thus, a better explanation of how LUA-IoT works and contributes to authentic and secure communication might lead to more trust from users.
Takeaway: *Through its usability, LUA-IoT nudges additional 83 % of users to a secure IoT configuration. With its far easier steps to perform than for traditional certificate handling, it only adds limited complexity to conventional workflows.*

9 Limitations and Opportunities

While LUA-IoT addresses the pressing need for a practical and usable approach to realize authentic and secure communication in the IoT, we still identify the need for follow-up research in the area, not only to address certain limitations but also to exploit opportunities that our research facilitates.

Exclusion of Very Constrained Devices: As captured in our requirements (R1), a practical authentication scheme for the IoT should also run on constrained devices. However, LUA-IoT is not compatible with very constrained devices. This restriction follows from LUA-IoT's dependence on a standard public key infrastructures, which includes the comparably intensive handling of asymmetric cryptographic operations. This limitation coincides with our fundamental design decision of LUA-IoT building on global trust (R5) while also keeping the hurdle for existing CAs to support LUA-IoT low (the chosen enrollment process does not significantly deviate from standard certificate enrollment). Thus, we decided to trade off usability and simplicity for compatibility.

We are convinced of this approach as many communication protocols for very constrained devices, e.g., ZigBee, have built-in security, establishing trust between the device and its designated bridge when pairing. This authentication is sufficient for the extremely localized networks of very constrained devices. Beyond such extremely localized communication, LUA-IoT can reliably authenticate communication between the (slightly) more powerful bridge and any other client. Hence, even then, LUA-IoT significantly increases communication security.

Focused and Small Online User Study: Apart from the heterogeneity, usability is an inherent requirement for authentication schemes in the IoT (R3). In this paper, we evaluated the usability of LUA-IoT in a rather confined online study with students from our lab (cf. Section 8). While this setting, despite our efforts of designing the online study as close to reality as possible, might not be entirely representative regarding (novice) users configuring their IoT devices in the real world, it still underpins impressively the simplicity LUA-IoT provides users when preparing their IoT devices for authentic and secure communication. In combination with LUA-IoT's design to rely on as many process steps users already know from other domains (cf. Section 5.2), we are confident that LUA-IoT has what it takes to nudge users to appropriate secured IoT deployments.

User Privacy: We only identified a single aspect that might keep users from utilizing LUA-IoT. Its use depends on distributing device certificates with the users' real identifiers, e.g., their email address. A potential alternative could be that CAs create a dummy identifier including the CAs name, e.g., `[userid]@letsencrypt.org` to ensure the uniqueness of embedded identifiers (otherwise other CAs inadvertently could issue (valid) certificates with the same identifier). Such an approach would complicate the parallel usage of two CAs or transition from one CA to another as the dummy identifiers mandate respective users to then reconfigure all of their devices. Ultimately, if rolled-out, users could choose their own identifiers or CA-specific dummy identifiers.

Manufacturer Incentive: While we have shown that LUA-IoT is a promising approach to increase the share of secure communication in the IoT, it cannot help without being supported on IoT deployments. The support of back-end IoT deployments, e.g., MQTT brokers, can be easily realized by a user application, similar to CertBot. However, the support of IoT devices, in most cases, depends on the hardware manufacturer. By open-sourcing an implementation [17], we intend to reduce the hurdle for manufacturers to program their own implementations of LUA-IoT. Additionally, alternative firmwares for many ESP8266 and ESP32-based devices exist [7, 24] in which LUA-IoT can be integrated quickly. When LUA-IoT is initially distributed via these custom firmwares, it could be important for manufacturers to keep up with their own firmware. Still, to profit from the opportunity LUA-IoT provides, i.e., practically securing the IoT, future research might look into how to convince manufacturers to adopt such security approaches.

Lastly, we look forward to future research on how to fine-tune LUA-IoT's parameters for use in atypical (IoT) scenarios. We are confident that respective studies will contribute to unleashing the full potential of our work.

10 Conclusion

The IoT not only offers various advantages to its users, it also increases the number of devices communicating over the Internet [29, 35, 44]. While the sensitivity of transmitted IoT data and control commands mandate appropriately secured, i.e., confidential, integrity-protected, and authentic communication, respective approaches to practically ensure authenticity in the IoT are missing. On the Web, the usability of Let's Encrypt contributed to a larger share of secure and authentic communication [65]. However, IoT deployments are often incompatible to such concepts due to missing essentials, e.g., a (dedicated) domain.

To address this research gap, we proposed LUA-IoT—our scheme to **L**et's **U**sably **A**uthenticate the **IoT**. With this framework, we transfer the good usability offered by Let's Encrypt to the IoT. Specifically, LUA-IoT binds the identity of IoT deployments to a common user identifier while align itself with workflows that users are already well familiar with. Thus, we provide a secure and simple-to-use approach for authentic communication in the IoT. In this regard, the results of our small user study are very promising since we observe a significant improvement of securely configured IoT deployments. Additionally, our performance evaluation shows that LUA-IoT is compatible to commodity IoT hardware.

To conclude, by introducing LUA-IoT we lay the foundation to effectively increase the share of authentic and secure communication in the IoT. To support its adaptation, we open-source our initial LUA-IoT implementation [17].

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2023 Internet of Production—390621612. The authors thank Fabian Thorsten Jess for the initial exploration of LUA-IoT and performing the user study.

References

1. Aas, J., Barnes, R., Case, B., Durumeric, Z., et al.: Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19). pp. 2473–2487. ACM (2019). https://doi.org/10.1145/3319535.3363192
2. Abdolinezhad, S., Sikora, A.: A Lightweight Mutual Authentication Protocol Based on Physical Unclonable Functions. In: Proceedings of the 2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST '22). pp. 161–164 (2022). https://doi.org/10.1109/HOST54066.2022.9840132
3. Akram, M., Barker, W.C., Clatterbuck, R., Dodson, D., et al.: Securing Web Transactions TLS Server Certificate Management. NIST SP 1800-16 (2020). https://doi.org/10.6028/nist.sp.1800-16
4. Alasmary, H., Tanveer, M.: ESCI-AKA: Enabling Secure Communication in an IoT-Enabled Smart Home Environment Using Authenticated Key Agreement Framework. Mathematics **11**(16), 3450 (2023). https://doi.org/10.3390/math11163450
5. Alzahrani, B.A., Chaudhry, S.A., Barnawi, A., Xiao, W., et al.: ILAS-IoT: An improved and lightweight authentication scheme for IoT deployment. Journal of Ambient Intelligence and Humanized Computing **13**(11), 5123–5135 (2022). https://doi.org/10.1007/s12652-020-02349-5
6. Aman, M.N., Chua, K.C., Sikdar, B.: A Light-Weight Mutual Authentication Protocol for IoT Systems. In: Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM '17). pp. 1–6. IEEE (2017). https://doi.org/10.1109/GLOCOM.2017.8253991
7. Arends, T.: Tasmota. https://github.com/arendst/Tasmota (2017 (accessed 09/04/2024))
8. Aura, T., Sethi, M., Peltonen, A.: Nimble Out-of-Band Authentication for EAP (EAP-NOOB). RFC 9140 (2021). https://doi.org/10.17487/RFC9140
9. Babun, L., Denney, K., Celik, Z.B., McDaniel, P., Uluagac, A.S.: A survey on IoT platforms: Communication, security, and privacy perspectives. Computer Networks **192**, 108040 (2021). https://doi.org/10.1016/j.comnet.2021.108040
10. Banks, A., Briggs, E., Borgendale, K., Gupta, R.: MQTT Version 5.0. OASIS Standard: mqtt-v5.0-os (2019)
11. Barnes, R., Hoffman-Andrews, J., McCarney, D., Kasten, J.: Automatic Certificate Management Environment (ACME). RFC 8555 (2019). https://doi.org/10.17487/RFC8555
12. Barrett, D., Silverman, R., Byrnes, R.G.: Linux Security Cookbook: Security Tools & Techniques. O'Reilly Media (2003)
13. Basic, F., Steger, C., Seifert, C., Kofler, R.: Trust your BMS: Designing a Lightweight Authentication Architecture for Industrial Networks. In: Proceedings of the 2022 IEEE International Conference on Industrial Technology (ICIT '22). pp. 1–6. IEEE (2022). https://doi.org/10.1109/ICIT48603.2022.10002825
14. Butun, I., Österberg, P., Song, H.: Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. IEEE Communications Surveys & Tutorials **22**(1), 616–644 (2020). https://doi.org/10.1109/COMST.2019.2953364

15. CA/Browser Forum: Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates. Tech. Rep. Version 2.0.2, CA/Browser Forum (2024)
16. Cheshire, S., Krochmal, M.: Multicast DNS. RFC 6762 (2013). https://doi.org/10.17487/RFC6762
17. COMSYS: LUA-IoT Reference Implementation. https://github.com/COMSYS/LUAIoT (2024)
18. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., et al.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (2008). https://doi.org/10.17487/RFC5280
19. Dahlmanns, M., Lohmöller, J., Fink, I.B., Pennekamp, J., et al.: Easing the Conscience with OPC UA: An Internet-Wide Study on Insecure Deployments. In: Proceedings of the ACM Internet Measurement Conference (IMC '20). pp. 101–110. ACM (2020). https://doi.org/10.1145/3419394.3423666
20. Dahlmanns, M., Lohmöller, J., Pennekamp, J., Bodenhausen, J., et al.: Missed Opportunities: Measuring the Untapped TLS Support in the Industrial Internet of Things. In: Proceedings of the 17th ACM ASIA Conference on Computer and Communications Security (ASIACCS '22). pp. 252–266. ACM (2022). https://doi.org/10.1145/3488932.3497762
21. Dahlmanns, M., Sander, C., Decker, R., Wehrle, K.: Secrets Revealed in Container Images: An Internet-wide Study on Occurrence and Impact. In: Proceedings of the 18th ACM Asia Conference on Computer and Communications Security (ASIACCS '23). pp. 797–811. ACM (2023). https://doi.org/10.1145/3579856.3590329
22. Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (2008). https://doi.org/10.17487/RFC5246
23. Elhabrush, H., Ahmeda, S.: Authentication Protocol for Wireless Sensor Network in the Internet of Things. In: Proceedings of the 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA '22). pp. 471–478. IEEE (2022). https://doi.org/10.1109/MI-STA54861.2022.9837557
24. ESPHome: ESPHome. https://github.com/esphome/esphome (2018 (accessed 09/04/2024))
25. ESPHome: Boards: esp32. https://devices.esphome.io/board/esp32 (2023 (accessed 09/04/2024))
26. ESPHome: Boards: esp8266. https://devices.esphome.io/board/esp8266 (2023 (accessed 09/04/2024))
27. Fielding, R., Nottingham, M., Reschke, J.: HTTP Semantics. IETF RFC 9110 (2022). https://doi.org/10.17487/RFC9110
28. Gnimpieba, Z.D.R., Nait-Sidi-Moh, A., Durand, D., Fortin, J.: Using Internet of Things Technologies for a Collaborative Supply Chain: Application to Tracking of Pallets and Containers. Procedia Computer Science **56**, 550–557 (2015). https://doi.org/10.1016/j.procs.2015.07.251, the 10th International Conference on Future Networks and Communications (FNC 2015) / The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015) Affiliated Workshops
29. Guo, H., Heidemann, J.: Detecting IoT Devices in the Internet. IEEE/ACM Transactions on Networking **28**(5), 2323–2336 (2020). https://doi.org/10.1109/TNET.2020.3009425
30. Haj-Hassan, A., Imine, Y., Gallais, A., Quoitin, B.: Zero-Touch Mutual Authentication Scheme for 6TiSCH Industrial IoT Networks. In: Proceedings of the 2022 International Wireless Communications and Mobile Computing (IWCMC '22). pp. 354–359. IEEE (2022). https://doi.org/10.1109/IWCMC55113.2022.9824568

31. He, Z., Furuhed, M., Raza, S.: Indraj: Digital Certificate Enrollment for Battery-Powered Wireless Devices. In: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19). pp. 117–127. ACM (2019). https://doi.org/10.1145/3317549.3323408
32. Holohan, E., Schukat, M.: Authentication Using Virtual Certificate Authorities: A New Security Paradigm for Wireless Sensor Networks. In: Proceedings of the 2010 9th IEEE International Symposium on Network Computing and Applications (NCA '10). pp. 92–99. IEEE (2010). https://doi.org/10.1109/NCA.2010.19
33. Holz, R., Amann, J., Mehani, O., Wachs, M., Ali Kaafar, M.: TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication. Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS '16) (2016). https://doi.org/10.14722/ndss.2016.23055
34. Holz, R., Braun, L., Kammenhuber, N., Carle, G.: The SSL Landscape: A Thorough Analysis of the x.509 PKI Using Active and Passive Measurements. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC '11). pp. 427–444. ACM (2011). https://doi.org/10.1145/2068816.2068856
35. Howarth, J.: 80+ Amazing IoT Statistics (2024-2030). https://explodingtopics.com/blog/iot-stats (2023 (accessed 09/04/2024))
36. Huang, D.Y., Apthorpe, N., Li, F., Acar, G., Feamster, N.: IoT Inspector: Crowdsourcing Labeled Network Traffic from Smart Home Devices at Scale. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **4**(2), 46:1–21 (2020). https://doi.org/10.1145/3397333
37. International Telecommunication Union: Overview of the Internet of things. ITU-T Y.2060 (2012)
38. Iqbal, M.A., Hussain, S., Xing, H., Imran, M.: Social IoT, chap. 9, pp. 195–211. John Wiley & Sons, Ltd (2021). https://doi.org/10.1002/9781119701460.ch9
39. Krombholz, K., Mayer, W., Schmiedecker, M., Weippl, E.: "I Have No Idea What i'm Doing": On the Usability of Deploying HTTPS. In: Proceedings of the 26th USENIX Security Symposium (USENIX SEC '17). pp. 1339–1356. USENIX Association (2017)
40. Kumar, S., Hu, Y., Andersen, M.P., Popa, R.A., Culler, D.E.: JEDI: Many-to-Many End-to-End Encryption and Key Delegation for IoT. In: Proceedings of the 28th USENIX Security Symposium (SEC '19). pp. 1519–1536. USENIX Association (2019)
41. Lee, H., Mun, H., Lee, Y.: Comparing Response Time of Home IoT Devices with or without Cloud. In: Proceedings of the 2020 IEEE International Conference on Consumer Electronics (ICCE '20). pp. 1–6. IEEE (2020). https://doi.org/10.1109/ICCE46568.2020.9043102
42. Lin, C., He, D., Kumar, N., Huang, X., et al.: HomeChain: A Blockchain-Based Secure Mutual Authentication System for Smart Homes. IEEE Internet of Things Journal **7**(2), 818–829 (2020). https://doi.org/10.1109/JIOT.2019.2944400
43. Ma, Z., Austgen, J., Mason, J., Durumeric, Z., Bailey, M.: Tracing Your Roots: Exploring the TLS Trust Anchor Ecosystem. In: Proceedings of the 21st ACM Internet Measurement Conference (IMC '21). pp. 179–194. ACM (2021). https://doi.org/10.1145/3487552.3487813
44. Madakam, S., Ramaswamy, R., Tripathi, S.: Internet of Things (IoT): A Literature Review. Journal of Computer and Communications **3**(5), 164–173 (2015). https://doi.org/10.4236/jcc.2015.35021
45. Maggi, F., Vosseler, R., Quarta, D.: The Fragility of Industrial IoT's Data Backbone: Security and Privacy Issues in MQTT and CoAP Protocols. White paper, Trend Micro Inc. (2018)

46. Mai, A., Schedler, O., Weippl, E., Krombholz, K.: Are HTTPS Configurations Still a Challenge?: Validating Theories of Administrators' Difficulties with TLS Configurations. In: Proceedings of the 4th International Conference on HCI for Cybersecurity, Privacy and Trust (HCI-CPT '22). vol. 13333, pp. 173–193. Springer (2022). https://doi.org/10.1007/978-3-031-05563-8_12
47. Malani, S., Srinivas, J., Das, A.K., Srinathan, K., Jo, M.: Certificate-Based Anonymous Device Access Control Scheme for IoT Environment. IEEE Internet of Things Journal **6**(6), 9762–9773 (2019). https://doi.org/10.1109/JIOT.2019.2931372
48. Padmanabhan, R., Rula, J.P., Richter, P., Strowes, S.D., Dainotti, A.: DynamIPs: analyzing address assignment practices in IPv4 and IPv6. In: Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '20). pp. 55–70. ACM (2020). https://doi.org/10.1145/3386367.3431314
49. Paracha, M.T., Dubois, D.J., Vallina-Rodriguez, N., Choffnes, D.: IoTLS: Understanding TLS Usage in Consumer IoT Devices. In: Proceedings of the 21st ACM Internet Measurement Conference (IMC '21). pp. 165–178. ACM (2021). https://doi.org/10.1145/3487552.3487830
50. Pritikin, M., Yee, P.E., Harkins, D.: Enrollment over Secure Transport. RFC 7030 (2013). https://doi.org/10.17487/RFC7030
51. Redmiles, E.M., Liu, E., Mazurek, M.L.: You Want Me To Do What? A Design Study of Two-Factor Authentication Messages. In: Proceedings of the 13th Symposium on Usable Privacy and Security (SOUPS '17). USENIX Association (2017)
52. Reese, K., Smith, T., Dutson, J., Armknecht, J., et al.: A Usability Study of Five Two-Factor Authentication Methods. In: Proceedings of the 15th Symposium on Usable Privacy and Security (SOUPS '19). pp. 357–370. USENIX Association (2019)
53. Rescorla, E., Tschofenig, H., Modadugu, N.: The Datagram Transport Layer Security (DTLS) Protocol Version 1.3. RFC 9147 (2022). https://doi.org/10.17487/RFC9147
54. Rose, K., Eldridge, S., Chapin, L.: The Internet of Things: An Overview. Tech. Rep. report-InternetofThings-20151015-en, The Internet Society (ISOC) (2015)
55. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., et al.: Modeling Attacks on Physical Unclonable Functions. In: Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10). pp. 237–249. ACM (2010). https://doi.org/10.1145/1866307.1866335
56. Salman, O., Abdallah, S., Elhajj, I.H., Chehab, A., Kayssi, A.: Identity-based authentication scheme for the Internet of Things. In: Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC '16). pp. 1109–1111. IEEE (2016). https://doi.org/10.1109/ISCC.2016.7543884
57. Sciancalepore, S., Capossele, A., Piro, G., Boggia, G., Bianchi, G.: Key Management Protocol with Implicit Certificates for IoT Systems. In: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems (IoT-Sys '15). pp. 37–42. ACM (2015). https://doi.org/10.1145/2753476.2753477
58. Shekari, T., Cardenas, A.A., Beyah, R.: MaDIoT 2.0: Modern High-Wattage IoT Botnet Attacks and Defenses. In: Proceedings of the 31st USENIX Security Symposium (SEC '22). pp. 3539–3556. USENIX Association (2022)
59. Shelby, Z., Hartke, K., Bormann, C.: The Constrained Application Protocol (CoAP). RFC 7252 (2014). https://doi.org/10.17487/RFC7252
60. Soltan, S., Mittal, P., Poor, H.V.: BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid. In: Proceedings of the 27th USENIX Security Symposium (SEC '18). pp. 15–32. USENIX Association (2018)

61. Srinivasa, S., Pedersen, J.M., Vasilomanolakis, E.: Open for Hire: Attack Trends and Misconfiguration Pitfalls of IoT Devices. In: Proceedings of the 21st ACM Internet Measurement Conference (IMC '21). pp. 195–215. ACM (2021). https://doi.org/10.1145/3487552.3487833
62. van der Stok, P., Kampanakis, P., Richardson, M., Raza, S.: EST-coaps: Enrollment over Secure Transport with the Secure Constrained Application Protocol. RFC 9148 (2022). https://doi.org/10.17487/RFC9148
63. Tanveer, M., Alkhayyat, A., Khan, A.U., Kumar, N., Alharbi, A.G.: REAP-IIoT: Resource-Efficient Authentication Protocol for the Industrial Internet of Things. IEEE Internet of Things Journal **9**(23), 24453–24465 (2022). https://doi.org/10.1109/JIOT.2022.3188711
64. Tanveer, M., Badshah, A., Alasmary, H., Chaudhry, S.A.: CMAF-IIoT: Chaotic map-based authentication framework for Industrial Internet of Things. Internet of Things **23**, 100902 (2023). https://doi.org/10.1016/j.iot.2023.100902
65. Tiefenau, C., von Zezschwitz, E., Häring, M., Krombholz, K., Smith, M.: A Usability Evaluation of Let's Encrypt and Certbot: Usable Security Done Right. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19). pp. 1971–1988. ACM (2019). https://doi.org/10.1145/3319535.3363220
66. Wan, Y., Xu, K., Wang, F., Xue, G.: IoTMosaic: Inferring User Activities from IoT Network Traffic in Smart Homes. In: Proceedings of the 41st IEEE Conference on Computer Communications (INFOCOM '22). pp. 370–379. IEEE (2022). https://doi.org/10.1109/INFOCOM48880.2022.9796908
67. Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., Kivinen, T.: Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 7250 (2014). https://doi.org/10.17487/RFC7250
68. Xie, Y., Yu, F., Achan, K., Gillum, E., et al.: How dynamic are IP addresses? ACM SIGCOMM Computer Communication Review **37**(4), 301–312 (2007). https://doi.org/10.1145/1282427.1282415
69. Xu, L.D., He, W., Li, S.: Internet of Things in Industries: A Survey. IEEE Transactions on Industrial Informatics **10**(4), 2233–2243 (2014). https://doi.org/10.1109/TII.2014.2300753
70. Yang, K., Li, Q., Sun, L.: Towards automatic fingerprinting of IoT devices in the cyberspace. Computer Networks **148**, 318–327 (2019). https://doi.org/10.1016/j.comnet.2018.11.013
71. Z-Wave Alliance: Application Work Group Z-Wave Specifications. Tech. Rep. Release 2023B, Z-Wave Alliance (2023)
72. Zhang, S., Sangdeh, P.K., Pirayesh, H., Zeng, H., et al.: AuthIoT: A Transferable Wireless Authentication Scheme for IoT Devices Without Input Interface. IEEE Internet of Things Journal **9**(22), 23072–23085 (2022). https://doi.org/10.1109/JIOT.2022.3186646
73. ZigBee Alliance: ZigBee Specification. Tech. Rep. 05-3474-21, ZigBee Alliance (2015)

A User Study Details

In the course of our research, we performed a user study to get an intuition on how usable LUA-IoT is. To this end, we first asked the 23 participants to configure an IoT device during an online task (Appendix A.1) and subsequently to perform a survey allowing to analyze the participant's perception (Appendix A.2). Thereby, our studies are guided by our ethical considerations (cf. Section 8.3).

A.1 IoT Device Configuration Task

When clicking on the link to our study in the email we wrote for recruitment, the participants directly can read the task description (Appendix A.1). After acknowledging our briefing, a tutorial starts to make the participants familiar with our study tool (Appendix A.1). Subsequently, after completing the tutorial, the participants can work on the task, i.e., configure their IoT device using the smart phone up to the point until they can switch on the TV using their smart phone (cf. Section 8.1).

Task Description The task description is presented on the start screen of our study together with a picture of the smart plug to configure and information on the study's duration, that decisions done in the study have no influence on the outside world, and that the participants should not type in any personal passwords.

> **Welcome to our Study on the Setup Procedure of a Smart Home Device**
>
> The goal of this study is the evaluation of current IoT setup procedures. The study is divided into three parts. The first part consists of a tutorial that helps you to understand the different components, that are present in the second part.
> In the second part of the study, you will set up the smart home device on the right-hand side. The task is completed once you integrated the device into your "virtual" home Wi-Fi and toggled the smart power plug once.
> Once you complete the task to set up the smart power plug, there are some final questions waiting for you. The question part automatically starts when you complete the setup of the smart home device. The final questions should only take a few minutes.

Tutorial To lower the burden of our online task in comparison to a real-life study and allow the participants to familiarize themselves with the study interface, we lead the participants through a tutorial that includes fundamental features to (physically) control the virtual power plug, its manual and the smartphone. The tutorial requires the participants to complete the following steps:

1. Insert the power plug into the socket.
2. Remove the power plug from the socket.
3. Turn the power plug to the left.
4. Turn the power plug to the right.
5. Go to the next page of the manual.

6. Go to the previous page of the manual.
7. Open and close one app on the smartphone.

A.2 Survey

In addition to general demographic questions, we asked the participants of our survey to answer the following questions. We split the questions by their scope, i.e., we asked general questions regarding smart homes to assess the knowledge participants already have and more detailed questions regarding our approach to check for its acceptance.

For some questions, it depends on the answer to a previous question or the participant being in the control group or not to show up. We mark these questions accordingly. Additionally, some questions allow a single answer only (answers marked with ○), and others allow multiple answers (□).

General Smart Home Questions

Q1: I already have experience with . . .

□ setting up smart home devices for myself.
□ setting up smart home devices for other persons.
□ using smart home devices.
□ neither the setup nor the usage of smart home devices.

If already set up for themselves or others

Q2: How often did you set up a device for yourself or other persons?

○ 1 ○ 2 ○ 3 ○ 4 ○ 5+

Q3: I used the following applications to set up a smart home device . . .

□ Apple Home
□ Google Home
□ Smart Life
□ Philips Hue
□ Bosch Smart Home
□ Amazon Alexa
□ Samsung Smart Things
□ Other (free text)

Q4: Do you find it annoying that there are several apps to set up an IoT device?

○ Yes ○ Neutral ○ No

Q5: Did you ever have the possibility to decide whether your device should use secure communication?

○ Yes ○ No ○ I don't know

If already used devices

Q6: When did you last use a smart home device?

○ In the last week.

○ In the last month, but not in the last week.

○ In the last six months, but not in the last month.

○ I don't know/remember.

Q7: Do you think that your devices communicate securely so that an attacker cannot read the content of a message if the attacker has access to the sent messages?

○ Yes ○ Maybe ○ No

Q8: Would you feel safer if your devices would encrypt the messages they exchange?

○ Yes ○ Maybe ○ No

Q9: Do you think that some smart home devices have access to sensitive information?

○ Yes ○ No ○ I don't know

If thinking that devices have access to sensitive information

Q10: Please specify the sensitive information that some smart home devices have access to:

free text

Q11: Please select the devices that have access to sensitive information:

□ Smart Power Plug
□ Camera
□ Smart Home Hub
□ Speakers (Amazon Echo, Home Pod)
□ Other (free text)

Q12: Do you think that attackers could specifically target IoT devices that have access to sensitive information?

○ Yes ○ No ○ I don't know

Risk Assessment

Imagine that you are living in a home that is equipped with every possible smart home device. Please answer if the following scenarios would be more or less concerning than the scenario where an attacker could have access to all the information the devices exchange since they do not use secure communication.

Q13: Losing my credit card or somebody having access to my bank account would be ... concerning.

○ more ○ less ○ equally

Q14: Somebody getting access to my email, telephone number, address, or name would be ... concerning.

○ more ○ less ○ equally

Q15: Leakage of harmless private video footage of me would be ... concerning.

○ more ○ less ○ equally

Questions on the Task

If already set up for themselves or others

Q16: How close was the task to set up this exemplary smart power plug to the setup processes you know?

○ Very close ○ Close ○ Neutral
○ Deviating ○ Very deviating

Q17: Overall the task to set up the smart power plug was?

○ Very easy ○ Easy ○ Neutral
○ Difficult ○ Very difficult

Q18: What challenges did you experience during the task to set up the smart power plug?

free text

Q19: I already have experience with . . .

□ self-signed certificates. □ certificate authority-signed certificates.

LUA-IoT Group You had the option to register the device at a certificate authority, and afterward, your device would request a certificate and communicate securely with other devices. There also exists another setup process where you need to enter a valid certificate and private key during the setup process to enable secure communication for your device.

Q20: Which option do you prefer to use?

○ Registration at a CA. ○ Inserting certificate and private key.

Q21: Would you use your chosen option to enable secure communication in your devices?

○ Yes ○ No ○ Depends on device

Q22: Which option is easier in your opinion?

○ Registration at a CA. ○ Inserting certificate and private key.

Q23A: Did you feel comfortable sharing your email address with the certificate authority?

○ Yes ○ No

Q24A: Would you prefer to use your telephone number or postal address to confirm your account?

○ Yes, telephone number
○ Yes, postal address
○ Yes, both
○ None

Q25A: What could be improved about the registration at the certificate authority?

free text

If secure communication was enabled successfully

Q26A: Why did you decide to use secure communication?

free text

If secure communication was not enabled

Q27A: Why did you decide against secure communication?

free text

Control Group You had the option to enter a valid certificate and private key during the setup process to enable secure communication for your device. There also exists another setup process where you need to register the device at a certificate authority with an identifier, and afterward, your device would request a certificate and communicate securely with other devices.

Q20: Which option do you prefer to use?

○ Registration at a CA. ○ Inserting certificate and private key.

Q21: Would you use your chosen option to enable secure communication in your devices?

○ Yes ○ No ○ Depends on device

Q22: Which option is easier in your opinion?

○ Registration at a CA. ○ Inserting certificate and private key.

Q23B: How difficult was the creation of the certificate and private key for you?

○ Very easy ○ Easy ○ Neutral
○ Difficult ○ Very difficult

Q24B: Please select the tools that helped you to create the certificate and private key.

☐ Video
☐ Website tutorial
☐ Created using an online tool
☐ Asked a colleague
☐ Terminal
☐ Other (free text)

If secure communication was enabled successfully

Q25B: Why did you decide to use secure communication?

free text

If secure communication was not enabled

Q26B: Why did you decide against secure communication?

free text