Automated Authentication Credential Derivation for the Secured Configuration of IoT Devices

Thomas Ulz, Thomas Pieber, Christian Steger
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
{thomas.ulz, thomas.pieber, steger}@tugraz.at

Andrea Höller, Sarah Haas, Rainer Matischek
Design Center Graz
Infineon Technologies Austria AG
Graz, Austria
{andrea.hoeller, sarah.haas, rainer.matischek}@infineon.com

***Abstract*—The number of embedded systems and resource constrained devices is steadily increasing due to trends such as the Internet of Things (IoT) and the Industrial IoT. With that, also the frequency and extent of cyber-attacks that target these systems are rapidly increasing. Not only are most devices not adequately secured against such attacks, but also use default settings and authentication credentials. These problems are especially critical if the devices are deployed in industrial contexts where managing configurations and authentication credentials is a complex and inconvenient process. Therefore, in this paper, we present a device configuration approach that automatically derives authentication credentials from device configurations. We demonstrate that the additional performance required by our approach is acceptable while the provided security is reasonable when compared to traditional authentication approaches such as passwords. The security benefits of our approach are highlighted by an extensive security and threat analysis that demonstrates that 9 out of 10 identified threats are mitigated by our approach.**

***Index Terms*—Embedded Security; Device Configuration; Automated Credential Derivation; Authenticated Key Exchange.**

I. Introduction

Internet of Things (IoT) devices are constantly exposed to potential adversaries and threats due to them being connected to the Internet continuously [1], [2]. Not only are privately used IoT devices targeted by attacks [3], but also IoT devices used in industrial contexts. Providing security for such Industrial IoT (IIoT) devices is especially crucial since security weaknesses might reveal confidential information, harm industrial processes, or in extreme cases, might cause physical damage and threaten human lives [4]. In addition to these safety critical issues, also privacy concerns due to industrial espionage caused by infeasible IoT device security need to be considered [5]. Several studies have shown that among the security weaknesses of IoT devices, using weak or even default authentication credentials is one of the primary reason for successful attacks [6], [7], [8]. Cam-Winget et al. [9] point out that very often remote access channels used for firmware and configuration updates are vulnerable to such weaknesses. In order to mitigate these issues, using sophisticated authentication mechanisms such as two-factor authentication could be one possible solution. However, due to most IoT devices being resource constrained, applying such concepts will not be possible [10]. Thus, most systems still rely on traditional authentication credentials such as username and password combinations. In order to increase the security of IoT devices still using such authentication credentials, passwords must frequently be changed while complying with password composition policies (i.e., requiring symbols and numbers in passwords). However, enforcing such policies often leads to even weaker passwords being chosen by users [11].

Therefore, in this paper, we present an automated credential derivation process used for secured configuration of IoT devices. To alleviate users of the need to choose sophisticated authentication credentials, our proposed approach will derive these credentials from previously applied configuration updates. As an example, let us consider a simple WiFi door sensor that can be used for monitoring purposes. We assume the only configuration parameters of such a sensor are its sampling frequency (`SINT`), a username (`USER`), and the corresponding password (`PASSWORD`). Fig. 1 shows an exemplary initial configuration C_0 in which `SINT` is set to an interval of 10 minutes and no `PASSWORD` for the default `USERNAME` root is configured. In this example, a user might change the sampling interval `SINT` but not the default `PASSWORD`, as highlighted in our example. In contrast to that, if our proposed approach is applied, updating `SINT` from configuration C'_0 to the value in C'_1, will also cause the default `PASSWORD` to change.

To obviate the need for users to remember these automatically generated authentication credentials, we demonstrate two mechanisms that are used to manage the authentication credentials for IoT devices. We also propose a hardware architecture that is capable of providing tamper resistance to protect confidential information. To demonstrate the provided security level of our proposed approach, we will compare the achievable authentication credential strength to a traditional password-based approach.

Contributions. In this paper, we present an automated authentication credential derivation process that improves the security of IoT devices while not complicating their usage. Authentication credentials are automatically derived whenever a configuration update is performed. Thus, insecure default passwords are changed as soon as the device is configured for its first use. Configurations are managed by a central instance, such that users do not need to remember their authentication credentials. To the best knowledge of the authors, no such contribution was previously made.

978-1-5386-4155-2/18/$31.00

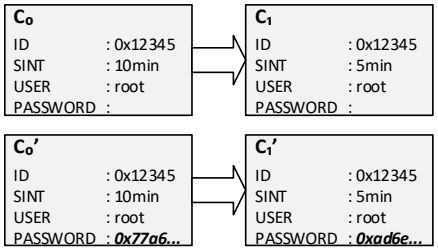

Fig. 1. Example configuration update changing the sampling interval `SINT` only ($C_0 \rightarrow C_1$) compared to using our proposed approach where the same change would trigger the automated credential derivation ($C'_0 \rightarrow C'_1$).

TABLE I
COMPARING THE ATTRIBUTES CONFIGURATION MANAGEMENT (MANAGED), SUFFICIENT SECURITY (SECURED), AND AUTOMATED CREDENTIAL DERIVATION (CRED. DERIV.) WITH RELATED WORK.

Related work	Managed	Secured	Cred. deriv.
Perera et al. [22]	✓	✗	✗
Nastic et al. [23]	✓	✗	✗
Perumal et al. [24]	✓	✗	✗
Santoso and Vun [25]	✓	✓	✗
Ulz et al. [26]	✓	✓	✗
This work	✓	✓	✓

Outline. The remainder of this paper is organized as follows. In Section II we give background information on technologies used in our approach and discuss related work. We then define our system model and discuss assumptions made regarding this model in Section III. Our proposed automated credential derivation process is then presented in Section IV and evaluated in Section V. We then conclude this paper in Section VI where also potential future work is discussed.

II. BACKGROUND AND RELATED WORK

In this section, we give background information on technologies involved in our proposed approach, as well as discuss related work for IoT device configuration.

A. Key Agreement Protocols

Key agreement protocols are used to perform key agreement between two or more communication partners over an unsecured channel such as the Internet. Usually, during key agreement, all involved partners can influence the key agreement process. The final key is composed of influences from all involved partners without revealing the key to any adversary that is capable of eavesdropping communication over the unsecured communication channel. One of the most widely used key agreement protocols, Diffie-Hellman (DH) [12], is used in the Transport Layer Security (TLS) protocol.

Encrypted key exchange (EKE) protocols belong to the category of key agreement methods that use passwords to authenticate the partners involved in the key agreement process [13]. The password is used as shared knowledge between involved partners and is incorporated into the key agreement process such that only partners that are in possession of the correct password can mutually agree on a key. The resulting session key is considered to be appropriately secure even if the shared knowledge is drawn from a small set of values. In the Simple Password-Based Encrypted Key Exchange (SPAKE) protocol [14] a modified DH algorithm that uses a shared password for key derivation during key agreement is used.

B. Tamper Resistant Hardware

Tamper resistant hardware provides a secured execution environment (SEE) as well as secured storage. Therefore, these components can be used for the execution of critical code parts such as cryptographic algorithms and for storing confidential information such as key material. A device that is labeled as tamper resistant [15] is capable of mitigating physical attacks such as non-invasive and invasive side-channel attacks [16] by applying appropriate countermeasures. The level of security that is provided by a certain SE can be assessed based on the common criteria (CC) information technology security evaluation [17], such that SEs can be compared based on their provided security level.

C. Authentication for IoT Devices

Jan et al. [18] state that authenticating devices before communicating with these devices is critical, especially in the IoT where a high number of potentially unsecured devices are present. The authors, however, highlight that due to most devices being resource constrained, no complex cryptographic operations can be performed. For example, an approach that performs mutual authenticated Diffie-Hellman key exchange using public keys by Xu et al. [19] might be infeasible due to the need to store many public keys. Roman et al. [20] state that an infrastructure for mutual authentication for IoT devices will be needed to account for such resource constrained devices. The authors also discuss an important principle that is applied in this paper: system security for constrained devices should rely on *what I have* and *what I know*. Liu et al. [21] discuss authentication protocols for the IoT and state that such a protocol has several tasks, one of them being *key switching*.

D. Related Work Secured Device Configuration

Configuring devices in the IoT is an active topic in research due to the various challenges presented by the large number of resource constrained devices. One approach to handle the large number of devices that need to be configured is to use self-configuration mechanisms [27], [28]. If self-configuration is not applicable, manual configuration processes, as well as initial provisioning methods need to be secured and simplified as stated by Truong et al. [29]. To support the configuration process, Nastic et al. [23] suggested using a central configuration management solution. Perumal et al. [24] presented an IoT device management framework that is suitable for smart home scenarios. In this framework, IoT device can be managed by a smartphone. However, configurations are stored and transferred unprotected. Regarding the distribution of configuration updates, using the Internet is the most common approach [23], [22], and thus, security needs to be considered. However, most solutions do not consider complete system

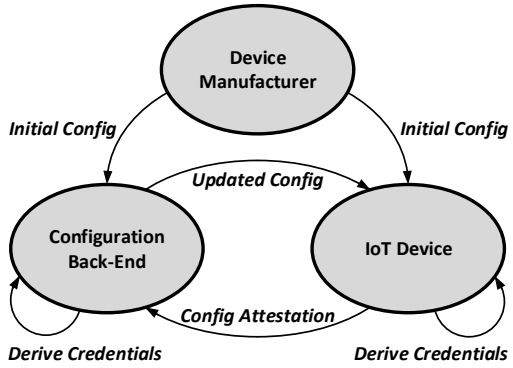

Fig. 2. Proposed principle of using configuration updates for automated credential derivation. Configuration updates are performed by different entities during an IoT device's entire lifecycle.

security (protocol, device, and overall system) but rather cover specific security aspects. Santoso and Vun [25] presented a secured configuration architecture for smart home appliances. The approach relies on mutual authentication based on a pre-shared secret. In their approach, a smartphone is used to manage existing configurations. However, the authors did not elaborate on how this shared secret is initially transferred to the device. They also did not specify, if a shared secret can be changed, for instance, if a device is resold. Ulz et al. [26] demonstrated an approach based on Near Field Communication (NFC) and dedicated hardware security elements. This approach only allows configuration updates via NFC and includes a secured communication protocol. However, the used symmetric cryptography results in a key distribution problem that was not covered by the authors. Still, since this approach is very promising, we extend it by the automated authentication credential derivation that is presented in this paper. A summary of related work is shown in TABLE I.

III. System Model and Assumptions

For our automated authentication credential derivation and IoT device configuration approach, we assume a system model that comprises the three entities shown in Fig. 2.

The **IoT Device** is the device for which configuration updates secured by automatically generated authentication credentials should be performed. On this device, the configuration update process, as well as the configuration data, need to be protected by appropriate security measures. Also, the authentication credential derivation process needs to be protected by appropriate security measures. We assume that potential attackers will be able to gain physical access to this device; therefore, appropriate countermeasures to protect confidential information need to be taken.

The **Device Manufacturer (DM)** produces the IoT device. Since in most cases devices are shipped pre-configured, the DM is responsible for applying initial configurations that lead to automatically generated initial authentication credentials as well. We assume the DM trustworthy.

The **Configuration Back-End (CBE)** is used to manage device configurations for an arbitrary number of IoT devices. Any configuration change except the *initial configuration* is initiated from this entity. The initial configuration is applied by the DM, and thus, this information needs to be imported into the CBE in our approach. We assume the CBE to be adequately secured against any type of attack. That is, the confidential information that is stored there is assumed to be protected against security breaches.

As can be seen in the system model shown in Fig. 2, any update process initiated from the CBE triggers a configuration attestation process from the IoT device. We assume such an attestation process that is capable of attesting the currently applied configuration to a remote instance to be existent in our system model, since the focus of this paper is on the automated credential derivation process. Configuration attestation processes that are suitable for IoT devices have been presented in literature [26], [30], [31].

IV. Automated Credential Derivation

In this section, our proposed automated authentication credential derivation process for the secured configuration of IoT devices is presented. We will discuss the basic process and the session key generation. After that, our approach is compared to traditional password-based authentication. We then list mechanisms that are specific to private or industrial use of IoT devices. Finally, we briefly discuss the hardware architecture we propose for a secured IoT device configuration process.

A. Basic Process

To automate the authentication credential derivation process, applied configurations of IoT devices are used in our approach. Any configuration update will thus trigger the derivation of new authentication credentials. The basic process is shown in Fig. 3. For this example we assume two entities, *Alice* and *Bob* that want to perform a secured configuration update. We assume that both Alice and Bob are in possession of the same shared secret, the k-th iteration of Bob's configuration, C_k. Having this information, Alice and Bob perform a session key generation based on C_k that yields the session key SK_k. Alice then encrypts the configuration update that results in the k+1-th configuration C_{k+1} with this session key and sends it to Bob who is able to decrypt that information using SK_k. After verifying and either applying or rejecting the configuration update, Bob informs Alice about the configuration applied to again establish the same level of shared knowledge. If Bob applies C_{k+1}, he and Alice will be able to generate a session key based on this information. If C_{k+1} is rejected, Alice and Bob still will be able to use C_k as basis for their session key generation process.

Advantages. Using this mechanism entails the following two advantages for users, compared to traditional authentication mechanisms such as passwords:

1) Device security is increased since any configuration change triggers the automated creation of new authen-

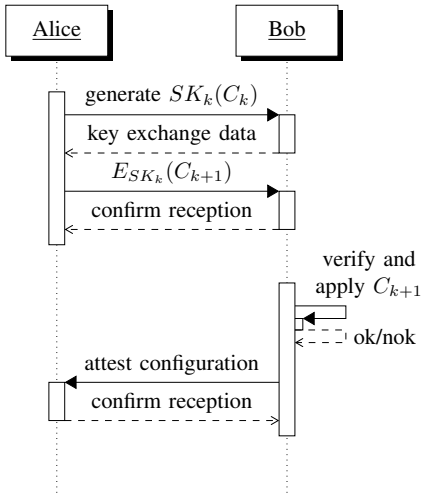

Fig. 3. Sequence diagram demonstrating the configuration update and attestation process where Alice wants to send new configuration data to Bob. Both communicating partners are in possession of knowledge regarding the currently applied configuration and thus, the shared knowledge needed to generate a session key SK.

tication credentials. It is basically impossible to operate devices using default authentication credentials.

2) Users do not need to remember sophisticated passwords since authentication credentials are automatically derived from device configurations which are managed by CBE. Thus, users basically use the CBE as a password manager.

B. Session Key Generation

The session key generation shown in the sequence diagram in Fig. 3 will be performed by using SPAKE2 [14] which is an EKE protocol. This protocol uses a username and password combination in the key derivation of the session key generation process. The protocol is shown in Fig. 4. Since SPAKE2 relies on Elliptic-Curve Diffie-Hellman (ECDH) [32] the ECC generator point G and a one-way function $H(\cdot)$ are defined as public parameters between *Alice* and *Bob*. After that, the respective user identities u_A and u_B as well as a shared secret p are used in the key derivation process to generate a session key SK. It is important to distinguish two types of secret in this information flow: (i) the shared secret (p) that is used for mutual authentication between involved parties and (ii) the session key (SK) that is generated by the algorithm to encrypt subsequent communication. A new shared secret p' could then be transferred using the encrypted channel.

Modifications: If we now apply the SPAKE2 algorithm to our process previously defined in Fig. 3, only a minor adoption to the algorithm needs to be made. For the session key SK to be dependent on the currently applied configuration C_k, we redefine the shared secret between Alice and Bob as

$$p := H(C_k).$$

By applying this definition, each session key generated will be dependent on C_k. However, defining C_k as shared

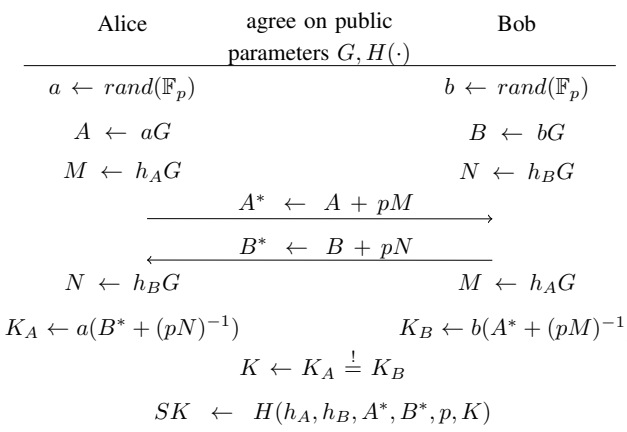

Fig. 4. SPAKE2 protocol [14] that is modified in our presented approach. In this protocol, G is the ECC generator point, $H(\cdot)$ is a one-way function, h_A and h_B are Alice's and Bob's hashed identities respectively, and p is a shared secret between Alice and Bob.

secret in the EKE process also implies that C_k is used for mutual authentication between Alice and Bob. Therefore, configuration data needs to be treated as confidential information as was assumed in our system model. The advantage of this approach is that any change to a configuration made will automatically trigger an authentication derivation process based on the new configuration and thus, will mitigate the problem of vulnerable IoT devices due to relying on default username and password combinations.

C. Comparison to Password-Based Authentication

Assumptions. Since we are relying on the security provided by the SPAKE2 algorithm, we refer to the corresponding security proof by Abdalla and Pointcheval [14]. Further, we assume that the device is using the hardware architecture proposed in Section IV-E and thus, provides sufficient countermeasures to mitigate physical or side-channel attacks.

To evaluate the achievable level of security provided by our presented approach, we will discuss the strength of our authentication credentials by applying the Password Quality Indicator (PQI) presented by Ma et al. [33]. The authors discuss why entropy alone cannot be used as a quality indicator for passwords and define the $PQI(D, L)$ where D is the Levenshtein distance between two strings and L is the effective password length. According to Ma et al. a credentials are considered *good* if $D \geq 3$ and $L \geq 14$. Since in our approach authentication credentials are automatically derived from configurations, we need to apply these parameters to configuration data.

Ma et al. [33] define D as the Levenshtein distance between passwords and a dictionary of words. To be applicable for our evaluation, we not only need to consider a dictionary but a set of *observable parameters*. Such parameters could include, for example, a WiFi name. Both D and L are then applied to the values of configuration key-value pairs only. Applied to the simple example shown in Fig. 1, we would use

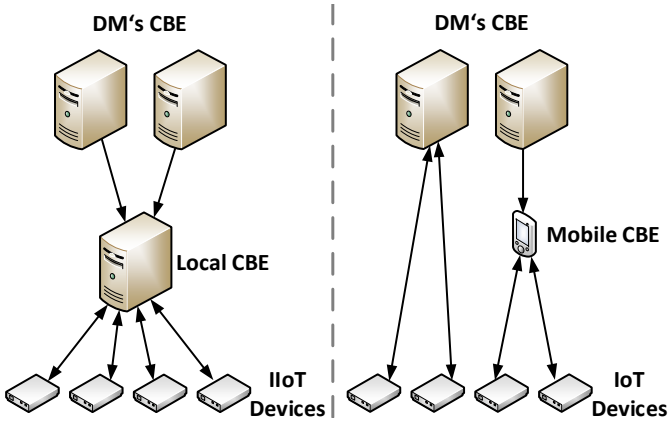

Fig. 5. Context specific mechanisms for configuration management. A local CBE is deployed to manage IIoT device configurations. Consumer IoT devices are either managed by the manufacturer's global CBE or by a mobile CBE.

a set of $\{wifi1, root\}$ for authentication credential derivation. Obviously, this results in $D = 1$ and $L = 9$ which is considered insecure by our measure. In comparison, passwords with a length of 9 characters and a Levenshtein distance of 1 can be considered quite common (e.g., *password1*). This shows that in general, our proposed approach would not be more secure than relying on user-defined passwords. To mitigate this problem, a configuration update could be automatically extended by random data that can be securely generated by a true random number generator (TRNG) [34] provided by most Secure Elements (SEs). In our approach, we suggest to extend configurations that are created at the CBE with a so-called nonce. The nonce is generated by the CBE and added as a configuration parameter to the encrypted configuration. The IoT device then is capable of extracting the transferred nonce from applied configuration updates. By doing so, both D and L can be increased to an arbitrary length. Similar approaches have been shown to enhance the security of password-based authentication methods [35].

D. Context Specific Mechanisms

Depending on the context in which an IoT device is operating, different mechanisms for device configuration management and password reset are required. Therefore, we propose different system architectures that are shown in Fig. 5. In *industrial scenarios* where IIoT device configurations contain confidential information that needs to be kept private, a local CBE can be deployed in an internal network. As can be seen in the left half of Fig. 5, such a local CBE needs to import initial configurations from the respective DMs. After that, the previously discussed device configuration and update process is performed between managed IIoT devices and the local CBE only. Since the local CBE can be viewed as a single point of failure, appropriate measures to properly secure information need to be taken. In *personal settings* it is infeasible to deploy a local CBE. Therefore, we propose two different system architectures that are shown in the right half of Fig. 5. On the one hand, a DM's CBE can be used for device configuration management. In this case, the configuration update and attestation process is performed between these two entities. If similar to the industrial scenario confidential configuration data needs to be kept private, we propose the use of a mobile CBE, such as also present in the system model presented by Ulz et al. [26]. Compared to the other two system architectures, data loss might be more probable when using such a mobile CBE. Therefore, a configuration reset and thus, authentication credential reset mechanism needs to be included in our presented approach as well. A *configuration reset* will be required whenever a CBE's configuration database is inconsistent such that the currently applied configuration on the managed device is not known to the configuration database. Such inconsistencies could be caused by loss of data on a mobile or local CBE. As a potential measure, we propose to include a hard-reset method into IoT devices that reset the currently applied configuration to the initial configuration C_0 which can be easily imported again into any CBE. The download of initial configurations could be achieved, for example, by downloading the information from DM's CBE or by applying QR codes to the respective devices that contain the required information. This approach would be similar to current approaches where default credentials are printed on stickers that are attached to the devices.

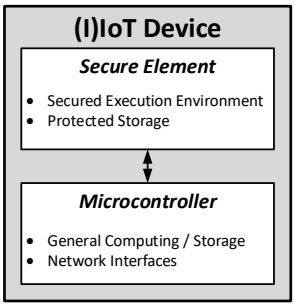

Fig. 6. Proposed hardware architecture for (I)IoT devices.

E. Hardware Architecture

Configurations stored on devices may contain confidential information such as key material or production-relevant information in the case of industrially used devices. To protect this confidential information, appropriate security measures need to be taken in hardware as well. Since attackers might be able to gain physical access, we propose to use the hardware architecture shown in Fig. 6 that suggests including an SE into IoT devices. In our proposed architecture, this SE is responsible for performing security critical operations such as the automated credential derivation presented in this paper. In addition, confidential configuration data is stored in the SE due to its tamper resistant nature. The micro-controller is used for general purpose computing tasks, and thus, a dual-execution principle is applied [36]. In addition, the device's required network interfaces are provided by the micro-controller.

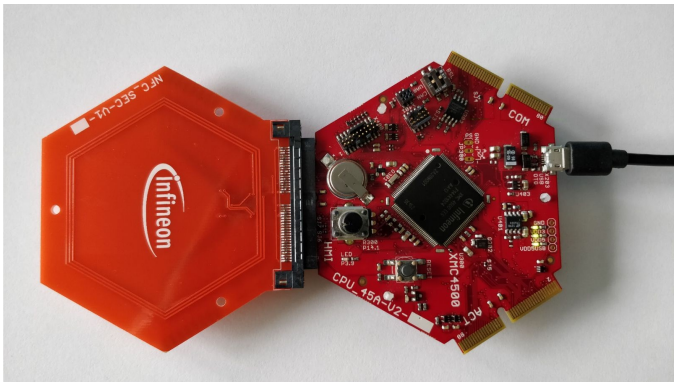

Fig. 7. Research IoT device prototype used for performance evaluation. The upper hexagon shaped board contains the XMC4500 microcontroller, the SLE78 SE is embedded in the lower hexagon shaped board.

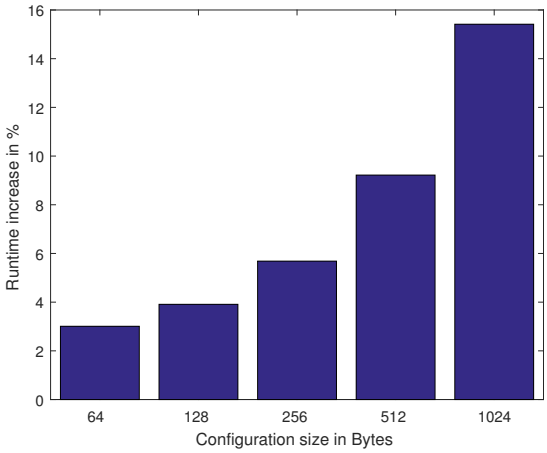

Fig. 8. Relative runtime increase due to applying hash function to configurations of different sizes. As a baseline we use a standard SPAKE2 implementation that requires a 32 Bytes hash operation for the password.

V. Evaluation

The evaluation of our presented credential derivation process is twofold. First, we discuss the overhead compared to traditional authenticated key exchange algorithms in a performance analysis. Second, we also analyse the security properties of our presented approach in a threat analysis.

A. Performance Analysis

To evaluate our proposed approach for automated authentication credential derivation, we use a research prototype according to the hardware architecture shown in Fig. 6. The prototype comprises an *Infineon XMC4500* microcontroller and an *Infineon SLE78* SE. The protoype is shown in Fig. 7. Since in our proposed architecture all security relevant operations are executed on the SE, the complete EKE process is implemented on this controller. To highlight the extent of runtime overhead resulting from the modified SPAKE2 algorithm, we conducted a performance analysis. As shown by Pieber et al. [37], running SPAKE2 on resource constrained hardware such as Infineon's SLE78 results in a larger runtime when compared to traditional ECDH. Depending on the configuration size, our approach requires extended hashing operations compared to the standard SPAKE2 implementation. As a baseline, we consider SPAKE2 with block sizes of 16 Bytes each for username and password, resulting in the hash function being executed on 32 Bytes of data. Fig. 8 shows the resulting overhead when hashing configurations of different sizes instead of a single password. As can be seen there, a configuration of 512 Bytes would result in an increase of runtime of roughly 10% compared to the basic SPAKE2 implementation. However, in their paper, Pieber et al. [37] show that pre-computing the required hash values can reduce the runtime of a SPAKE2 implementation to values similar to traditional ECDH. Of course, pre-computing these values based on applied configurations is also a possibility for our proposed approach. Thus, mitigating the additionally required runtime during session key generation.

B. Threat Analysis

To demonstrate the robustness of our presented approach against various types of attacks, we conduct a threat analysis [38]. In this analysis, we identify the involved *Entities (E)* and *Assets (A)* that need to be protected by our approach. We then list potential *Threats (T)*, and *Countermeasures (C)* that are provided by our approach to mitigate these threats. If a threat is not entirely mitigated by our approach, the residual *Risks (R)* are also listed. In addition, we categorize all threats according to the STRIDE threat model [39]. Threats are categorized by their potential impact, according to the following criteria: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. Although we do not claim that our presented threat analysis is exhaustive, we do think that it adequately highlights the security of our proposed automated authentication credential derivation process.

Entities. The following entities are identified in our automated authentication derivation process. If necessary, we list assumptions regarding the respective entity to narrow the scope of this threat analysis.

- **(E1)** The IoT or IIoT device that is being configured.
- **(E2)** The DM's CBE. Since the security aspects of a CBE are out of scope for this paper, we assume that the DM's CBE is sufficiently secured against attacks such that no loss of confidential data will occur there.
- **(E3)** The device user's local or mobile CBE. We also assume that the users's CBE is sufficiently secured against attacks such that no confidential data will be lost by attacks targeting the user's CBE.
- **(E4)** A potential adversary. We do not make any assumption about the extend of attacks an adversary is able to perform. That is, we assume the adversary is able to perform remote attacks as well as physical attacks.

Assets. We identify the following assets that need to be protected by our proposed approach.

(A1) The IoT device itself must be protected from malicious actions that might be enabled by security weaknesses.

(A2) The configuration data that is transferred must be protected since it might contain confidential information such as keys or production relevant information for IIoT devices.

Threats. After identifying involved entities and assets that need to be protected, we are going to list identified threats, categorize them based on the STRIDE threat model, and list corresponding countermeasures or residual risks.

(T1) An adversary might be able to eavesdrop transferred data, and thus, be able to learn confidential information.
STRIDE: I
(C1) Transferred configuration data is encrypted using session keys. Therefore, data confidentiality is provided.

(T2) An adversary might act as man-in-the-middle (MITM) and impersonate the CBE and the IoT device respectively.
STRIDE: S, T, R, I
(C2) When using the generated session key for authenticated encryption, data confidentiality, integrity, and authenticity can be provided.

(T3) The adversary can act as MITM during key agreement.
STRIDE: S, T, R, I, E
(C3) Since our approach is based on SPAKE2, MITM attacks are mitigated by mutual authentication and the DH principle.

(T4) An adversary easily can learn the initial configuration C_0, record all data transfers and thus, infer any subsequent authentication credential.
STRIDE: S, T, R, I, E
(C4) Since session keys that cannot be learned by the adversary are used to protect transferred data, the adversary cannot learn any subsequent authentication credential.

(T5) An adversary might learn a configuration C_k by observing the IoT device's environment and behaviour. The adversary then is able to infer the current session key.
STRIDE: S, T, R, I, E
(C5) Random information that is added to configuration data and transferred to the IoT device mitigates this threat.

(T6) The IoT device's user might not change the initial configuration C_0, and thus, no new authentication credential is derived automatically.
STRIDE: S, T, E
(C6) A devices that is running on default configurations will not be useful for the user. For instance, the device must at least be connected to a network.

(T7) An adversary might perform attacks such as trying to provoke buffer overflows to compromise the device and reveal confidential information.
STRIDE: T, I
(C7) Since all cryptographic operations are performed at the SE, and confidential information is also stored there, such attacks are mitigated by the SE's security measures.

(T8) An adversary might perform physical attacks targeting the device to reveal confidential information.
STRIDE: T, I
(C8) In our approach, we are using tamper resistant SE. The SE's level of security is verified by the CC certification process.

(T9) Intentional or unintentional backdoors might exist in software or hardware (e.g. for debugging purposes) that can be exploited by an adversary.
STRIDE: S, T, R, I, D, E
(C9) When including a CC certified SE, the trustworthiness of all hardware and software components of the device need to be verified in a certification process.

(T10) Denial-of-Service (DoS) attacks targeting the proposed configuration interface with automated authentication credential derivation.
STRIDE: D
(C10) Since all cryptographic operations are performed at the SE, normal operation of the IoT device is not influenced by DoS that target the configuration interface.
(R10) However, a residual risk remains, since such DoS attacks will of course drain the device's battery by triggering operations at the SE.

VI. Conclusions and Future Work

In this paper, we present a novel automated authentication credential derivation process that is suited for personal as well as industrial used IoT devices. Instead of relying on users to change authentication credentials, configuration updates trigger the automated derivation of new authentication credentials. To increase the usability of our approach, we do not require users to remember these authentication credentials any more. We propose a system architecture that, besides managing configurations, also keeps track of a user's derived authentication credentials. To account for different usage scenario of devices, we present and discuss different configuration update and configuration-reset mechanisms, that we deem suitable for industrial or personal scenarios respectively. Thus, while increasing system security due to automatically triggered authentication credential updates, usability compared to traditional approaches is also improved. The runtime evaluation of our presented approach highlights that the resulting overhead due to using a modified SPAKE2 algorithm is in an acceptable range. The threat analysis then demonstrates, that 9 out of 10 threats can effectively be mitigated by our proposed approach. The only residual risk, DoS attacks that drain the device's battery, cannot be mitigated by any other known approach other than turning the device off.

As future work, we plan to investigate methods for initial key and configuration provisioning at the DM's facility that is capable of protecting this confidential information from the DM that is deploying the data on the device. This would allow customers to pre-configure devices such that the authentication credentials are only known to them to further improve the usefulness of our approach.

ACKNOWLEDGMENT

This work has been performed within the IoSense (http://iosense.eu) project. This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 692480. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Netherlands, Spain, Austria, Belgium, Slovakia. IoSense is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between May 2016 and April 2019. More information: https://iktderzukunft.at/en/.

REFERENCES

[1] R. H. Weber, "Internet of things - new security and privacy challenges," *Computer law & security review*, vol. 26, no. 1, pp. 23–30, 2010.

[2] 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, pp. 2481–2501, 2014.

[3] H. Ning, H. Liu, and L. T. Yang, "Cyberentity Security in the Internet of Things," *Computer*, vol. 46, no. 4, pp. 46–53, 2013.

[4] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling CyberPhysical Systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2012.

[5] L. Da Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[6] M. Patton, E. Gross, R. Chinn, S. Forbis, L. Walker, and H. Chen, "Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT)," in *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*. IEEE, 2014, pp. 232–235.

[7] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin, "Security Analysis on Consumer and Industrial IoT Devices," in *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*. IEEE, 2016, pp. 519–524.

[8] E. Bertino and N. Islam, "Botnets and Internet of Things Security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[9] N. Cam-Winget, A.-R. Sadeghi, and Y. Jin, "Can IoT be Secured: Emerging Challenges in Connecting the Unconnected," in *Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE*. IEEE, 2016, pp. 1–6.

[10] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and Privacy Challenges in Industrial Internet of Things," in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015, pp. 1–6.

[11] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman, "Of Passwords and People: Measuring the Effect of Password-Composition Policies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 2595–2604.

[12] W. Diffie and M. Hellman, "New Directions in Cryptography ," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[13] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks ," in *Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on*. IEEE, 1992, pp. 72–84.

[14] M. Abdalla and D. Pointcheval, "Simple Password-Based Encrypted Key Exchange Protocols," in *CT-RSA, LNCS*, vol. 3376. Springer, pp. 191–208.

[15] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper Resistance Mechanisms for Secure Embedded Systems," in *VLSI Design, 2004. Proceedings. 17th International Conference on*. IEEE, 2004, pp. 605–611.

[16] A. Zankl, H. Seuschek, G. Irazoqui, and B. Gulmezoglu, "Side-Channel Attacks in the Internet of Things," *Solutions for Cyber-Physical Systems Ubiquity*, pp. 325–357, 2017.

[17] D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer standards & interfaces*, vol. 29, no. 2, pp. 244–253, 2007.

[18] M. A. Jan, P. Nanda, X. He, Z. Tan, and R. P. Liu, "A Robust Authentication Scheme for Observing Resources in the Internet of Things Environment," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*. IEEE, 2014, pp. 205–211.

[19] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[20] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[21] J. Liu, Y. Xiao, and C. P. Chen, "Authentication and Access Control in the Internet of Things," in *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*. IEEE, 2012, pp. 588–592.

[22] C. Perera, P. P. Jayaraman, A. Zaslavsky, D. Georgakopoulos, and P. Christen, "Sensor Discovery and Configuration Framework for The Internet of Things Paradigm," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 94–99.

[23] S. Nastic, S. Sehic, D.-H. Le, H.-L. Truong, and S. Dustdar, "Provisioning Software-Defined IoT Cloud Systems," in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*. IEEE, 2014, pp. 288–295.

[24] T. Perumal, S. K. Datta, and C. Bonnet, "IoT Device Management Framework for Smart Home Scenarios," in *Consumer Electronics (GCCE), 2015 IEEE 4th Global Conference on*. IEEE, 2015, pp. 54–55.

[25] F. K. Santoso and N. C. Vun, "Securing IoT for Smart Home System," in *Consumer Electronics (ISCE), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1–2.

[26] T. Ulz, T. Pieber, C. Steger, S. Haas, R. Matischek, and H. Bock, "Hardware-Secured Configuration and Two-Layer Attestation Architecture for Smart Sensors," in *Digital System Design (DSD), 2017 Euromicro Conference on*. IEEE, 2017, pp. 229–236.

[27] I. Chatzigiannakis, H. Hasemann, M. Karnstedt, O. Kleine, A. Kroller, M. Leggieri, D. Pfisterer, K. Romer, and C. Truong, "True Self-Configuration for the IoT," in *Internet of Things (IOT), 2012 3rd International Conference on the*. IEEE, 2012, pp. 9–15.

[28] S.-M. Kim, H.-S. Choi, and W.-S. Rhee, "IoT Home Gateway for Auto-Configuration and Management of MQTT Devices," in *Wireless Sensors (ICWiSe), 2015 IEEE Conference on*. IEEE, 2015, pp. 12–17.

[29] H.-L. Truong and S. Dustdar, "Principles for Engineering IoT Cloud Systems," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 68–76, 2015.

[30] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "SEDA: Scalable Embedded Device Attestation," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 964–975.

[31] A. Ibrahim, A.-R. Sadeghi, G. Tsudik, and S. Zeitouni, "DARPA: Device Attestation Resilient to Physical Attacks," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 171–182.

[32] R. Schroeppel, H. Orman, S. OMalley, and O. Spatscheck, "Fast Key Exchange with Elliptic Curve Systems," *Advances in Cryptology-CRYPT0'95*, pp. 43–56, 1995.

[33] W. Ma, J. Campbell, D. Tran, and D. Kleeman, "Password Entropy and Password Quality," in *Network and System Security (NSS), 2010 4th International Conference on*. IEEE, 2010, pp. 583–587.

[34] B. Sunar, W. J. Martin, and D. R. Stinson, "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks," *IEEE Transactions on computers*, vol. 56, no. 1, 2007.

[35] C.-W. Lin, J.-J. Shen, and M.-S. Hwang, "Security Enhancement for Optimal Strong-Password Authentication Protocol," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 2, pp. 7–12, 2003.

[36] M. Sabt, M. Achemlal, and A. Bouabdallah, "The Dual-Execution-Environment Approach: Analysis and Comparative Evaluation," in *IFIP International Information Security Conference*. Springer, 2015, pp. 557–570.

[37] T. Pieber, T. Ulz, C. Steger, and R. Matischek, "Hardware Secured, Password-based Authentication for Smart Sensors for the Industrial Internet of Things," in *International Conference on Network and System Security*. Springer, 2017, pp. 632–642.

[38] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat Modeling as a Basis for Security Requirement," in *Symposium on requirements engineering for information security (SREIS)*, vol. 2005. Citeseer, 2005, pp. 1–8.

[39] M. Howard and D. LeBlanc, *Writing Secure Code*. Microsoft Press, 2003.