Advanced Device Authentication for the Industrial Internet of Things

Rainer Falk and Steffen Fries
Corporate Technology
Siemens AG
Munich, Germany
e-mail: {rainer.falk|steffen.fries}@siemens.com

Abstract—**Device authentication is an essential security feature to ensure the reliable operation of cyber physical systems and the industrial Internet of Things. Solutions have to be both robust and practical to use. After giving an overview on device authentication options, several proposals for advanced device authentication means are presented to increase the attack robustness of device authentication. A well-known cryptographic device authentication using a symmetric cryptographic key or a digital certificate with a corresponding private key for device authentication can be extended with additional validations to check the device identity. Ideas from advanced human user authentication means like multi-factor authentication, continuous authentication, and secret sharing are applied to enhance device authentication.**

Keywords–device authentication; Internet of Things; embedded security; cyber security.

I. INTRODUCTION

The need for technical information technology (IT) security measures increases rapidly to protect products and solutions from manipulation and reverse engineering [1][2]. The scope of the security considerations is further broadened to also include operational technology (OT) environments, in which IT technology is applied to industrial control systems. Cryptographic IT security mechanisms have been known for many years, and are applied in smart devices (Internet of Things, Cyber Physical Systems, industrial and energy automation systems, operation technology) [3]. Such mechanisms target authentication, system and communication integrity, and confidentiality of data in transit or at rest. Security standards have been developed that define security processes, security requirements, and security solutions [3]. The standard IEC 62443 addresses general industrial and automation control systems and can be applied to different vertical automation systems like factory automation, process automation, or building automation. Also, several security standards and guidelines have been defined specifically for particular vertical application domains [4][5][6]. Examples are ISO/IEC 62351 [4] defining security for energy automation systems, and ISO 15118 [7] that defines security for the charging of electric vehicles.

A central security mechanism is authentication that is required for human users, devices, and for software processes: By authentication, a claimed identity is proven. Authentication of a human user can be performed by verifying something the person knows (e.g., a password), something the person possesses (e.g., a physical authentication token, smart card, or a passport), or something the person is (biometric property, e.g., a fingerprint, voice, iris, or behavior).

Advanced authentication techniques make use of multiple authentication factors, and performing authentication checks continuously during an ongoing, authenticated session. With multi-factor authentication, several independent authentication factors are verified, e.g., a password and an authentication token. This increases the security level of the authentication process as multiple independent authentication factors are verified. With continuous authentication, also called active authentication, the behavior of a user during an authenticated session is monitored to determine if the authenticated user is still the one using the session. This increases the security level of a session after a user has been authenticated. It also helps to improve the user friendliness of a security solution as continuous user authentication is not intrusive to the user as repeated explicit re-authentications would be.

While advanced authentication techniques like multi-factor authentication and continuous authentication are known for human users, it seems that these technologies have not yet been applied for device authentication neither in research work nor in real world deployments.

With ubiquitous machine-oriented communication, e.g., the Internet of Things (IoT) and interconnected cyber physical systems, devices have to be authenticated in a secure way. This paper presents and investigates several approaches for advanced device authentication, being an extended version of [1]. The different approaches can be applied independently or in combination to increase the security level for device authentication. While authentication alone does not ensure a secure overall solution, it is an essential building block to realize secure, robust security architectures for industrial Internet of Things and for automation and control systems in general.

An overview on industrial security resp. secure industrial IoT is given in Section II. After describing single device authentication means in Section III, the combination of authentications is covered in Section IV. The advantages of enhanced device authentication factors to increase the security level of Internet of Things systems and Cyber Physical Systems is investigated in Section V. Section VI summarizes related work. Section VII concludes with a summary and an outlook. Note that the paper investigates different options for providing enhance authentication from a conceptual point of view. The options are discussed in the context of system design and require an implementation as the consequent next step.

II. Industrial Security

Industrial automation control systems (IACS) monitor, and control automation systems in different automation domains, e.g., energy automation, railway automation, or process automation [8]. The main functionality can be summarized on a high level to performing control operations in the physical world using actuators, based on physical measurements obtained by sensors. Automation control systems are using open communication protocols like Ethernet, IP, TCP/UDP, or serial internally, and for communication with external systems (e.g., for monitoring, diagnosis, configuration), realizing an industrial Internet of Things (IoT), or the Web of systems. The term Internet of Things commonly refers to a set of technologies supporting the connection of hitherto stand-alone devices to an IP-based network. These technologies are important enablers for the convergence of today's automation architectures with service-oriented approaches while meeting industry-grade safety, security, reliability, and real-time requirements. As networked automation control systems are exposed to external systems, they have to be protected against attacks to prevent manipulation of control operations.

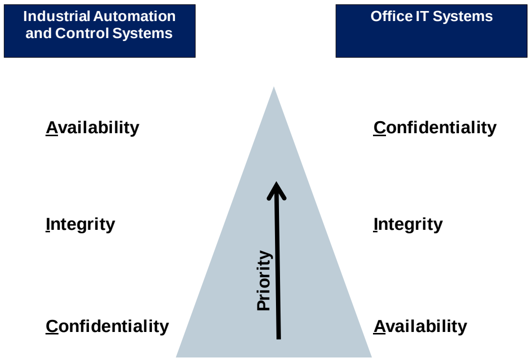

Figure 1. The CIA Pyramid [8]

The three basic security requirements are confidentiality, integrity, and availability. They are also named “CIA” requirements. Fig.1 shows that in common information technology (IT) systems, the priority is “CIA”. However, in automation systems or industrial IT, the priorities are commonly just the other way round: Availability has typically the highest priority, followed by integrity. Confidentiality is often no strong requirement for control communication. Shown graphically, the CIA pyramid is inverted (turned upside down) in many automation systems.

Specific requirements and side conditions of industrial automation systems like high availability, planned configuration (engineering info), long life cycles, unattended operation, real-time operation, and communication, as well as safety requirements have to be considered when designing a security solution. The security requirements, for instance defined in IEC 62443, can be mapped to different automation domains, including energy automation, railway automation, building automation, process automation.

The security standard IEC 62443 [3] defines security for industrial automation control systems. Several parts have been finalized, or are currently in the process of being defined. The different parts cover common definitions, and metrics, requirements on setup of a security organization, and processes, defining technical requirements on a secure system, and to secure system components.

A complex automation system is structured into zones that are connected by so-called “conduits”. For each zone, the targeted security level (SL) is derived from a threat and risk analysis. The threat and risk analysis evaluates the exposure of a zone to attacks as well as the criticality of assets of a zone. While IEC 62443-3-2 defines security levels, and zones for the secure system design, IEC 62443-3-3 describes the requirements to comply with a dedicated security level in an abstract way, not prescribing the actual implementation.

Four security levels have been defined, targeting different categories of attacks:

SL1: Protection against casual, or coincidental violation

SL2: Protection against intentional violation using simple means, low resources, generic skills, low motivation

SL3: Protection against intentional violation using sophisticated means, moderate resources, IACS specific skills, moderate motivation

SL4: Protection against intentional violation using sophisticated means, extended resources, IACS specific skills, high motivation

For each security level, IEC 62443 part 3-3 defines a set of requirements. Seven foundational requirements group specific requirements of a certain category:

- FR 1 Identification and authentication control
- FR 2 Use control
- FR 3 System integrity
- FR 4 Data confidentiality
- FR 5 Restricted data flow
- FR 6 Timely response to events
- FR 7 Resource availability

The security standard IEC62443 part 3.3 states several requirements affecting device authentication under the group FR1 “identification and authentication control” (see Figure 2 below). The requirements being most relevant for device authentication are summarized here:

- SR1.1 Human user identification and authentication: The capability to identify and authenticate all human users is required. While for SL1, a group based authentication is possible, a unique identification of human users is required for SL2. A multi-factor authentication is required for human users when accessing from an untrusted network in SL3, while SL4 requires support for a multifactor authentication of human users for all networks.

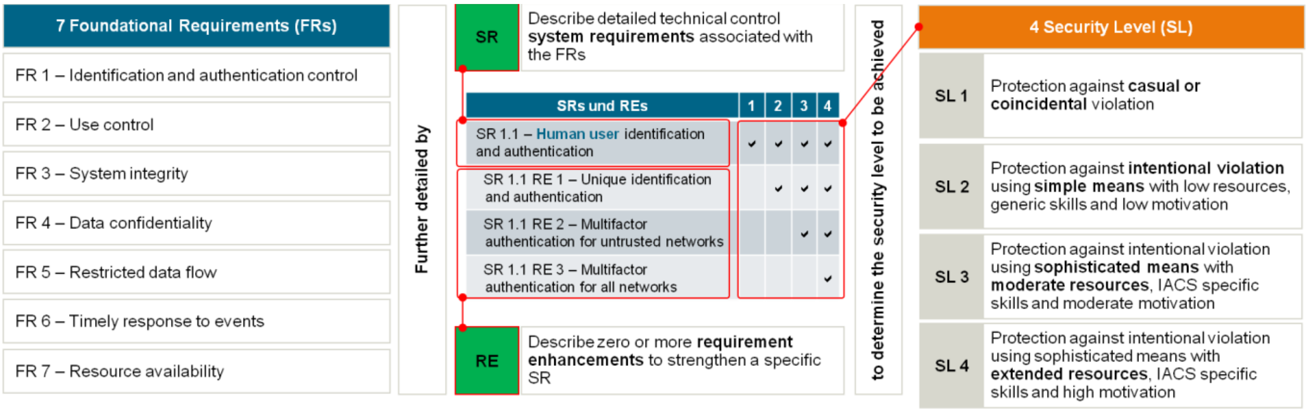
7 Foundational Requirements (FRs)
FR 1 – Identification and authentication control
FR 2 – Use control
FR 3 – System integrity
FR 4 – Data confidentiality
FR 5 – Restricted data flow
FR 6 – Timely response to events
FR 7 – Resource availability
Further detailed by
SR
Describe detailed technical control system requirements associated with the FRs
SRs und REs
1
2
3
4
SR 1.1 – Human user identification and authentication ✓ ✓ ✓ ✓
SR 1.1 RE 1 – Unique identification and authentication ✓ ✓ ✓
SR 1.1 RE 2 – Multifactor authentication for untrusted networks ✓ ✓
SR 1.1 RE 3 – Multifactor authentication for all networks ✓
RE
Describe zero or more requirement enhancements to strengthen a specific SR
to determine the security level to be achieved
4 Security Level (SL)
SL 1
Protection against casual or coincidental violation
SL 2
Protection against intentional violation using simple means with low resources, generic skills and low motivation
SL 3
Protection against intentional violation using sophisticated means with moderate resources, IACS specific skills and moderate motivation
SL 4
Protection against intentional violation using sophisticated means with extended resources, IACS specific skills and high motivation

Figure 2. The interrelation of foundational requirements, security requirements, and security levels in IEC 62443

- SR1.2 Software process and device identification and authentication: All devices, and software processes shall be possible to be identified, and authenticated. This requirement is relevant from security level SL2, and higher. While in SL2, group- or role-based identification, and authentication is permitted, for SL3, and SL4, a unique identification, and authentication of devices is required.
- SR1.5 Authenticator management: Authenticators are credentials used to authenticate users, devices, or software processes. They have to be initialized, and refreshed. Initial authenticators shall be possible to be changed. The requirement is relevant for SL2, SL3, and SL4. For SL3, and SL4, a hardware mechanism is required to protect authenticators.
- SR 1.7 Strength of password-based authentication: The required password strength has to be configurable based on minimum length and variety of character types. For SL3, a password history and lifetime restrictions has to be supported for human user passwords. For SL4, the password lifetime has to be restricted for all users, including devices and software processes.
- SR1.8 Public key infrastructure (PKI) certificates: When a PKI is used, it shall be operated according to commonly accepted best practices, or public key certificates shall be obtained from an existing PKI. The requirement is relevant for SL2, SL3, and SL4.
- SR1.9 Strength of public key authentication: When digital certificates are used, the certificate, the certificate path, and the certificate revocation status have to be checked. In SL3, and SL4, private keys have to be protected using a hardware-based mechanism.
- SR1.10 Authenticator feedback: The feedback of authentication information during the authentication process shall be possible to obscure.
- SR 1.11 Unsuccessful login attempts: The number of consecutive, unsuccessful login attempts during a given time period shall be possible to be limited for all users, i.e., human users, devices, and for software processes.

The importance that is given to authentication to protect an industrial automation and control system can be seen from this list of security requirements. These requirements have to be fulfilled while respecting side-conditions on high availability, and keeping safety-critical control networks closed. These imply that a control system should continue to operate locally, independently from any backend systems, or backend connectivity. Local emergency actions, as well as essential control functions shall not be hampered with by security mechanisms.

III. Device Authentication Methods

Device authentication is required by security standards. For example, IEC 62443 part 3-3 [3] includes security requirements for authentication of all users, including devices and software processes. As for users, authentication of a device can be based on different authentication factors, similar to user authentication means [14]:

- Something the device knows: credential (device key, e.g., a secret key or a private key)
- Something the device has (integrated authentication IC, authentication dongle)
- Something the device is (logical properties, e.g., the device type, configuration data, firmware version; physical properties: physical unclonable function (PUF), radio fingerprint)

Besides these well-established authentication factors, more unconventional authentication factors can also be used:

- Something the device does (behavior, functionality, e.g., automation control protocol)
- Something the device knows about its environment (sensors)
- Something the device can (functional capability, actuators)
- The context of the device (neighbors, location, connected periphery)

Different usages in IoT systems apply device authentication:

- Identity Authentication toward a remote system (access control, communication security). May be a supervisory system, or a peer device.
- Network access security (IEEE 802.1X [9], mobile network access authentication [10]).
- Original device authentication
- Attestation of device integrity
- Attestation of device configuration

The remainder of this section provides an overview about device authentication means. The authentication would typically be performed by an authentication server that, after successful authentication, may allow access to further system specific data directly or issues a temporal token (e.g., SAML assertion [11], OAUTH token [12], short-term X.509 certificate [13]).

A. *Cryptographic Device Authentication*

The authentication of a device allows a reliable identification. For authentication, a challenge value is sent to the object to be authenticated. This object calculates a corresponding response value, which is returned to the requestor and verified. The response can be calculated using a cryptographic authentication mechanism, or by using a PUF [2].

For cryptographic authentication, different mechanisms may be used. Examples are keyed hash functions like HMAC-SHA256 or symmetric ciphers in cipher block chaining (CBC-MAC) mode, or symmetric ciphers in Galois counter mode (GMAC) up to digital signatures (e.g., RSA or ECDSA). For the symmetric ciphers, AES would be a suitable candidate. Common to keyed hashes or symmetric key based cryptographic authentication approaches is the existence of a specific secret or private key, which is only available to the object to be authenticated and the verifier. One resulting requirement from this fact is obviously the need for robust protection of the applied secret key. Also, asymmetric cryptography can be used for component authentication. A suitable procedure based on elliptic curves has been described in [30]. Also in this use case, the secret key has to be protected on the authenticating component.

The device is authenticated as only an original device can determine the correct response value corresponding to a given challenge. The verifier sends a random challenge to the component that determines and sends back the corresponding response. The verifier checks the response. Depending on the result, the component is accepted as genuine/authenticated or it is rejected.

Various approaches are available to realize a cryptographic device authentication:

- Software credential: Credentials are hidden in software, configuration information, or the system registry. Be aware that practices of storing cryptographic credentials in firmware or cleartext configurations are weak [17][18]. However, techniques for whitebox cryptography are available that hide keys in software [19].
- Central processing unit (CPU) and microcontroller integrated circuits (IC) with internal key store: Some modern CPUs and microcontrollers include battery-backed SRAM or non-volatile memory, e.g., security fuses, that can be used to store cryptographic keys on the IC [20]. Also, an internal hardware security module (HSM) or secure execution environment can be included (e.g., Infineon Aurix with integrated HSM [21], or ARM TrustZone [22]).
- Separate authentication ICs can be integrated (e.g., Atmel CryptoAuthentication ECC508A [23] , Infineon Optiga Trust E [24]).
- Crypto controller (e.g., Infineon SLE97 [25]).
- Trusted platform module (TPM 1.2 [26], TPM 2.0 [27], TPM automotive thin profile [40]).

B. *Device Authentication based on Intrinsic Device Properties*

Physical and logical properties of a device can be verified as part of a device authentication. For this purpose, information about the device properties can be provided in a cryptographically protected way. In particular, an attestation, a digitally signed information confirming properties of a device, can be created by a protected component of the device.

Properties of the device can be logical information (software version, device configuration, serial number of components of the device) or physical properties of the device that can be determined by sensors or a PUF [15].

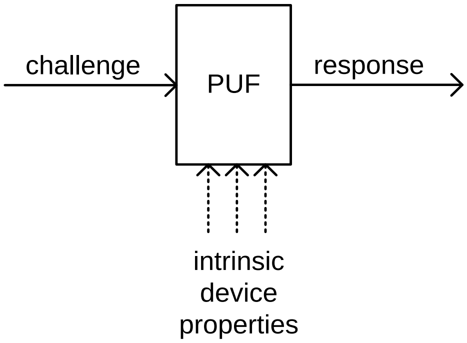

Figure 3. Challenge-Response-PUF [2]

Fig. 3 shows the basic concept of a PUF [2]. A PUF performs a computation to determine a response value depending on a given challenge value. Intrinsic device properties influence the PUF calculation so that the calculation of the response is different on different devices, but reproducible – with some bit errors – on the same device.

A PUF is used here for device authentication in a different way: It is by itself not a strong authentication. Instead, a cryptographically protected attestation can be used to attest physical properties of a device that are measured using a PUF. So, a PUF is not used directly for authentication, but indirectly as integrated device sensor to measure physical properties of the device. It can be considered as a “two-factor device authentication” where the PUF is used as second authentication factor.

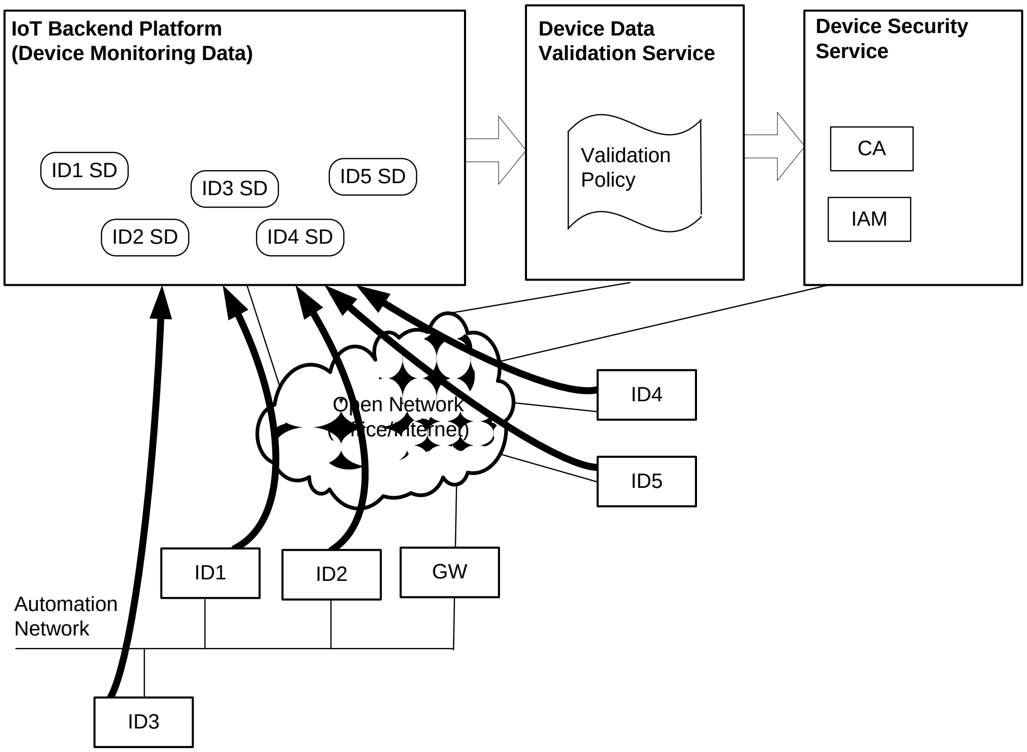

Figure 4. Validation of Device Monitoring Data

C. Authentication based on Device Context and Monitoring Information

Information about the context of a device can be used, e.g., the device location, or information about the environment of neighbor devices, the network reachability under a certain network address, or over a certain communication path.

The device context is determined and checked. The context information can be provided by the device itself, or the device's context information can be requested from a context server. One example from industrial environments is the system and device engineering, which basically provides information about the type and functionality of connected devices. Hence, it can be used to retrieve information about the devices deployment environment. The device location can be obtained using known localization technologies, e.g., global navigation satellite systems (GNSS) as GPS, GALILEO, BEIDOU, GLONASS, or localization using base stations (WLAN, cellular, broadcast) and beacons [28].

Furthermore, the device operation can be monitored: The behavior of the main, regular functionality of the device can be monitored and checked for plausibility.

Fig. 4 above shows an example for an IoT system with IoT devices (ID1, ID2, etc.) that communicate with an IoT backend platform. The devices provide current monitoring information about their status, measurements, etc. to the backend platform (e.g., for predictive maintenance). The backend platform maintains supervisory data (SD) data for the IoT devices (ID1 SD, ID2 SD, etc.) as "digital twin". Furthermore, context information about the environment of a device can be provided by the device itself using its sensors, or by neighboring devices.

The devices authenticate, e.g., using a device certificate, towards a device security service that maintains information about registered devices and their permissions. Furthermore, the device security service can issue and revoke device credentials (e.g., device certificate, authentication tokens).

In addition, a device data validation service can ensure that the device operation can be monitored, supporting also a continuous verification of the devices purpose. The validation service requests information about the IoT device supervisory data of supervised devices and checks it for validity using a configurable validation policy. Hence, the behavior of the main, regular functionality of the device can be monitored and checked for plausibility. Additionally, some arbitrary dummy functionality can be realized for monitoring purposes (e.g., predictable, pseudo-random virtual sensor measurement).

If a policy violation is detected, a corrective action is triggered: provide alarm message for display on a dash board (the alarm message can be injected in the device supervisory data set of the affected device maintained by the IoT backend platform). Furthermore, an alarm message can be sent to the IoT backend platform to terminate the communication session of the affected IoT device. Moreover, the device security service can be informed so that it can revoke the devices access permissions, or revoke the device authentication credential.

D. Authentication based on Device Capability

The authenticity of an automation device, e.g., for industrial automation and control systems or industrial IoT, can be verified by checking that a device can in fact perform a certain operation. The device is given an instruction to perform a certain test operation. It is checked that the device can perform a certain computation on provided test data: The device is given a set of input parameters (test data) and has to provide the correct result that is a valid result of the computation. The computational function could be a cryptographic puzzle involving a secret. The functionality can be realized by software/firmware on the control device, by a programmable hardware (FPGA), or by a periphery device (e.g., separate signal processor or IO device). Furthermore, it can be verified that a device can act on the expected physical environment (proofing that it has control on a certain effect in the physical world). The effect is observed by a separate sensor device. In an embodiment, the separate sensor device may provide an assertion.

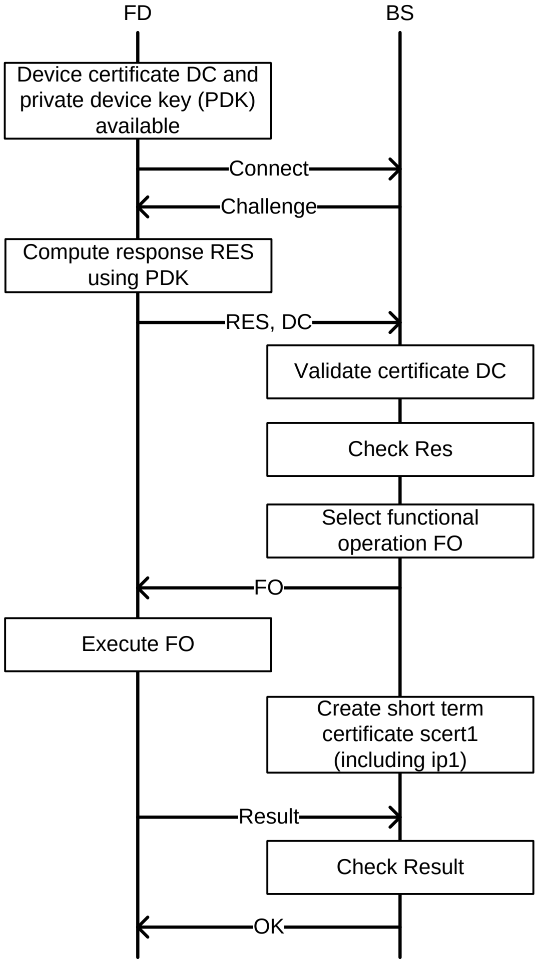
FD
BS
Device certificate DC and private device key (PDK) available
Connect
Challenge
Compute response RES using PDK
RES, DC
Validate certificate DC
Check Res
Select functional operation FO
FO
Execute FO
Create short term certificate scert1 (including ip1)
Result
Check Result
OK

Figure 5. Verification of Device Capability

Fig. 5 shows a possible message exchange. The functional capability check is performed over a cryptographically authenticated communication link (e.g., transport layer security (TLS) protocol [31]). A device passes the authentication if both its cryptographic authentication is valid and its functional operation (FO) is verified successfully. For a successful attack, where a fake device is to be accepted, it is not sufficient that the attacker has access to the used cryptographic key. In addition, the attacker has to realize the expected functionality of the real device.

An example for combining authentication and the property to control a specific environment can be given by the recently established letsencrypt [45] infrastructure. Here, a (web)server applies for an X.509 certificate to be used for authentication in the context of https connections made to the web server. The certificate will be issued once the server can prove that it controls the domain it is requesting a certificate for. The proof is provided by putting dedicated information onto a random address in the applying servers address space. If this information can be retrieved externally, the proof of control is provided.

IV. Combined Device Authentications

This section describes various advanced options for device authentication where multiple device authentications are combined.

A. Multi-Factor Device Authentication

A device can support multiple independent authentications. These authentication options may be performed iteratively.

In particular, an initial cryptographic device authentication can be used to setup an authenticated communication session with an authentication server. Additional checks can be performed to complete the device authentication, e.g., in the scope of a specific application.

B. Separate Re-authentication Connection

In communication security, a secure session is established by an authentication and key agreement protocol (e.g., IKEv2, TLS authentication and key agreement). The authentication is typically performed for each communication session.

It is proposed that a single device has to set-up multiple authenticated communication sessions. The device has to re-authenticate regularly towards a backend system respectively a separate authentication server using a first communication session. If this is not done, the second communication session is terminated or blocked by the backend system. This realizes a form of continuous device authentication where a device is continuously re-authenticated during a communication session, but without degrading the main communication link, for which delays and interruptions shall be avoided.

Figure 6 shows an example message exchange where a separate communication session is established for performing a regular re-authentication. A second communication session is setup for control communication that may have specific requirements on real-time behavior, interruptions, delay, and jitter. This second communication session is terminated if the re-authentication on the first session is not performed as expected.

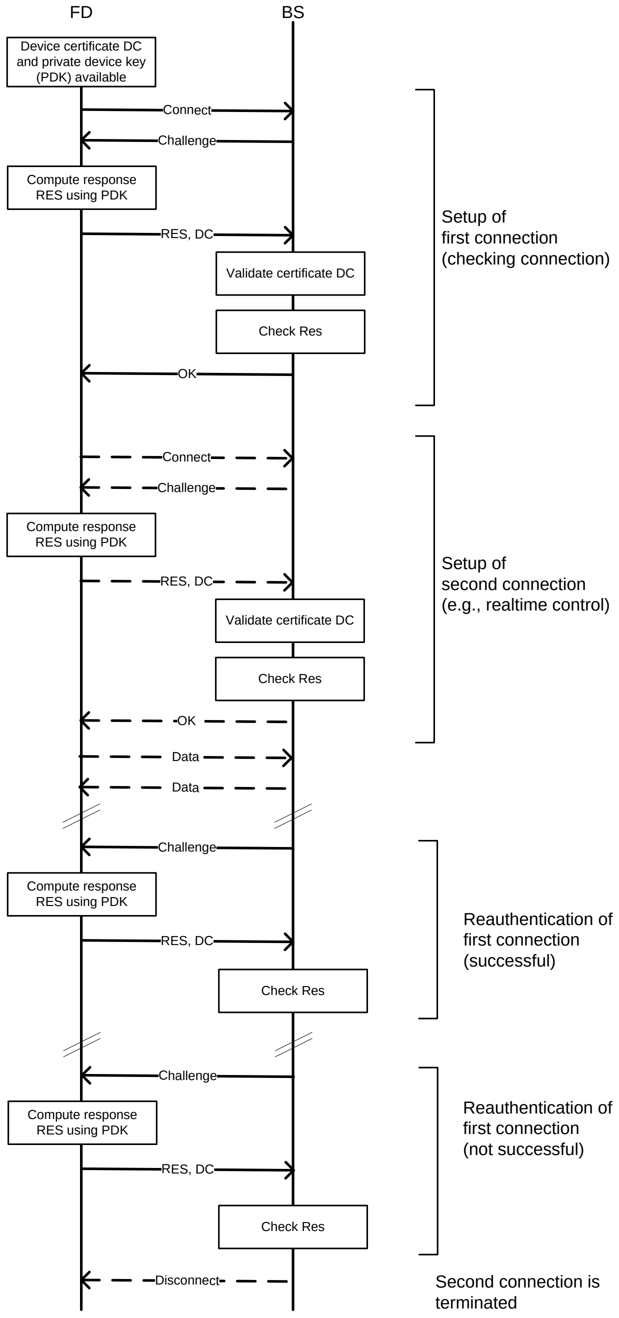
FD
BS
Device certificate DC and private device key (PDK) available
Connect
Challenge
Compute response RES using PDK
RES, DC
Validate certificate DC
Check Res
OK
Setup of first connection (checking connection)
Connect
Challenge
Compute response RES using PDK
RES, DC
Validate certificate DC
Check Res
OK
Setup of second connection (e.g., realtime control)
Data
Data
Challenge
Compute response RES using PDK
RES, DC
Check Res
Reauthentication of first connection (successful)
Challenge
Compute response RES using PDK
RES, DC
Check Res
Reauthentication of first connection (not successful)
Disconnect
Second connection is terminated

Figure 6. Continuous Device Authentication

The second communication session can be used for real-time / delay sensitive control traffic. The communication session will often be established for a long time (e.g., months). The re-authentication of the device can be performed independently using a second communication session without interfering with the first communication session (interruptions, delays during re-authentication). Note that the different communication sessions may terminate at different points in the backend systems. Hence, besides the multiple authentication sessions from the device, there needs to by a synchronization of the authentication sessions in the backend.

Also, the re-authentication of the first connection may be used to create a dynamic cryptographic binding with a further (separate) security session. This property can be used, to ensure that the entities involved in a separate security session know that there is a persistent first session with either the same entity or a different entity. This approach may be used for instance in publish/subscribe use cases to ensure that there is a persistent connection with the publish/subscribe server, while actually having an end-to-end communication session between the clients.

C. *System Authentication*

In industrial control systems and the Internet of Things, often a set of field devices will be used to realize a system. It is proposed to check the authentication of a set of devices (system authentication) that have to authenticate towards a backend system. A single device is accepted as authenticated only as long as a defined set of associated devices, forming the system, authenticates as well (with plausible context of the devices, e.g., network connectivity, location). The devices may have a different criticality assigned to enable a distinction between necessary and optional devices. The communication link of a device (as member of a group) is set to an active state (permission to send/receive data) only if all required devices of the group have authenticated successfully. Thereby, an attacker cannot perform a successful attack by setting up only a single fake device. A single device is accepted as authenticated only as long as a defined set of associated devices authenticates as well (with plausible context, e.g., network connectivity, location).

Fig. 7 shows an example where all three field devices (FD1, FD2, FD3) forming a group of devices, i.e., a system of interrelated field devices, have to authenticate against the backend system (BS). Only when all three devices have been authenticated, the exchange of data transfer with these devices is enabled.

D. *Device-internal Authentication Verification*

Device internal authentication may be directly integrated in different variants, like on a microcontroller, a safety subsystem, a main board, peripherals, housing authentication, or extension cards.

Multiple authentications can also be performed internally within a device. Subsystems or components of a device – e.g., main board, housing, safety subsystem, and extension cards – are checked internally within the device before authentication is enabled toward external systems. An explicit internal authentication using challenge response can be performed. The message flow would look almost identical for the one shown in Fig. 7, with the exception that device-internal components are authenticated instead of field devices of a device group.

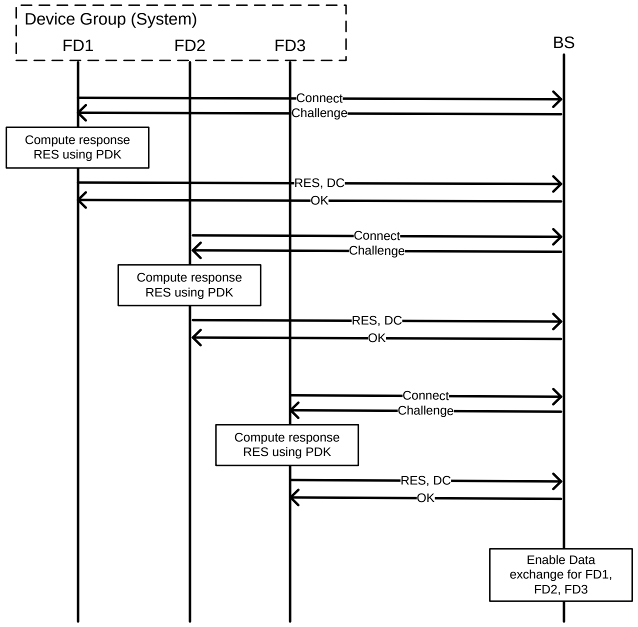
Device Group (System)
FD1
FD2
FD3
BS
Connect
Challenge
Compute response RES using PDK
RES, DC
OK
Connect
Challenge
Compute response RES using PDK
RES, DC
OK
Connect
Challenge
Compute response RES using PDK
RES, DC
OK
Enable Data exchange for FD1, FD2, FD3

Figure 7. System Authentication

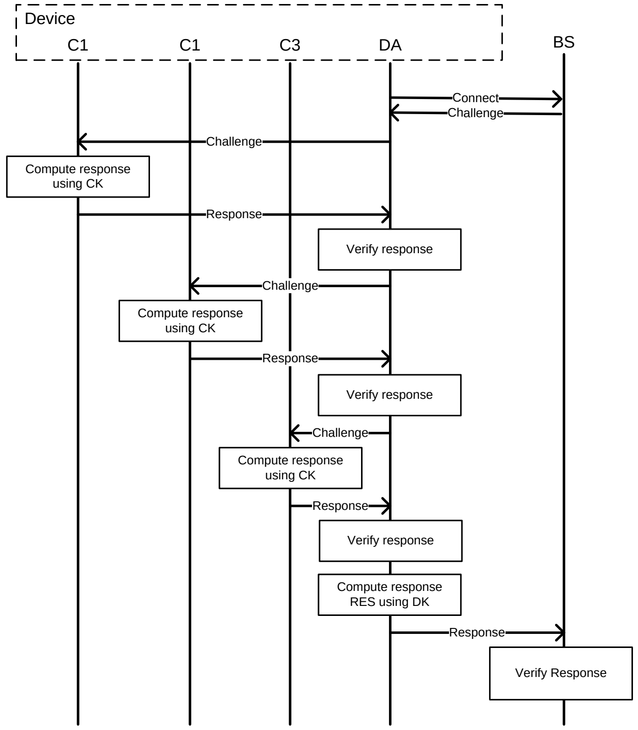

Figure 8. Device-internal Authentication

Fig. 8 shows an example where a device is authenticated by a backend system (BS). The device authenticates extranally using its device authentication functionality (DA). Before DA computes the response using the device key DK and providing the response to BS, it authenticates the device-internal components C1, C2, C3. Each component is authenticated using a device-internal challenge-response authentication.

Alternatively, a cryptographic secret sharing scheme can be used where a cryptographic operation can be performed only when all the required shares, i.e., partial computations that are performed independently, are available. For a device authentication, typically a public/private key pair is used. The public key is contained in an X.509 certificate and is associated to the devices by containing information about the device identity (e.g., serial number, MAC address). The private key – the secret – is supposed to never leave the device. Multiple parts of the device can be involved to access the private key needed to perform certain cryptographic operations (e.g., a digital signature, device authentication).

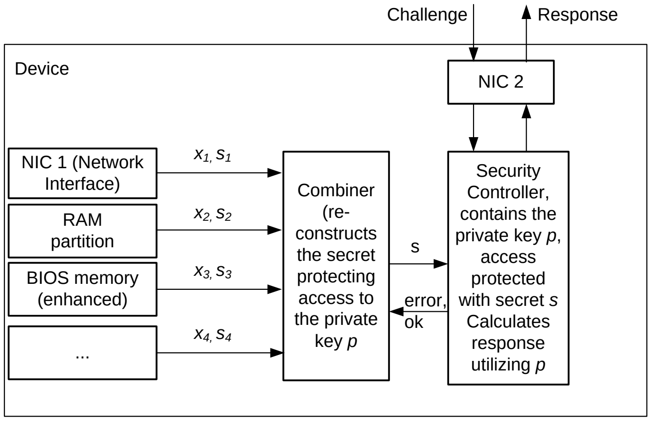
Challenge
Response
Device
NIC 2
NIC 1 (Network Interface)
RAM partition
BIOS memory (enhanced)
...
x1, s1
x2, s2
x3, s3
x4, s4
Combiner (re-constructs the secret protecting access to the private key p
s
error, ok
Security Controller, contains the private key p, access protected with secret s Calculates response utilizing p

Figure 9. Device-internal Authentication based on Secret Sharing

Fig. 9 shows an example where a device uses a secret-sharing scheme internally. The device key used for device authentication is determined during runtime by combining the different shares. This can be achieved by distributing the secret key (private key) in shares between device components. One approach for sharing may utilize Shamir's secret key sharing:

$$f(x) = s + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1} \mod p$$

with

- s as the secret key (here, the private key),
- a_i randomly chosen values,
- x_i may be public values.

Following Shamir's Scheme, the polynomial $f(x)$ is constructed in a way that the order of the polynomial is t-1. Now n nodes can be calculated with $n \geq t$ and based on that the initial secret can be split into n parts x_i, $s_i = f(x_i)$, (with $x_i \neq 0$), which in turn can be distributed to n different components of the device. To reconstruct the polynomial and thus the secret, the Lagrange interpolation is used.

Determining the value for the case $x=0$ leads to the constant part of the polynomial, which constitutes the secret.

Note that for the reconstruction, not all *n* parts are necessary, *t* parts are sufficient. This leads to the possibility to determine, which parts of the system need to be available to enable usage of the private key, e.g., in a challenge response authentication.

In contrast of sharing the private key directly, a secret, typically used to protect access to the private key, may be shared instead. This may be beneficial, if the private key is stored on dedicated security hardware.

In a variant the device may use the X.509 certificate to authenticate to other peers without making them aware of the internal dependencies to access the private key. In a further variant, the dependency is made public through an extension of the certificate. The extension may contain abstract information, e.g., threshold of device components necessary to access the private key or specific by listing the components necessary to access the private key.

Thereby, the device may use an X.509 certificate to authenticate towards a peer or the infrastructure. As the access to the private key is bound to the existence of a certain threshold of original components, however, the authenticating site is able to authenticate the device, and additionally gets information about the system integrity.

V. Evaluation

The security of a cyber system can be evaluated in practice in various approaches and stages of the system's lifecycle:

- Threat and risk analysis (TRA) of cyber system
- Checks during operation to determine key performance indicators (e.g., check for compliance of device configurations).
- Security testing (penetration testing)
- Security incident and event management (SIEM)

During the design phase of a cyber system, the security demand is determined, and the appropriateness of a security design is validated using a threat and risk analysis. Assets to be protected and possible threats are identified, and the risk is evaluated in a qualitative way depending on probability and impact of threats. The effectiveness of the proposed enhanced device authentication means can be reflected in a system TRA. The proposed enhancements to simple cryptographic device authentication can lead to a reduction of the probability and/or the impact of a threat, so that the overall risk for successful attacks is reduced.

Two exemplary threats affecting a device are given (using for this example a simple qualitative assessment metric of low/medium/high):

- An attacker obtains device authentication credential by attacking the authentication protocol (probability: medium, impact: high; risk: high).
- An attacker succeeds in exploiting an implementation vulnerability of a device to get root access to the device and manipulate the device functionality (probability: high, impact: high; risk: high).

With selected additional protection measures, the risk can be reduced to an acceptable level: A device authentication credential cannot be used by an attacker for a successful attack as the device credential alone does not allow for a successful device authentication. With functional verification of device capability, a manipulated device can be detected. For a successful attack, the attacker would have to ensure continuously the correct operation of the device as verified by the capability check, which increases the effort for the attacker. While in real-world attack models, it is never possible to prevent all attacks, the presented countermeasures help to increase the required effort for a successful, undetected attack.

The obtained information can be used also by SIEM tools, and to perform forensic analysis. Big data analytics using artificial intelligence can analyze the data to detect suspicious behavior of devices.

VI. Related Work

Authentication within the Internet of Things is an active area of research and development. Gupta described multi-factor authentication of users towards IoT devices [35]. The Cloud Security Alliance published recommendations on identity and access management within the IoT [36]. Ajit and Sunil describe challenged to IoT security and solution options. Authentication systems for IoT where analyzed by Borgohain, Borgohain, Kumar, and Sanyal [38].

Al Ibrahim and Nair have combined multiple PUF elements into a combined system PUF [39].

An "automotive thin profile" of the Trusted Platform Module TPM 2.0 has been specified [40]. A vehicle is composed of multiple control units that are equipped with TPMs. A rich TPM manages a set of thin TPMs, so that the vehicle can be represented by a vehicle TPM to the external world.

For electric vehicle charging, a vehicle authentication scheme has been described by Chan and Zhou [41] that involves two authentication challenges, sent over different communication links (wireless link, charging cable) to the electric vehicle.

Host-based intrusion detection systems (HIDS) as SAMHAIN [42] and OSSEC [43] analyze the integrity of hosts and report the results to a backend security monitoring system.

Continuous user authentication, i.e., the checking during a session whether the user is still the same as the authenticated one, has been described by [32] and [33].

Haider at al. describe a multi-factor memory authentication that combines hardware-based memory integrity verification and software-based bounds checking [44].

VII. Conclusion

Robust and practical device authentication is an essential security feature for cyber physical systems and the Internet of Things to verify the identity of devices that communicate over open networks. The security design principle of "defense in depth" basically means that multiple layers of defenses are designed. This design principle can not only be applied at the system level, but also at the level of a single security mechanism.

This paper proposed means for advanced device authentication to increase the attack robustness of device authentication. A well-known cryptographic device authentication can be extended with additional validations to check the device identity. The paper described how concepts known from advanced human user authentication like multi-factor authentication and continuous authentication can be applied to device authentication. They can be used to improve the security level for device authentication in an industrial control system and in an industrial IoT environment. Also, the concept of authentication of a single entity, as a single human user, a single device, or a single process, is expanded to the authentication of a system that comprises a multitude of entities.

The consequent next step is to setup pilots to integrate a selection of enhanced device authentication means as proof of concept, allowing to verify the concepts as such in a realistic application environment, and to analyze the advantages and the applicability of these advanced authentication technologies in a real-world setting. With the upcoming cloud-based platforms for industrial Internet of Things supporting cloud-based apps executable in the IoT backend [46], it is possible to realize advanced device authentication technologies flexibly by setting-up specific cloud apps that implement advanced device authentication functionality. So, a cloud-based industrial IoT backend, which can also be called the industrial IoT operating system, provides the technical basis for a quick introduction of new research-oriented technology developments into productive use.

REFERENCES

[1] R. Falk and S. Fries, “Advanced Device Authentication: Bringing Multi-Factor Authentication and Continuous Authentication to the Internet of Things,” The First International Conference on Advances in Cyber-Technologies and Cyber-Systems (CYBER 2016), pp. 69-74, 9 -13 October 2016, Venice, Italy, Thinkmind, available from: http://www.thinkmind.org/index.php?view=article&articleid=cyber_2016_4_20_80029, last access: May 2017

[2] R. Falk and S. Fries, “New Directions in Applying Physical Unclonable Functions,” The Ninth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), pp. 31-36, 23-28 August 2015, Venice, Italy, Thinkmind, available from: https://www.thinkmind.org/index.php?view=article&articleid=securware_2015_2_20_30028, last access: May 2017

[3] IEC 62443, “Industrial Automation and Control System Security,” (formerly ISA99), available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx , last access: May 2017

[4] ISO/IEC 62351-8, “Role-based access control for power system management,” June 2011

[5] NIST IR 7628, “Guidelines for Smart Grid Cyber Security,” Sep. 2014, available online http://dx.doi.org/10.6028/NIST.IR.7628r1, last access: May 2017

[6] BDEW, “Requirements for Secure Control and Telecommunication Systems,” whitepaper, February 2015, available online: https://www.bdew.de/internet.nsf/id/232E01B4E0C52139C1257A5D00429968/$file/OE-BDEW-Whitepaper_Secure_Systems%20V1.1%202015.pdf, last access: May 2017

[7] ISO, “Road vehicles - Vehicle to grid communication interface - Part 3: Physical and data link layer requirements,” ISO 15118-3, 2015, available online: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=59675, last access: May 2017

[8] R. Falk and S. Fries, “Using Managed Certificate Whitelisting as a Basis for Internet of Things Security in Industrial Automation Applications,” International Journal on Advances in Security, vol 8, nr. 1-2, pp. 89-98, 2015, Available online: http://www.iariajournals.org/security/ , last access: May 2017

[9] “IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control,” IEEE standard, 802.1X-2010, available from https://standards.ieee.org/findstds/standard/802.1X-2010.html , last access: May 2017

[10] G. Horn and P. Schneider, “Towards 5G Security,” 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-15), Helsinki, Finland, 20-22 August, 2015, available from http://networks.nokia.com/sites/default/files/document/conference_paper__towards_5g_security_.pdf , last access: May 2017

[11] Wikipedia, “Security Assertion Markup Language,” available from https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language , last access: May 2017

[12] J. Richer, “User Authentication with OAuth 2.0,” available from http://oauth.net/articles/authentication/ , last access: May 2017

[13] E. Gerck, “Overview of Certification Systems: X.509, CA, PGP and SKIP,” MCG, 1998, available from http://mcwg.org/mcg-mirror/certover.pdf , last access: May 2017

[14] L. O'Gorman, “Comparing passwords, tokens, and biometrics for user authentication,” Proceedings of the IEEE, vol. 91, issue 12, pp. 2021 – 2040, 2003

[15] C. Herder, Y. Meng-Day, F. Koushanfar, and S. Devadas, “Physical Unclonable Functions and Applications: A Tutorial,” Proceedings of the IEEE, vol. 102, nr. 8, pp. 1126-1141, August 2014, available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6823677, last access: May 2017

[16] R. Falk and S. Fries, “Advances in Protecting Remote Component Authentication,” International Journal on Advances in Security, vol 5, nr. 1-2, pp. 28-35, 2012, Available online: http://www.iariajournals.org/security/ , last access: May 2017

[17] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, “A Large-Scale Analysis of the Security of Embedded Firmwares,” 23rd USENIX Security Symposium, August 20–22, 2014, San Diego, CA, available from https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-costin.pdf , last access: May 2017

[18] R. Santamarta, “Identify Backdoors in Firmware By Using Automatic String Analysis,” 2013, available from http://blog.ioactive.com/2013/05/identify-back-doors-in-firmware-by.html , last access: May 2017

[19] J. A. Muir, “A Tutorial on White-box AES,” Cryptology ePrint Archive, Report 2013/104, available from https://eprint.iacr.org/2013/104.pdf , last access: January 2017

[20] M. Balakrishnan, “Freescale Trust Computing and Security in the Smart Grid,” Freescale white paper, document number: TRCMPSCSMRTGRDWP REV 1, 2013, available from

http://cache.nxp.com/files/32bit/doc/white_paper/TRCMPSCSMRTGRDWP.pdf , last access: May 2017

[21] Infineon, “Highly integrated and performance optimized 32-bit microcontrollers for automotive and industrial applications,” 2016, available from http://www.infineon.com/dgdl/TriCore_Family_BR-2016_web.pdf?fileId=5546d46152e4636f0152e59a1581001d, last access: May 2017

[22] ARM: “Building a Secure System using TrustZone Technology,” ARM whitepaper PRD29-GENC-009492C, 2005 - 2009, available from http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf , last access May 2017

[23] Atmel, “ATECC508 Atmel CryptoAuthentication Device,” summary datasheet, 2015, available from http://www.atmel.com/images/atmel-8923s-cryptoauth-atecc508a-datasheet-summary.pdf , last access: May 2017

[24] Infineon, “Optiga Trust E SLS32AIA,” product brief, 2016 available from http://www.infineon.com/dgdl/Infineon-OPTIGA%E2%84%A2+Trust+E+SLS+32AIA-PB-v02_16-EN.pdf?fileId=5546d4624e765da5014eaabac63f5a38, last access: May 2017

[25] Infineon, “SOLID FLASH™ SLE 97 Family,” product brief, 2012, available from http://www.infineon.com/dgdl/Infineon-SOLID_FLASH_SLE_97_Family_32-bit_High_Performance-PB-v08_12-EN.pdf?fileId=db3a30433917ea3301392ec288fc4ff0, last access: May 2017

[26] Trusted Computing Group: “TPM Main Specification,” Version 1.2, , available from http://www.trustedcomputinggroup.org/resources/tpm_main_specification, last access: May 2017

[27] Trusted Computing Group, “Trusted Platform Module Library Specification, Family 2.0,” 2014, available from http://www.trustedcomputinggroup.org/resources/tpm_library_specification, last access: May 2017

[28] K. Pahlavan et al., “Taking Positioning Indoors, Wi-Fi Localization and GNSS,” InsideGNSS, pp. 40-47, May 2010, available from http://www.insidegnss.com/auto/may10-Pahlavan.pdf , last access: May 2017

[29] B. Parno, “ Bootstrapping Trust in a Trusted Platform,” 3rd USENIX Workshop on Hot Topics in Security, July 2008, available from http://www.usenix.org/event/hotsec08/tech/full_papers/parno/parno_html/, last access: May 2017

[30] M. Braun, E. Hess, and B. Meyer, “Using Elliptic Curves on RFID Tags,” International Journal of Computer Science and Network Security, vol. 2, pp. 1-9, February 2008

[31] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” RFC 5246, Aug. 2008, available from http://tools.ietf.org/html/rfc5246 , last access: May 2017

[32] H. Xu, Y. Zhou, and M. R. Lyu, “Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones,” Symposium on Usable Privacy and Security (SOUPS) 2014, July 9–11, 2014, Menlo Park, CA, available from: https://www.usenix.org/system/files/conference/soups2014/soups14-paper-xu.pdf , last access: May 2017

[33] K. Niinuma and A. K. Jain, “Continuous User Authentication Using Temporal Information,” available from http://biometrics.cse.msu.edu/Publications/Face/NiinumaJain_ContinuousAuth_SPIE10.pdf , last access: April 2016

[34] N. Costigan and I. Deutschmann, “DARPA’s Active Authentication program,” RSA Conference Asia Pacific 2013 available from https://www.rsaconference.com/writable/presentations/file_upload/sec-t05_final.pdf , last access: May 2017

[35] U. Gupta, “Application of Multi factor authentication in Internet of Things domain: multi-factor authentication of users towards IoT devices,” Cornell university arXiv:1506.03753, 2015, available from: http://arxiv.org/ftp/arxiv/papers/1506/1506.03753.pdf , last access: May 2017

[36] A. Mordeno and B. Russel, “Identity and Access Management for the Internet of Things - Summary Guidance,” Cloud Security Alliance, 2015, available from: https://downloads.cloudsecurityalliance.org/assets/research/internet-of-things/identity-and-access-management-for-the-iot.pdf , last access: May 2017

[37] J. Ajit and M.C. Suni, “Security considerations for Internet of Things,” L&T Technology Services, 2014, http://www.lnttechservices.com/media/30090/whitepaper_security-considerations-for-internet-of-things.pdf, last access: May 2017

[38] T. Borgohain, A. Borgohain, U. Kumar, and S. Sanyal, “Authentication Systems in Internet of Things,” Int. J. Advanced Networking and Applications, vol. 6, issue 4, pp. 2422-2426, 2015, available from http://www.ijana.in/papers/V6I4-11.pdf , last access: May 2017

[39] O. Al Ibrahim and S. Nair, “Cyber-Physical Security Using System-Level PUFs,” 7th International Wireless Communications and Mobile Computing Conference (IWCMC), 2011, available from http://lyle.smu.edu/~nair/ftp/research_papers_nair/CyPhy11.pdf , last access: May 2017

[40] Trusted Computing Group, “TCG TPM 2.0 Automotive Thin Profile,” level 00, version 1.0, 2015, available from http://www.trustedcomputinggroup.org/resources/tcg_tpm_20_library_profile_for_automotivethin, last access: May 2017

[41] A. C-F. Chan and J. Zhou, “Cyber-Physical Device Authentication for Smart Grid Electric Vehicle Ecosystem,” IEEE Journal on Selected Areas in Communications, vol. 32, issue 7, pp. 1509 – 1517, 2014

[42] R. Wichmann, “The Samhain HIDS,” fact sheet, 2011, available from http://la-samhna.de/samhain/samhain_leaf.pdf, last access: January 2017

[43] OSSEC, “Open Source HIDS SECurity,” web site, 2010 - 2015, available from http://ossec.github.io/, last access: May 2017

[44] S. K. Haider et al., “M-MAP: Multi-Factor Memory Authentication for Secure Embedded Processors,” 33rd IEEE International Conference on Computer Design (ICCD), Oct. 2015, IEEE, available from: https://eprint.iacr.org/2015/831, last access: May 2017

[45] Letsencrypt, letsencrypt.org, last access: January 2017

[46] Siemens, “MindSphere – Siemens Cloud for Industry,” The Magazine, Siemens, 2015, available online: https://www.siemens.com/customer-magazine/en/home/industry/digitalization-in-machine-building/mindsphere-siemens-cloud-for-industry.html, last access: May 2017