Lightweight Secure Bootstrap and Message Attestation in the Internet of Things

Clémentine Gritti, Refik Molva, Melek Önen
EURECOM, Sophia Antipolis, France
[clementine.gritti,refik.molva,melek.onen]@eurecom.fr

ABSTRACT

Internet of Things (IoT) offers new opportunities for business, technology and science but it also raises new challenges in terms of security and privacy, mainly because of the inherent characteristics of this environment: IoT devices come from a variety of manufacturers and operators and these devices suffer from constrained resources in terms of computation, communication and storage.

In this paper, we address the problem of trust establishment for IoT and propose a security solution that consists of a secure bootstrap mechanism for device identification as well as a message attestation mechanism for aggregate response validation. To achieve both security requirements, we approach the problem in a confined environment, named SubNets of Things (SNoT), where various devices depend on it. In this context, devices are uniquely and securely identified thanks to their environment and their role within it. Additionally, the underlying message authentication technique features signature aggregation and hence, generates one compact response on behalf of all devices in the subnet.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; • **Networks → Mobile and wireless security**;

KEYWORDS

Internet of Things, Secure Bootstrap, Message Attestation, Identity-Based Cryptography, Multisignature

ACM Reference Format:
Clémentine Gritti, Refik Molva, Melek Önen. 2018. Lightweight Secure Bootstrap and Message Attestation in the Internet of Things. In *Proceedings of ACM SAC Conference, Pau, France, April 9-13, 2018 (SAC'18),* 8 pages.
https://doi.org/10.1145/3167132.3167218

1 INTRODUCTION

The Internet of Things (IoT) has been a major phase in the evolution of computing and communication technologies in the last decade. Thanks to the seamless integration of a variety of devices with the physical environment, IoT offers new opportunities for business, technology and science but it also raises new challenges in terms of security and privacy. These challenges often are about classical requirements such as authentication, data security, malware detection and denial of service prevention. The novelty of these problems in the IoT setting is due to some inherent characteristics of the new environment:

- *Diversity of sources:* The devices involved in IoT originate from a huge variety of vendors, as opposed to previous evolutionary phases like the early Internet, the mobile communications, the cloud computing, etc. Similarly, the services deployed on these devices are operated by a large number of parties. The variety of both vendors and operators obviously does not increase the level of trust and integrity guarantees that can be expected.
- *Computational constraints:* Although IoT covers a wide range of devices, most IoT scenarios involve some devices with very low storage and processing capacity. The design of security and privacy solutions for IoT is thus severely impacted by the computational constraints of these devices.
- *Lack of governance:* The difficulty of establishing global security infrastructure that already affects the current Internet is increased in the case of IoT due to the sheer diversity of parties involved in the provision and operation of services.

In this paper, we focus on the basic problem of trust establishment in IoT by taking into account the aforementioned challenges. The first requirement of a trusted operation in IoT is the proper identification of the origin of messages transmitted by the devices, albeit a basic security requirement, such identification is not always possible in IoT due to the dynamic and often self-organizing nature or applications. In typical IoT settings, the seamless integration of devices with the environment does not allow for classical registration and setup procedures that would help to configure authentication mechanisms, such as credentials, certificates, etc. In less dynamic scenarios where setup procedures for security can be afforded, one still has to face additional difficulties due to the diversity of sources and lack of governance. Hence, the straightforward authentication based on Public Key Infrastructures (PKIs) is not applicable in most IoT settings.

As a first step towards trust establishment in IoT, we propose a dynamic security scheme that consists of:

- A secure bootstrap mechanism that allows for device identification without complex setup operations;
- A message attestation mechanism that allows for the aggregate authentication of messages transmitted by a set of devices.

This solution mainly addresses the data origin authentication problem as part of local communications in confined environments such as smart home, smart vehicle, smart wearables and body networks. Further extensions of the solution can easily be envisioned in order

SAC'18, April 9-13, 2018, Pau, France

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167218

to address the problem of secure end-to-end communications across IoT.

The proposed solution combines the use of identity-based cryptography to achieve a certificate-less secure bootstrap and a multisignature primitive to obtain a verifiable aggregate response from multiple devices within the confined environment.

In the next section, we carefully state the problems and challenges that bring device identification and message attestation in IoT. In Section 3, we present our idea to overcome previously exposed concerns. In Section 4, we define our scheme for secure bootstrap and aggregate response authentication in IoT. We also give a construction of such scheme that is then proven secure in the random oracle model. In Section 6, we give a concrete application of our scheme in the context of smart transportation.

2 PROBLEM STATEMENT

One of the fundamental security issues raised by the IoT is the problem of identity management. Device identification is a well-known and widely-studied problem. However, existing solutions are unfortunately not adapted to the IoT context as this raises some new challenges for this problem. A first challenge comes from the observable increase in the number of devices and in their diversity. The variety of devices' origins due to the presence of numerous manufacturers and operators on the market makes the identity assignment task difficult as each device should be uniquely identified. Unlike cellular telephone systems where there are a few well established manufacturers that are thus somehow trustworthy, the IoT environment involves devices coming from a large number of different operators and various manufacturers. Moreover, IoT might come from unknown origins and start transmitting without initial registration or clearance phase.

On the other hand, Public Key Infrastructure (PKI) has been developed for certificate-based identity authentication. However, some cases (not necessarily in the IoT environment) have shown that such frameworks are not always manageable and even implementable. For instance, PKI for the Secure Border Gateway Protocol (S-BGP)[6, 17] met practical limitations. Since BGP counts on routers from many organizations believing and passing along information they receive, multiple attacks threaten it. Security solutions for BGP require PKI; however, the latter has a noticeable impact on the performance of BGP [19]. In the IoT context, organizational obstacles could also be encountered when designing a global PKI where certification authorities would not be easily recognized and accepted.

Additionally, PKI raises new technical problems for IoT. Devices may be limited in terms of computation, communication and storage resources, and cannot easily implement this technology. Indeed, the size of PKI certificates can be too large for such devices. Another issue comes from neatly naming identities when creating certificates. For instance, traditional X.509 PKIs look too complex in defining appropriate distinguished names for numerous parties. Device identities become meaningless in global IoT.

To summarize, while device identification and authentication are considered as an old problem, classical solutions unfortunately fall short in the context of IoT. We propose to consider this problem in a confined environment defined as SubNet of Things (SNoT) whereby

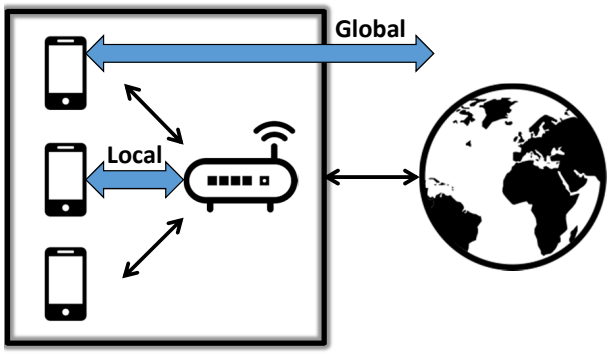

Figure 1: Secure bootstrap and verifiable data management in SNoT.

devices of each subnet converge to a central node connected to the Internet. For instance, one SNoT can represent a smart home gathering several automated home electronic devices sending various messages to the controller of this home.

Figure 1 illustrates the concept of secure communication in IoT being first approached locally in SNoT and further extended to the global space. Local communication alludes to data exchanges between some devices and one central node. These parties form a kind of confined environment, thus creating a local subnet in the global space.

The notion of local environment raises two security requirements:

- *Secure bootstrap:* The challenge is to assign locally unique identities to devices to ensure robust and reliable authentication.
- *Verifiable data management:* Once the confined environment is defined and devices receive the identities, the devices can exchange messages with the central node who acts as a controller of the subnet. In this context, we are interested in validating one global response by all devices in the subnet rather than individual authentication of each response. This aggregate response should be complete with respect to the functions of the devices in the subnet.

3 IDEA OF THE SOLUTION

As previously mentioned, we propose to address the problem of identity and message authentication in IoT at a local level. Instead of considering devices in the global space, we take the direction of SNoT that approaches the issues in a local context. We determine subnets as confined environments such as smart city, smart home, smart vehicle, human body network, etc.

To address the problem of secure bootstrap, we start by assigning each device one role within the SNoT. Thus, a device does not receive a global identification, but one depending on its subnet. Moreover, devices form an apparent set linked to their environment. Hence, messages generated by devices would be authenticated on behalf of this environment, and not of each device separately.

Indeed, from an external point of view, only the environment's identification is accessible but not the devices' ones.

An example to illustrate our idea occurs in the area of smart health care. Multiple medical machines monitor patients' vital activities to allow efficient and reliable health checking done by the hospital staff. Nevertheless, medical machines are provided by external manufacturers with no necessarily clear identification. When medical equipment is acquired by a hospital, it is distributed to the operation rooms, patient rooms, etc. Some machines are dedicated to one and only one room. Hence, the latter forms a proper confined environment, and the hospital is able to give an identity to these machines in function of their environment and role.

Another example takes place in the area of smart transportation. An Electronic Control Unit (ECU) is an embedded system that controls one or more electrical systems or subsystems in a transport vehicle, such as engine control module or transmission control module. An ECU participates to the good operation and safety of the vehicle by transmitting responses on its current condition. Since up to 100 ECUs can be embedded into smart vehicles, a batch response generated on behalf of all the ECUs might be more suitable than individual responses created by each ECU.

The idea to enable secure bootstrap in our solution is done by identifying devices regarding their credentials. More precisely, credentials of a device refer to the environment in which this device is installed as well as its role in this environment. By using such credentials, the identity of a device is thus locally unique.

To avoid the use of bulky certificates, secure bootstrap in the proposed local context can be efficiently achieved through Identity-Based Signatures (IBS) [18]. Indeed, IBS schemes by default set the device's identity as its public key, and the corresponding private key is generated according to this identity. Nevertheless, this does not resolve the second problem on which we focus, namely aggregate response validation. In more details, IBS schemes only allow for individual signature generation and would require the verification of each signature.

Alternatively, Multisignature (MS) schemes [3, 12, 14, 15] allow multiple devices to jointly sign a same message and obtain an aggregate signature on this message on behalf of all the devices. The final signature has the desirable property of compactness: its size should not be linear in the number of signing devices. Unfortunately, existing MS schemes are based on the use of PKIs and thus, devices' authentication would be based on digital certificates which make the communication inefficient and unpractical for IoT.

We therefore propose to take advantage of the two previously presented primitives to achieve security and efficiency. The solution is a 2-level hierarchical Identity-Based Multisignature scheme that would assign credentials to environments at the first level and the environments themselves would generate credentials for their devices (level 2). Additionally, as its name implies, the proposed mechanism allows for the aggregation of signatures originated from all devices in a same environment.

4 SECURE BOOTSTRAP AND AGGREGATE RESPONSE PROTOCOL

4.1 Overview of the Scheme

The protocol for secure bootstrap and aggregate response validation involves three distinct parties that are the following:

- *Environment:* An environment is a well-defined confined subnet. We can describe it as a universal entity in the whole Internet such that it creates a local frame in the global space. For instance, a subnet can represent a smart vehicle or a hospital room as previously illustrated in Section 3. The environment receives an identifying information that is used to create its private key.
- *Devices:* Devices join environments and are assigned identities regarding their environment and the role that they perform. Their signing keys are created regarding such credentials by the environment in which they are located. Their task is to generate a response one after the other using the responses of the other devices embedded into the same environment. The result calculated on behalf of all the devices is an aggregate response that is forwarded to a response collector for validation.
- *Response Collector:* The response collector is responsible for validating the aggregate response generated by the devices embedded into a same environment. The response is verified regarding a given message that should have been used by all the participating devices. The response collector is not a designated party; it could be anyone since only public elements are required to proceed the validation of the response.

The protocol for secure bootstrap and aggregate response validation splits into the following three phases:

Device Setup: During this phase, devices are securely assigned unique identities and receive relevant key material when joining their environment. On the other hand, the environment itself is also given some key material from an off-line Trusted Third Party (TTP) who is not involved in any other steps of the protocol.

Aggregate Response Generation: Devices are asked to sign a message sequentially. The resulting signature is seen as their batch response. It is calculated according to the response collector's request.

A first device receiving the request signs a message using its own signing key and forwards the resulting signature $\sigma_{i,1}$ to a second device. The latter uses its individual signing key and the previous signature $\sigma_{i,1}$ to generate its signature on the same message and forwards it to a third device. From there, the process is repeated: a device takes as inputs its own signing key, the previous signature $\sigma_{i,j-1}$ received from the previous device and the same message to create its own response.

The last created signature $\sigma_{i,l}$, where l is the number of devices in the environment, is set as the aggregate response on behalf of all the signing devices. The response is not a simple combination of individual responses from these devices. The device who generated $\sigma_{i,l}$ stores it until a response collector asks for it.

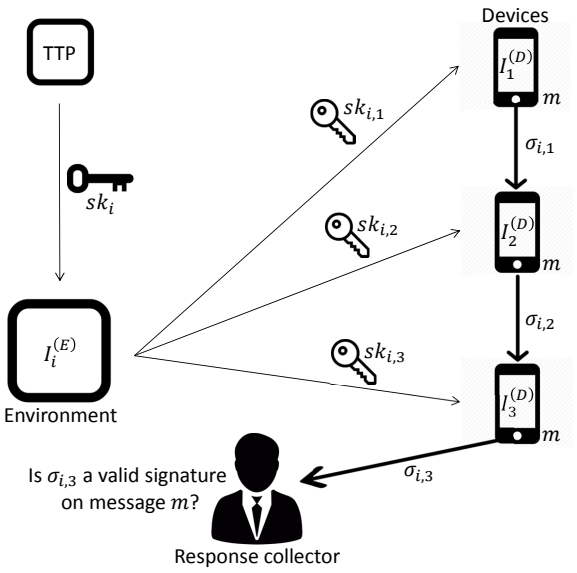

Figure 2: Scheme overview with an environment, three devices and a response collector participating.

Aggregate Response Validation: This part is executed by the response collector. It could be anyone since no secret elements are required to run the validation process. The response collector receives the aggregate response $\sigma_{i,l}$ and verifies its validity. If the result is positive, then the response is a valid signature on a certain message (the same one for all the devices).

Figure 2 illustrates the scheme where an environment, three devices and a response collector participate. More details on the components (keys and signatures) are given in the next section.

4.2 Definition of the Scheme

In the following, the identity $I_j^{(D)}$ refers to the role of the device and the identity $I_i^{(E)}$ refers to the identifying information of the environment. We assume that all the identities are unique. The 2-Identity-Based Multisignature scheme (2-IBMS) contains the following six algorithms:

Device Setup.

- $\text{Setup}(\lambda) \rightarrow (params, msk)$. This algorithm is run by the TTP. On input the security parameter λ, it outputs the public parameters $params$ and the master secret key msk. The public parameters $params$ are made public to the multiple devices (the signers), environment and response collector. The master secret key msk is kept secret by the TTP.
- $\text{KeyGen}_1(params, msk, I_i^{(E)}) \rightarrow sk_i$. This algorithm is run by the TTP. On inputs the public parameters $params$, the master secret key msk and the identity $I_i^{(E)}$ of an environment, it outputs the private key sk_i for this environment. The private key sk_i is forwarded to the environment.
- $\text{KeyGen}_2(params, sk_i, I_j^{(D)}) \rightarrow sk_{i,j}$. This algorithm is run by the environment with identity $I_i^{(E)}$ for a device with identity $I_j^{(D)}$. On inputs the public parameters $params$, the private key sk_i of the environment, and the identity $I_j^{(D)}$ of a device located in the environment, it outputs the signing key $sk_{i,j}$ for this device. We suppose that the environment runs this algorithm once the device is installed in it. The signing key $sk_{i,j}$ is given to the device.

Aggregate Response Generation.

- $\text{Sign}_j(params, sk_{i,j}, \sigma_{i,j-1}, m) \rightarrow \sigma_{i,j}$. This algorithm is run by the device with identity $I_j^{(D)}$ in the environment with identity $I_i^{(E)}$. On inputs the public parameters $params$, the signing key $sk_{i,j}$ of the device, the signature $\sigma_{i,j-1}$ generated by the previous signing device with identity $I_{j-1}^{(D)}$ in the same environment (for $j = 1$, $\sigma_{0,i} = \perp$ since there is no previous signature), and a message m, it outputs the signature $\sigma_{i,j}$. The signature $\sigma_{i,j}$ is forwarded to the next signing device with identity $I_{j+1}^{(D)}$ in the same environment.

Aggregate Response Validation.

- $\text{Verif}(params, I_i^{(E)}, \{I_j^{(D)}\}_{j\in[1,l]}, \sigma_{i,l}, m) \rightarrow$ {"Accept","Reject"}. This algorithm is run by the response collector that could be anyone since only the public parameters and identities (made public) are required. Let l be the number of devices in the same environment. We suppose that the response collector knows the identities of the devices and of their environment. On inputs the public parameters $params$, the identity $I_i^{(E)}$ of an environment, the set of identities $\{I_j^{(D)}\}_{j\in[1,l]}$ of the devices in this environment, a multisignature $\sigma_{i,l}$ corresponding to the signature generated by the l-th (last) device in this environment, and a message m, it outputs either "Accept" or "Reject".

Correctness. For all $(params, msk) \leftarrow \text{Setup}(\lambda)$, keys $sk_i \leftarrow \text{KeyGen}_1(params, msk, I_i^{(E)})$ and $sk_{i,j} \leftarrow \text{KeyGen}_2(params, sk_i, I_j^{(D)})$ for $j \in [1, l]$, and message m, if the $l-1$ signatures are generated as $\sigma_{i,j} \leftarrow \text{Sign}_j(params, sk_{i,j}, \sigma_{i,j-1}, m)$ for $j \in [1, l-1]$, and if the multisignature is generated as $\sigma_{i,l} \leftarrow \text{Sign}_l(params, sk_{i,l}, \sigma_{i,l-1}, m)$, then $\text{Verif}(params, I_i^{(E)}, \{I_j^{(D)}\}_{j\in[1,l]}, \sigma_{i,l}, m) \rightarrow$ "Accept".

4.3 Construction of the Scheme

4.3.1 Preliminaries.

Bilinear Maps.

Let G and G_T be two multiplicative cyclic groups of prime order p according to the security parameter λ. Let g be a generator of G. Let $e : G \times G \rightarrow G_T$ be a bilinear map with the following properties:

- *Bilinearity:* $\forall u, v \in G, \forall a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
- *Non-degeneracy:* $e(g, g) \neq 1_{G_T}$.

G is said to be a bilinear group if the group operation in $G \times G$ and the bilinear map e are both efficiently computable. We can easily see that e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

Computational Diffie-Hellman (CDH) Assumption.

The CDH problem is as follows. Let G be a group of prime order p according to the security parameter λ. Let $a, b \in_R Z_p$ and g be a generator of G. If an adversary $\mathcal{A}$ is given a CDH tuple (g, g^a, g^b), it remains hard to compute $g^{ab} \in G$. The CDH assumption holds if no Probabilistic Polynomial-Time (PPT) adversary $\mathcal{A}$ has non-negligible advantage in solving the CDH problem.

4.3.2 Construction. The 2-Identity-Based Multisignature scheme (2-IBMS) construction in the random oracle model is inspired from the IBMS one presented by Gentry and Ramzan [10]. However, in the latter, no hierarchical organization is supported. Our main challenge is thus to integrate a 2-level hierarchy composed by the TTP as the root, then the environments held by this TTP at level 1 and finally the devices embedded in the environments at level 2. Our 2-IBMS construction is as follows:

- $\mathsf{Setup}(\lambda) \rightarrow (params, msk)$. The TTP setups the protocol as follows. Given the security parameter λ, let G, G_T be two cyclic (multiplicative) groups of prime order p. Let g be a generator of G and $e : G \times G \rightarrow G_T$ be a bilinear map. The TTP picks at random $\alpha, \beta \in_R Z_p$ and computes $h_1 = g^\alpha$ and $h_2 = g^\beta$. Let $H_1, H_2, H_3 : \{0,1\}^* \rightarrow G$ be three cryptographic hash functions seen as random oracles. Finally, the TTP sets the public parameters as $params = (p, G, G_T, e, g, h_1, h_2, H_1, H_2, H_3)$ and the master secret key as $msk = (\alpha, \beta)$.
- $\mathsf{KeyGen}_1(params, msk, I_i^{(E)}) \rightarrow sk_i$. The TTP generates the private key sk_i for an environment with identity $I_i^{(E)} \in \{0,1\}^*$ as follows. It computes $g_i = H_1(I_i^{(E)})$ and $C_i^{(1)} = g_i^\alpha$, and sets $C_i^{(2)} = \beta$. It sets the private key of the environment with identity $I_i^{(E)}$ as $sk_i = (C_i^{(1)}, C_i^{(2)})$.
- $\mathsf{KeyGen}_2(params, sk_i, I_j^{(D)}) \rightarrow sk_{i,j}$. The environment generates the signing key $sk_{i,j}$ for a device with identity $I_j^{(D)} \in \{0,1\}^*$ located in it as follows. It parses its signing key sk_i as $(C_i^{(1)}, C_i^{(2)})$. It computes $g_j = H_2(I_j^{(D)})$ and $sk_{i,j} = C_i^{(1)} \cdot g_j^{C_i^{(2)}}$, and sets the signing key of the device with identity $I_j^{(D)}$ in the environment with identity $I_i^{(E)}$ as $sk_{i,j}$.
- $\mathsf{Sign}_j(params, sk_{i,j}, \sigma_{i,j-1}, m) \rightarrow \sigma_{i,j}$. A device with identity $I_j^{(D)}$ in the environment with identity $I_i^{(E)}$ signs the message $m \in \{0,1\}^*$ as follows. It picks at random $t_j \in_R Z_p$ and computes $g_m = H_3(m)$ and its individual elements $B_{i,j}^{(1)} = g_m^{t_j} \cdot sk_{i,j}$ and $B_{i,j}^{(2)} = g^{t_j}$. For $j = 1$, we recall that $\sigma_{i,0} = \perp$. Thus, the first device to sign sets its individual signature as $\sigma_{i,1} = (B_{i,1}^{(1)}, B_{i,1}^{(2)})$.

  Given the signature $\sigma_{i,j-1} = (S_{i,j-1}^{(1)}, S_{i,j-1}^{(2)})$ from the previous signing device in the same environment, for $j > 1$, the device generates the aggregate elements:

$$S_{i,j}^{(1)} = S_{i,j-1}^{(1)} \cdot B_{i,j}^{(1)} = \prod_{j'=1}^{j} B_{i,j'}^{(1)} = g_m^{\sum_{j'=1}^{j} t_{j'}} \cdot g_i^{j\alpha} \cdot \prod_{j'=1}^{j} g_{j'}^{\beta}$$

$$S_{i,j}^{(2)} = S_{i,j-1}^{(2)} \cdot B_{i,j}^{(2)} = \prod_{j'=1}^{j} B_{i,j'}^{(2)} = g^{\sum_{j'=1}^{j} t_{j'}}$$

  and sets the signature as $\sigma_{i,j} = (S_{i,j}^{(1)}, S_{i,j}^{(2)})$.
- $\mathsf{Verif}(params, I_i^{(E)}, \{I_j^{(D)}\}_{j \in [1,l]}, \sigma_{i,l}, m) \rightarrow$ {"Accept","Reject"}. Let l be the number of devices in the environment. Given the identity $I_i^{(E)}$ of the environment, the set of identities $\{I_j^{(D)}\}_{j \in [1,l]}$ of the devices in this environment, the multisignature $\sigma_{i,l} = (S_{i,l}^{(1)}, S_{i,l}^{(2)})$ (i.e. the signature generated by the l-th (last) device with identity $I_l^{(D)}$), and a message m, the response collector checks whether the following equation holds:

$$e(S_{i,l}^{(1)}, g) = e(H_3(m), S_{i,l}^{(2)}) \cdot e(H_1(I_i^{(E)}), h_1^l) \cdot e(\prod_{j=1}^{l} H_2(I_j^{(D)}), h_2).$$

  If the above equation holds, then the response collector outputs "Accept"; otherwise, it outputs "Reject".

Correctness. Let l be the number of devices in the same environment with identity $I_i^{(E)}$, $\{I_j^{(D)}\}_{j \in [1,l]}$ be the set of identities of the devices in this environment, $\sigma_{i,l} = (S_{i,l}^{(1)}, S_{i,l}^{(2)})$ be the multisignature and m be a message.

$$\begin{aligned}
e(S_{i,l}^{(1)}, g) &= e(g_m^{\sum_{j=1}^{l} t_j} \cdot g_i^{l\alpha} \cdot \prod_{j=1}^{l} g_j^{\beta}, g) \\
&= e(g_m^{\sum_{j=1}^{l} t_j}, g) \cdot e(g_i^{l\alpha}, g) \cdot e(\prod_{j=1}^{l} g_j^{\beta}, g) \\
&= e(g_m, g^{\sum_{j=1}^{l} t_j}) \cdot e(g_i, g^{l\alpha}) \cdot e(\prod_{j=1}^{l} g_j, g^{\beta}) \\
&= e(g_m, S_{i,l}^{(2)}) \cdot e(g_i, h_1^l) \cdot e(\prod_{j=1}^{l} g_j, h_2) \\
&= e(H_3(m), S_{i,l}^{(2)}) \cdot e(H_1(I_i^{(E)}), h_1^l) \cdot e(\prod_{j=1}^{l} H_2(I_j^{(D)}), h_2)
\end{aligned}$$

5 PROTOCOL EVALUATION

5.1 Security Model

Our security model for a 2-IBMS scheme follows the one in [13] defined for a Multi-Key Hierarchical Identity-Based Signature (MK-HIBS) scheme. Let $\mathcal{B}$ be a challenger interacting with an adversary

$\mathcal{A}$. We define the success probability of $\mathcal{A}$ in winning as follows:

$$\begin{aligned} Pr[(params, msk) \leftarrow \mathsf{Setup}(\lambda),\\ sk_i \leftarrow \mathcal{A}^{O_{\mathsf{KeyGen}_1}(\_)}(params),\\ \{sk_{i,j} \leftarrow \mathcal{A}^{O_{\mathsf{KeyGen}_2}(\_,\_)}(params)\}_{j\in[1,l]},\\ (m,\sigma) \leftarrow \mathcal{A}^{O_{\mathsf{Sign}}(\_,\_,\_)}(params) :\\ \mathsf{Verif}(params, I_i^{(E)}, \{I_j^{(D)}\}_{j\in[1,l]}, \sigma, m) \rightarrow \text{"Accept"}\\ \wedge (I_i^{(E)}, I_j^{(D)}, m, \sigma) \notin Q] < \varepsilon(\lambda) \end{aligned}$$

where O_{KeyGen_1} takes as input an identity $I_i^{(E)}$ and produces the output of $\mathsf{KeyGen}_1(params, msk, I_i^{(E)})$, O_{KeyGen_2} takes as inputs identities $I_i^{(E)}$ and $I_j^{(D)}$ and produces the output of $\mathsf{KeyGen}_2(params, \mathsf{KeyGen}_1(params, msk, I_i^{(E)}), I_j^{(D)})$, and O_{Sign_j} takes as inputs identities $I_i^{(E)}$ and $I_j^{(D)}$ and message m, and produces the output of $\mathsf{Sign}_j(params, sk_{i,j}, \sigma_{i,j-1}, m)$. Q denotes the set of pairs (m,σ) where $\mathcal{A}$ obtained a signature σ on m under identities $I_i^{(E)}$ and $I_j^{(D)}$ by querying O_{Sign_j}. We say that a 2-IBMS scheme is secure against forgery under chosen message and chosen identity attacks if $\varepsilon(\lambda)$ is negligible.

5.2 Sketch of the Security Proof

Let $\mathcal{A}$ be an adversary that wants to break the 2-IBMS scheme in the random oracle model. A challenger $\mathcal{B}$ interacts with $\mathcal{A}$ in order to solve the CDH problem. $\mathcal{B}$ is given a CDH tuple (g, g^a, g^b) such that h_2 is set to be equal to g^b. The goal of $\mathcal{B}$ is to output g^{ab}. $\mathcal{B}$ controls the hash functions H_1, H_2 and H_3.

Let $c \in [1, q_{H_2}]$, where q_{H_2} is the maximum number of queries to H_2 made by $\mathcal{A}$. When $\mathcal{A}$ makes the c-th distinct query to H_2 on input $I_j^{(D)}$, then $\mathcal{B}$ sets $I = I_j^{(D)}$. If $I_j^{(D)} = I$, then $\mathcal{B}$ includes g^a when computing the answer in G for a H_2's query. Otherwise, $\mathcal{B}$ simply computes a random element of G.

$\mathcal{B}$ chooses a random element $coin \in_R \{0,1\}$, so that $Pr[coin = 0] = 1 - 1/q_{H_3}$, where q_{H_3} is the maximum number of queries to H_3 made by $\mathcal{A}$. If $coin = 0$, then $\mathcal{B}$ includes g^a when computing the answer in G for a H_3's query. Otherwise, $\mathcal{B}$ simply computes a random element of G.

When $\mathcal{A}$ requests query to O_{KeyGen_2} on $I_i^{(E)}$ and $I_j^{(D)}$, $\mathcal{B}$ returns an answer comprising the element g^b if and only if $I_j^{(D)} \neq I$. Otherwise, $\mathcal{B}$ aborts.

$\mathcal{A}$ can request a query O_{Sign_j} on $I_i^{(E)}, I_j^{(D)}$ and m. If $I_j^{(D)} = I$ and $coin = 0$, the output from a query to H_3 embeds g^a. Moreover, $\mathcal{B}$ cannot compute the signing key corresponding to the identity $I_j^{(D)}$ because g^{ab} appears. Nevertheless, the outputs from queries to H_2 and H_3 might cause some cancellations that help $\mathcal{B}$ for computing a valid signature.

Eventually, $\mathcal{A}$ outputs a multisignature $\sigma^* = (S_{i,l}^{(1)*}, S_{i,l}^{(2)*})$ on m^*, $I_i^{(E)*}$ and $I_j^{(D)*}$ for $j \in [1,l]$ (for some value l). If σ^* is a valid multisignature, then it should satisfy the following equation:

$$e(S_{i,l}^{(1)*}, g) = e(H_3(m^*), S_{i,l}^{(2)*}) \cdot e(H_1(I_i^{(E)*}), h_1^l) \cdot e(\prod_{j=1}^{l} H_2(I_j^{(D)*}), h_2)$$

where $h_1 = g^\alpha$ and $h_2 = g^b$. Let $Pr[\sigma^* \text{ is valid}] \geq \varepsilon$. If $coin^* = 0$ or $I \neq I_j^{(D)*}$ for an index $j \in [1,l]$, then $\mathcal{B}$ aborts. Let $coin^* = 1$ and $I = I_J^{(D)*}$ for some $J \in [1,l]$. From the simulation of H_3, we have $H_3(m^*)$ is a uniform element in G not depending on g^a. However, for $j \neq J$, we have $H_2(I_j^{(D)*})^b = (g^b)^{v_j}$ for some value v_j from the list of the outputs from queries to H_2; and for J, $H_2(I_J^{(D)*})^b = g^{ab} \cdot (g^b)^v$ for some value v. Thus, $\mathcal{B}$ can recover g^{ab} from $\mathcal{A}$'s output and its knowledge of the outputs from queries to H_2 and H_3 if and only if $\mathcal{A}$'s forgery is a valid multisignature. Therefore, if $\mathcal{B}$ does not abort, then it can extract the value g^{ab} with probability at least ε.

5.3 Performance

In the above construction, the size of the public parameters is constant. In addition, the private key sk_i generated for an environment and the signing key $sk_{i,j}$ generated for a device have both constant size. Since involved entities have limited storage capacities, such result makes the construction practical.

Informally, the size of the multisignature linearly depends on the size of the group formed by all the devices' identities. The response collector must know at least the identity group description to perform its auditing task. However, we might suppose that the group does not change for a long time or has a short description. The same hypothesis was observed in [8, 16] for Identity-Based Group and Ring Signatures (IBGS and IBRS resp.). Similarly to previous works, we omit this cost, and thus the size of the multisignature does not depend on the number l of devices. Therefore, we reach the significant requirement for communication among IoT devices, that is exchanging compact components from device to device. Pairing computation, that is the most costly operation, is not required to generate the signature. Only multiplications and exponentiations in G happen, that have cheaper operational costs compared to the ones for pairings.

While the response collector has to keep all the identities $I_i^{(E)}$ and $I_j^{(D)}$, where $j \in [1,l]$, we suggest that they are small elements in $\{0,1\}^*$. The verification only requires a constant number of pairing operations to be performed. The main operation costs come from multiplications and exponentiations in G.

Following the example of ECUs embedded in a vehicle, the response collector requires the identities of the vehicle and of all the ECUs with respect to their specific role. A "luxury" car is composed of 100 ECUs. Therefore, the response collector must store around 100 identities per vehicle. Let N be the number of vehicles to be audited. The response collector must keep $N * (100 + 1) = 101N$ identifying elements.

6 APPLICATION: TRANSPORTATION

We propose to illustrate our 2-IBMS scheme ensuring secure bootstrap and aggregation response validation in the case of smart transportation. We consider the scenario of periodic motor vehicle inspection.

Assume that a motor vehicle factory designs, manufactures and sells vehicles on the automotive market. Each vehicle from this factory is considered as a confined environment and retains a main

embedded system defined with a unique identity corresponding to this vehicle. The factory acts as the off-line TTP and delivers private keys to the vehicles.

A vehicle in this factory is equipped with multiple ECUs with various functionalities such as engine control module, powertrain control module, transmission control module, brake control module and so on. Thanks to our solution, each ECU is assigned a unique identity with respect to its role (e.g. transmission control module) and its environment (vehicle ID).

At the inspection phase, ECUs reply one after the other to the inspection request by signing their answer. The last participating ECU computes the aggregate response which is further forwarded to the inspector (e.g. the police officer). The latter finally verifies the multisignature: the vehicle passes the inspection only if the received multisignature is successfully validated.

7 RELATED WORK

Boneh and Franklin [4] succinctly showed that an Identity-Based Encryption scheme implies a signature scheme. Some papers [1, 7] suggest how to transform a classic signature scheme into an IBS scheme using some certification method. A classic MS scheme may be extended into an IBMS scheme [9]. However, the multisignature is not compact since the certificates must be included in each signature. Another attempt tries to aggregate the signatures in the certificates [5]. However, the public keys embedded into the signatures cannot be aggregate. Therefore, it appears that there is no simple transformation of a compact signature scheme into an identity-based one [2].

The IBMS scheme in [2] enables the signing keys to be generated regarding identities of parties and the latter compute a multisignature together. However, the set of identities of the signers are known in advance since the set is needed as input for the signature generation algorithm. There is no hierarchy since all the parties are at the same level. An IBMS scheme is given in [10] as a base for an Identity-Based Aggregate Signature scheme. The signers create their own signatures and then aggregate them at the end to obtain the multisignature. This can be done sequentially, taking as input the signature of the previous signer to generate the signature of the current signer. Again, no hierarchy can be supported.

A HIBS scheme have been defined along with the corresponding security model in [11]. The number of levels in the hierarchy is arbitrary. A MK-HIBS scheme in [13] considers a hierarchy of the parties. A TTP plays the root as in our case. Using our context, the environment would be met just below the TTP. Instead of having all the devices as siblings, the sequence that they form can be translated as a hierarchical path, the first device being just below the environment, the second device under the first device, and so on, until the last device at the bottom of the path.

The following results are inspired from [13] where l devices participate in signing. Table 1 compares the computational costs in schemes in [2, 10, 13] and ours. The results for the MK-IBMS scheme [13] are for a 1-level hierarchy. Let "S" be for signing and "V" be for verification. Let "Mult" be the number of multiplications in G, "Exp" be the number of exponentiation in G, "Pairing" be the number of pairing computations in G_T, "Hash" be the number of hash operations, "mMult" be the number of modular multiplications and "mExp" be the number of modular exponentiations. Let l be the number of signing devices.

Table 1: Computational Costs for IBMS Schemes

	IBMS [2]		IBMS [10]		MK-HIBS [13]		our 2-IBMS	
	S	V	S	V	S	V	S	V
Mult			$3l-2$	$l-1$	l	$l-1$	$3l$	l
Exp			$2l$	0	2	0	$2l$	1
Pairing			0	3	0	3	0	4
Hash	$l(l+1)$	$l-1$	l	$l+1$	l	$l+1$	l	$l+2$
mMult	l^2+l-1	l						
mExp	$2l$	2						

Schemes in [2, 10, 13] do not involve a hierarchical organization of the parties. If we extend our SNoT context to the aforementioned schemes, then environment and devices are at the same level such that the environment does not participate in the generation of the signatures. Hence, results in [2, 10, 13] do not actually take into account the presence of the environment; while in our scheme, the environment acts as the parent of the devices and thus this incurs light additional costs. More precisely, one extra computation per operation in the signing and verification processes are due to the extra level in the hierarchy induced by the environment compared to the results in [13]. Nevertheless, in [13], no locally unique identification of the devices is possible. Therefore, our scheme remains efficient and practical while satisfies the concern of secure bootstrap in a confined environmental context.

8 CONCLUSION

Trust establishment in IoT is a major concern due to the diversity of device origins, computational and communication constraints, and lack of governance. In this paper, we addressed the above problem by assigning local identities to devices based their role and environment and by generating a compact unique response for the environment. The newly proposed scheme includes a secure bootstrap mechanism for device identification as well as a message attestation mechanism for aggregate response validation. The presented solution is provably secure in the random oracle model.

ACKNOWLEDGMENTS

This work was supported by the EU FP7 ERANET program under grant CHIST-ERA-2016 UPRISE-IOT.

REFERENCES

[1] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. 2004. Security Proofs for Identity-Based Identification and Signature Schemes. In *Proceedings of the 23rd International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04)*, Christian Cachin and Jan L. Camenisch (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 268–286. https://doi.org/10.1007/978-3-540-24676-3_17

[2] Mihir Bellare and Gregory Neven. 2006. Identity-Based Multi-signatures from RSA. In *Proceedings of the 7th Cryptographers' Track at the RSA Conference on Topics in Cryptology (CT-RSA'07)*. Springer-Verlag, Berlin, Heidelberg, 145–162. https://doi.org/10.1007/11967668_10

[3] Alexandra Boldyreva. 2003. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography (PKC '03)*. Springer-Verlag, London, UK, UK, 31–46. http://dl.acm.org/citation.cfm?id=648120.747061

[4] Dan Boneh and Matt Franklin. 2001. Identity-Based Encryption from the Weil Pairing. In *Proceedings of the 21th Annual International Conference on Advances in Cryptology (CRYPTO'01)*, Joe Kilian (Ed.), Vol. 2139. Springer Berlin Heidelberg, 213–229. https://doi.org/10.1007/3-540-44647-8_13

[5] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. 2003. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'03)*. Springer-Verlag, Berlin, Heidelberg, 416–432. http://dl.acm.org/citation.cfm?id=1766171.1766207

[6] Kevin Butler, Toni R. Farley, Patrick McDaniel, and Jennifer Rexford. 2010. A Survey of BGP Security Issues and Solutions. *Proc. IEEE* 98, 1 (Jan 2010), 100–122. https://doi.org/10.1109/JPROC.2009.2034031

[7] Y. Dodis, J. Katz, S. Xu, and M. Yung. 2003. Strong key-insulated signature schemes. In *Proceedings of the International Conference on Public Key Cryptography (PKC'03)*, Y. Desmedt (Ed.), Vol. 2567. Springer-Verlag, Berlin, Heidelberg, 130–144.

[8] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. 2004. Anonymous Identification in Ad Hoc Groups. In *Proceedings of the 23rd International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04)*, Christian Cachin and Jan L. Camenisch (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 609–626. https://doi.org/10.1007/978-3-540-24676-3_36

[9] David Galindo, Javier Herranz, and Eike Kiltz. 2006. On the Generic Construction of Identity-based Signatures with Additional Properties. In *Proceedings of the 12th International Conference on Theory and Application of Cryptology and Information Security (ASIACRYPT'06)*. Springer-Verlag, Berlin, Heidelberg, 178–193. https://doi.org/10.1007/11935230_12

[10] Craig Gentry and Zulfikar Ramzan. 2006. Identity-Based Aggregate Signatures. In *Proceedings of the 9th International Conference on Theory and Practice of Public-Key Cryptography (PKC'06)*. Springer-Verlag, Berlin, Heidelberg, 257–273. https://doi.org/10.1007/11745853_17

[11] Craig Gentry and Alice Silverberg. 2002. Hierarchical ID-Based Cryptography. In *Proceedings of the Advances in Crypotology 9th International Conference on Theory and Application of Cryptology and Information Security (ASIACRYPT'02)*, Yuliang Zheng (Ed.), Vol. 2501. Springer Berlin Heidelberg, 548–566. https://doi.org/10.1007/3-540-36178-2_34

[12] K. Itakura and K. Nakamura. 1983. A Public Key Cryptosystem suitable for Digital Multi-Signatures. *NEC Research and Development* 71 (1983), 1–8.

[13] Hoon Wei Lim and Kenneth G. Paterson. 2007. Multi-key Hierarchical Identity-Based Signatures. In *Proceedings of the 11th IMA International Conference in Cryptography and Coding*, Steven D. Galbraith (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 384–402. https://doi.org/10.1007/978-3-540-77272-9_23

[14] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. 2006. Sequential Aggregate Signatures and Multisignatures Without Random Oracles. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'06)*, Serge Vaudenay (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 465–485. https://doi.org/10.1007/11761679_28

[15] Silvio Micali, Kazuo Ohta, and Leonid Reyzin. 2001. Accountable-subgroup Multisignatures: Extended Abstract. In *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*. ACM, New York, NY, USA, 245–254. https://doi.org/10.1145/501983.502017

[16] Lan Nguyen. 2005. Accumulators from Bilinear Pairings and Applications. In *Proceedings of the 2005 International Conference on Topics in Cryptology (CT-RSA'05)*. Springer-Verlag, Berlin, Heidelberg, 275–292. https://doi.org/10.1007/978-3-540-30574-3_19

[17] Karen Seo, Charles Lynn, and Stephen Kent. 2001. Public-key infrastructure for the Secure Border Gateway Protocol (S-BGP). In *Proceedings of the DARPA Information Survivability Conference (DISCEX '01)*, Vol. 1. 239–253 vol.1. https://doi.org/10.1109/DISCEX.2001.932219

[18] Adi Shamir. 1985. Identity-based Cryptosystems and Signature Schemes. In *Proceedings of the Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'84)*. Springer-Verlag New York, Inc., New York, NY, USA, 47–53. http://dl.acm.org/citation.cfm?id=19478.19483

[19] Meiyuan Zhao, Sean W. Smith, and David M. Nicol. 2005. Evaluating the Performance Impact of PKI on BGP Security. In *Proceedings of 4th Annual PKI Research Workshop (PKI'05)*.