applied sciences

MDPI

Article

Continuous Entity Authentication in the Internet of Things Scenario

Alfredo De Santis [1], Anna Lisa Ferrara [2], Manuela Flores [1,3] and Barbara Masucci [1,*]

[1] Department of Computer Science, University of Salerno, 84084 Fisciano, Italy; ads@unisa.it (A.D.S.); manuela.flores@unipa.it or mflores@unisa.it (M.F.)
[2] Department of Biosciences, Division of Computer Science, University of Molise, 86100 Campobasso, Italy; annalisa.ferrara@unimol.it
[3] Department of Mathematics and Computer Science, University of Palermo, 90133 Palermo, Italy
* Correspondence: bmasucci@unisa.it

Abstract: In the context of the Internet of Things (IoT), the proliferation of identity spoofing threats has led to the need for the constant entity verification of devices. Recently, a formal framework has been proposed to study resistance to impersonation attacks for One-Message Unilateral Entity Authentication (OM-UEA) schemes, in which the prover continuously authenticates itself through the use of a sequence of authentication messages. Given the limited computing power of the parties (particularly the prover) and the often limited bandwidth channel, in the IoT scenario it is desirable to design unilateral entity authentication schemes that require the use of a single message per session and light computations. In this paper, we first show that OM-UEA schemes can be implemented through digital signatures and that a weak form of unforgeability is sufficient to achieve security against active adversaries. We then apply the signature scheme proposed by Yang et al. in ASIACCS 2020 to our framework, resulting in an OM-UEA scheme that requires minimal computational effort and low storage requirements for the prover. Inspired by this last construction, we propose an OM-UEA scheme based on the hardness of the discrete logarithm problem, which further improves the computational performance for the prover. Our findings offer feasible options for implementing secure continuous entity authentication in IoT applications.

Keywords: authentication schemes; one-message unilateral entity authentication; continuous entity authentication

Citation: De Santis, A.; Ferrara A.L.; Flores, M.; Masucci, B. Continuous Entity Authentication in the Internet of Things Scenario. *Appl. Sci.* **2023**, *13*, 5945. https://doi.org/10.3390/app13105945

Academic Editor: Yoshiyasu Takefuji

Received: 16 March 2023
Revised: 27 April 2023
Accepted: 9 May 2023
Published: 11 May 2023

1. Introduction

The Internet of Things (IoT) has become a ubiquitous presence in our daily lives, but it also raises security concerns that must be addressed. To maintain the privacy and security of connected devices, continuous entity authentication has become a crucial aspect of IoT security. Unlike static authentication, which only verifies the identity of a device or user at the start of a session, continuous authentication provides ongoing security throughout the entire session. This method helps prevent session hijacking and enables quick authentication for frequent communication.

De Santis et al. [1] recently introduced a new cryptographic primitive, called One-Message Unilateral Entity Authentication (OM-UEA), which is useful for implementing continuous unilateral entity authentication where the parties, known as the prover and the verifier, do not share any secret information. In this approach, the prover continuously authenticates itself through the use of a sequence of authentication messages, while the verifier is able to establish confidence in the identity of the prover and to verify that a sequence of messages is indeed generated by the same entity.

Unilateral entity authentication is often required in the context of the Internet of Things (IoT), where different devices, such as sensors, smart TVs, smart locks and so on, need to continuously authenticate themselves to an IoT gateway, where data can either be sent to

the cloud for analysis or analyzed locally. A good example is the use of tamper-resistant wearable wristbands that, when in range, continuously authenticate to the IoT gateway to constantly prove that a prisoner is currently in the perimeter of a prison or that a seriously ill person is not leaving the ward.

One-Message Unilateral Entity Authentication schemes are particularly suitable for modeling scenarios in which a prover must authenticate multiple times to the verifier, providing a sequence of authentications so that the verification process must take into account both the order between messages and the fact that the prover's identity must be the same throughout the entire sequence. To achieve this, the concept of "history" is introduced, where an authentication message is linked to previous ones, allowing the verification process to depend on authentication messages previously received and accepted by the verifier. For example, a simple protocol for remotely instructing a server (verifier), such as a smart home appliance, to execute a sequence of actions can use each authentication message to trigger a corresponding action. With a One-Message Unilateral Authentication scheme, the verifier can verify the authenticity of the source sending the commands and ensure the correct execution of the workflow by requiring certain actions to be executed only after the corresponding authentication messages have been received as part of the same sequence of steps.

In [1], the authors developed a comprehensive framework to help designers assess the resistance of any OM-UEA scheme to impersonation attacks. Moreover, they proposed three generic constructions for OM-UEA schemes, which are based on three distinct cryptographic primitives, and analyzed their security with respect to different security definitions. More precisely, the first construction is based on one-way function families, whereas the second and third constructions are based on one-way permutation families and trapdoor one-way permutation families, respectively. No practical instantiations of these constructions have been proposed that can be used in applications where the parties have limited computing resources. Hence, efficient solutions are needed to enable seamless authentication in the Internet of Things (IoT) ecosystem.

1.1. Contribution

In this paper, we first show that OM-UEA schemes can be implemented using digital signatures and that a weak form of unforgeability is sufficient to achieve security against active adversaries. Furthermore, we instantiate our construction with the lightweight signature scheme recently proposed by Yang et al. [2] and show that it results in a one-message unilateral entity authentication scheme that requires low computational effort, as well as low storage overhead, for the prover. Specifically, to compute each authentication message the prover only needs to perform two modular multiplications and three modular additions. Moreover, the private key in such a scheme is a bit string having a small size.

Afterwards, we consider the problem of further reducing the computational complexity for the prover, with the goal of offering a different option for implementing secure continuous entity authentication in IoT applications. In particular, we propose a different construction for OM-UEA schemes, whose security relies on the hardness of the discrete-logarithm problem. In such a construction, the prover only needs to perform one modular multiplication and one modular addition for each authentication message.

1.2. Related Works

Despite the increasing number of IoT applications, including smart homes, smart cities, smart cars, and medical and healthcare equipment, very few solutions for the continuous authentication of IoT devices have been proposed in recent years [3–13]. Some of them are based on device fingerprints and require additional hardware [7], whereas others rely on time-bound key generation techniques [11]. Recent proposals include lightweight protocols allowing continuous d2d authentication for resource-constrained devices [10,12,13] and an efficient protocol for continuous entity authentication, requiring a single-message, which is based on Elliptic Curve Cryptography [3]. Moreover, some recent results leverage

blockchain technologies to authenticate IoT devices [14–16]. A survey of methods designed to achieve continuous authentication in the IoT scenario can be found in [17].

Continuous message authentication is a research topic which is strictly related to continuous entity authentication. Indeed, entity authentication is a specific type of message authentication that only allows the authentication of entities and not arbitrary messages. Data authentication and integrity can be achieved via digital signature schemes. However, digital signatures are usually impractical to use with highly resource-constrained devices. Indeed, despite the large number of constructions which have been proposed so far, they do not provide efficient computations at both the signer's and verifier's side, or small sized private keys and signatures at the same time.

The problem of reducing the complexity of the operations at the signer's side has been considered by Even et al. [18], who proposed the online/offline paradigm. Such a paradigm relies on a trusted party to pre-compute intermediate tokens via expensive operations in an offline stage; such tokens are saved by the signer, which later uses them to sign messages. The drawback of such a solution is that it requires large storage overhead for the signer.

The problem of achieving space efficiency at the signer's side has been later considered by Shamir et al. [19], who proposed an improved online/offline signature scheme. However, their proposal still requires a storage which is linear in the number of signatures that the signer can generate, making the scheme unsuitable for storage-limited signer devices. Other solutions in the online/offline paradigm have been proposed in [20,21]. Unfortunately, none of the above solutions is suitable in a scenario where the signer is a device having both limited computing power and low storage capability, such as an IoT device.

In order to achieve fast signing and verifying, Lamport [22] proposed a one-time signature scheme using one-way functions; however, the drawback of his scheme is the requirement for the public key, which has a large size. Message encoding techniques to reduce the public key size for one-time signature schemes have later been proposed by Reyzin et al [23], but the resulting scheme results in signatures of large sizes. Moreover, the schemes in [22,23] can be used to sign only one message and require expensive key generation for each message to be signed.

Yang et al. [2] proposed a lightweight signature framework, called *LiS*, which is suitable for continuous message authentication in IoT applications. Such a framework includes two signature schemes, called *LiS1* and *LiS2*.

Digital signatures are often used along the Trusted Platform Module (TPM) to achieve device authentication in IoT settings [24]. In this approach, the TPM is associated with a public–private key pair, with the private key securely stored in the TPM. The TPM can then compute digital signatures, which can be used to authenticate the identity of the device. The OM-UEA scheme based on digital signatures we proposed can be utilized along with a TPM to efficiently generate a sequence of identity authentications to be used in applications where the order of the authentication messages needs to be considered in the verification process.

1.3. Organization

The remainder of the paper is organized as follows: in Section 2 we first recall the theory regarding the discrete-logarithm problem, digital signatures, and one-message unilateral entity authentication schemes. In Section 3, we propose and analyze our digital-signature based construction, and in Section 4 we propose and analyze our construction based on the hardness of the discrete logarithm problem. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. Notation

In this paper, we use the standard notation for describing probabilistic algorithms and experiments, which was introduced in [25]. Given a probabilistic algorithm $A(\cdot,\cdot,\ldots)$, we write $a \leftarrow A(x,y,\ldots)$ to denote the experiment of running A on input $x,y,\ldots$, producing as output a, where the probability is taken over the randomness used by the algorithm A. Given a set X, we write $x \leftarrow X$ to denote the experiment of randomly choosing an element x from X. On the other hand, if w is neither an algorithm nor a set, then we write $x \leftarrow w$ to denote the assignment of w to x. We say that a function $\epsilon : N \rightarrow R$ is *negligible* if, for any constant $c > 0$, there exists an integer τ_c such that for all $\tau \geq \tau_c$, $\epsilon(\tau) < \tau^{-c}$.

2.2. The Discrete-Logarithm Problem

In this section, we recall the *discrete-logarithm problem* in a cyclic group G with order q and generator g. Given a uniform element $h \in G$, we are required to compute the unique value $x \in Z_q$ such that $g^x = h$. Such a value is called the *discrete logarithm of h with respect to g* and is denoted by $x = \log_g h$. Let $\mathcal{G}$ be a *group generation algorithm* that, on input 1^τ, outputs a description of a cyclic group G, its order q, and a generator $g \in G$. The discrete-logarithm problem is formalized in the next definition.

Definition 1. *Let 1^τ be a security parameter. The discrete-logarithm experiment for an adversary A is as follows:*

Experiment $\mathbf{DLog}_{A,\mathcal{G}}(1^\tau)$
$(G,q,g) \leftarrow \mathcal{G}(1^\tau)$
$h \leftarrow G$
$x \leftarrow A(1^\tau, G, q, g, h)$
if $g^x = h$
then *output* 1
else *output* 0

The advantage of A is defined as

$$\mathbf{Adv}^{\mathbf{DLog}}_{A,\mathcal{G}}(1^\tau) = Pr[\mathbf{DLog}_{A,\mathcal{G}}(1^\tau) = 1].$$

The discrete-logarithm problem is hard relative to $\mathcal{G}$ if, for each probabilistic polynomial time adversary A, the advantage $\mathbf{Adv}^{\mathbf{DLog}}_{A,\mathcal{G}}(1^\tau)$ is negligible in τ.

There are various classes of cyclic groups in which the discrete-logarithm problem is believed to be hard. In this paper, we considered cyclic groups G, which are subgroups of Z^*_p, where p is a prime and the order of the group is a prime q such that q divides $p-1$.

2.3. Digital Signatures

Digital signatures are a fundamental tool in different protocols designed to work in a distributed environment, since they provide message authentication and integrity. We start by recalling the definition of digital signature schemes.

Definition 2. *A triple $\Sigma = (KGen, Sign, Vrfy)$ of polynomial time algorithms is a digital signature scheme if:*

- *The probabilistic key-generation algorithm $KGen$, given a security parameter 1^τ, produces as outputs a public key pk and a secret key sk.*
- *The probabilistic signing algorithm $Sign$, given the private key sk and a message $m \in \{0,1\}^*$, produces as output a signature σ.*
- *The deterministic verification algorithm $Vrfy$, given the public key pk, a message m, and a signature σ, outputs a bit b, where $b = 1$ means that the signature is valid, whereas $b = 0$ means that the signature is invalid.*

It is required that, for every pair (pk, sk) which can be output by $KGen(1^\tau)$, every message $m \in \{0,1\}^*$, and every σ which can be output by $Sign(sk, m)$, it holds that $Vrfy(pk, m, \sigma) = 1$.

Given the public key pk of a signer S, a *forgery* is a pair (m, σ), where σ is a valid signature for m and m has not been previously signed by S. We are interested in signature schemes where an adversary should be unable to forge a signature for any message m even if it is given access to signatures for a set of other messages not of his choice. In particular, for our goals it is sufficient to consider a weak notion of security known as *existential unforgeability under known message attack* [25]. We formalize such a goal in the next definition.

Definition 3. *Let $\Sigma = (KGen, Sign, Vrfy)$ be a digital signature scheme. The signature-forgery experiment for an adversary A is as follows:*

Experiment **Sig-Forge**$_{A,\Sigma}(1^\tau)$
$(pk, sk) \leftarrow KGen(1^\tau)$
Let Q be a set of pairs (message, signature)
$(m, \sigma) \leftarrow A(pk, Q)$
if $Vrfy(pk, m, \sigma) = 1$ *and* $(m, \sigma) \notin Q$
then *output* 1
else *output* 0

The advantage of A is defined as

$$\mathbf{Adv}^{\textbf{Sig-Forge}}_{A,\Sigma}(1^\tau) = Pr[\textbf{Sig-Forge}_{A,\Sigma}(1^\tau) = 1].$$

The signature scheme Σ is existentially unforgeable under a known-message attack if, for each polynomial time adversary A, the advantage $\mathbf{Adv}^{\textbf{Sig-Forge}}_{A,\Sigma}(1^\tau)$ is negligible in τ.

2.4. OM-UEA Schemes

A One-Message Unilateral Entity Authentication (OM-UEA) scheme enables the prover to establish its identity to the verifier without any pre-existing shared secret information. The prover possesses certain secret information, which it uses to create an authentication message that is sent to the verifier. The verifier can then use publicly available information to test the authenticity of the received message. This enables the prover to establish its identity to the verifier without requiring any prior shared secret between the two parties.

OM-UEA schemes have been introduced by De Santis et al. [1]. In particular, they proposed a theoretical characterization for OM-UEA schemes by providing formal definitions of security with respect to different kinds of adversarial behaviors. In their framework, the prover generates a sequence of authentication messages which are ordered according to the session counter and the verification procedure for the i-th message might also depend on the sequence of authentication messages received for previous sessions, which is denoted by $hist_i$, where $hist_0$ corresponds to the empty sequence. If the scheme is successful, the verifier is assured that it is in communication with the actual prover. We recall the formal definition of OM-UEA schemes. In particular, we consider schemes which can be used for at most $\ell = \ell(\tau)$ consecutive authentications, where $\ell(\cdot)$ is a polynomially-bounded function fixed at the time of parameter generation.

Definition 4 ([1])**.** *An OM-UEA scheme for ℓ authentication sessions is a triple of polynomial time algorithms $\Pi = (Gen, Auth, Ver)$ such that:*

- *The probabilistic parameter-generation algorithm Gen, on inputs a security parameter 1^τ and the number ℓ of sessions, produces three values as output: the public information, the private information for the prover, and the initial state. This triple is represented as $(pub, priv, s_0)$.*

- *The probabilistic message authentication algorithm $Auth$, on inputs the security parameter 1^τ, the current session counter i, where $1 \leq i \leq \ell$, the public information pub, the private information $priv$, and the current state s_{i-1}, outputs the pair (a_i, s_i), where a_i is the authentication message for the i-th session and s_i is the new state.*
- *The deterministic verification algorithm Ver, on inputs the security parameter 1^τ, the current session counter i, where $1 \leq i \leq \ell$, an authentication message a_i, the public information pub, and the sequence of authentication messages $hist_{i-1} = (a_1, \ldots, a_{i-1})$ generated by the previous $i-1$ authentication sessions, where $hist_0$ corresponds to the empty sequence, outputs a bit b, where a value of 1 indicates a* valid *authentication and a value of 0 indicates an* invalid *authentication.*

For every triple $(pub, priv, s_0)$ output by $Gen(1^\tau, \ell)$ and every session counter $1 \leq i \leq \ell$, if we compute (a_j, s_j) using the function $Auth(1^\tau, j, pub, priv, s_{j-1})$ for each $j = 1, \ldots, i$, then the function $Ver(1^\tau, i, a_i, pub, hist_{i-1})$ should return the value 1.

De Santis et al. [1] formalized security requirements for OM-UEA schemes by considering different kinds of adversaries. In their model, the adversary is able to control the communication channel between the parties and its goal is to deceive the verifier into considering as valid an authentication message that was not actually generated by the prover. They considered both passive and active adversarial behaviors. A passive adversary is only allowed to passively observe different executions of the scheme and to collect authentication messages generated by the prover, whereas an active one can also modify some authentication messages before they reach their destination. Moreover, they considered both static and adaptive adversaries. Informally, a static adversary randomly chooses the session to be attacked, whereas the adaptive one uses the information it progressively learns to makes its choice. For both passive and active attacks, De Santis et al. [1] examined the relationships between security definitions that arise from adaptive and static adversarial behaviors. In particular, they showed the equivalence between security against static adversaries and security against adaptive ones. On the other hand, they also proved that security against active static attacks implies security against passive static attacks, while the opposite does not hold. Thus, from now on, we will only consider active static adversaries. In the following, we recall the definition of such a kind of adversary.

An active static adversary works in two phases, so it is denoted by $\texttt{ACT-STAT} = (\texttt{ACT-STAT}_1, \texttt{ACT-STAT}_2)$. In the first phase $\texttt{ACT-STAT}_1$, on inputs the security parameter 1^τ and the maximum number of sessions ℓ, it chooses a pair of session counters (i, j) between 1 and t, where $t = t(\tau)$ is the polynomial corresponding to the running time of the adversary, such that $j \leq i$. This models the fact that $\texttt{ACT-STAT}$ has chosen to attack the i-th authentication session and to modify the last $i - j$ authentication messages (clearly, if $j = i$, the sequence of modified messages is empty). Notice that such a choice occurs before the algorithm Gen is run. For each $1 \leq i \leq \ell$, we define an algorithm $Auth_{i-1}$ which, given the security parameter 1^τ, the public information pub, the private information $priv$, and the sequence $states_{i-1} = (s_0, \ldots, s_{i-2})$ of the first $i-1$ states, outputs the sequence $hist_{i-1} = (a_1, \ldots, a_{i-1})$ by running $Auth(1^\tau, j, pub, priv, s_{j-1})$ for $j = 1, \ldots, i-1$, where $hist_0$ corresponds to the empty sequence.

In the second phase, $\texttt{ACT-STAT}_2$ is able to obtain the sequence $hist_{i-1} = (a_1, \ldots, a_{i-1})$ of authentication messages generated for the first $i-1$ sessions. Afterwards, $\texttt{ACT-STAT}_2$, on inputs 1^τ, the public information pub, the pair of session counters (i, j), and the sequence $hist_{i-1}$, outputs a sequence of *fake authentication messages* $(a'_j, \ldots, a'_{i-1}, a'_i)$ and succeeds if the verifier accepts all of them. In particular, $a'_j, a'_{j+1}, \ldots, a'_{i-1}$ will replace the authentication messages $a_j, a_{j+1}, \ldots, a_{i-1}$, which are the last $i - j$ elements of the sequence $hist_{i-1}$; such a modified sequence $(a_1, \ldots, a_{j-1}, a'_j, \ldots a'_{h-1})$ is denoted by $hist'_{h-1}$, for any $h = j, \ldots, i$. The above security requirement is formalized in the next definition.

Definition 5 ([1] $[\ell\texttt{-IMP-A-ST}]$)**.** *Let Π be an OM-UEA scheme. The active static impersonation experiment for an active static adversary $\texttt{ACT-STAT} = (\texttt{ACT-STAT}_1, \texttt{ACT-STAT}_2)$ is as follows:*

Experiment $\mathbf{Imp}_{\texttt{ACT-STAT},\Pi}(1^\tau)$
$(i,j) \leftarrow \texttt{ACT-STAT}_1(1^\tau, \ell)$
$(pub, priv, s_0) \leftarrow Gen(1^\tau, \ell)$
$hist_{i-1} \leftarrow \texttt{Auth}_{i-1}(1^\tau, pub, priv, states_{i-1})$
$(a'_j, \ldots, a'_i) \leftarrow \texttt{ACT-STAT}_2(1^\tau, pub, i, j, hist_{i-1})$
if $\forall h \in [j,i]$
$Ver(1^\tau, h, a'_h, pub, hist'_{h-1}) = 1$
where $hist'_{h-1} = (a_1, \ldots, a_{j-1}, a'_j, \ldots, a'_{h-1})$
then return 1
return 0

The advantage of ACT-STAT *is defined as*

$$\mathbf{Adv}^{\mathbf{Imp}}_{\texttt{ACT-STAT},\Pi}(1^\tau) = Pr[\mathbf{Imp}_{\texttt{ACT-STAT},\Pi}(1^\tau) = 1].$$

*The OM-UEA scheme for ℓ session $\Pi = (Gen, Auth, Ver)$ is secure in the sense of ℓ-*IMP-A-ST *if, for all probabilistic-polynomial time active static adversaries* ACT-STAT*, the advantage of* ACT-STAT *in the impersonation experiment is negligible in the security parameter τ.*

De Santis et al. [1] proposed three constructions for OM-UEA schemes and analyzed their security with respect to the different security notions proposed in their paper. More precisely, the first construction is based on one-way function families and does not require the parameter ℓ to be fixed in advance. On the other hand, the second and third constructions are based on one-way permutation families and trapdoor one-way permutation families, respectively. In particular, the second construction requires the parameter ℓ to be fixed in advance, whereas the third one supports an arbitrary polynomial number of sessions. While the first construction does not provide security against active static attacks but only against passive attacks, the other two constructions satisfy such a notion of security. However, the drawback of such solutions is that they are not practical for use in applications where the parties have limited computing power and limited bandwidth channel.

3. A Construction Based on Digital Signatures

In this section, we consider the problem of constructing an OM-UEA scheme by using a different primitive, compared to the constructions in [1]. We propose a construction, resulting in an OM-UEA for ℓ sessions, which uses as a building block a digital signature scheme. We refer to the proposed construction as the Digital Signature-based Entity Authentication Construction (DS-EAC).

The Digital Signature-based Entity Authentication Construction (DS-EAC) . Let $\Sigma = (KGen, Sign, Vrfy)$ be a digital signature scheme and let $\ell = \ell(\tau)$, where $\ell(\cdot)$ is a polynomially-bounded function. We construct a one-message unilateral entity authentication scheme, $\Pi = (Gen, Auth, Ver)$, for ℓ sessions as follows:

- The parameter-generation algorithm Gen, on inputs a security parameter 1^τ and the number of sessions ℓ, outputs the tuple $(pub, priv, s_0)$ constructed as follows:
  - Runs the key-generation algorithm $KGen(1^\tau)$ to obtain the pair $(pub, priv)$;
  - Sets $pub = pk$ and $priv = sk$;
  - Sets the initial state s_0 to be empty.
- The authentication message-generation $Auth$, on inputs a security parameter 1^τ, a session counter $1 \leq i \leq \ell$, the public value pub, the secret key $priv$ for the prover, and the previous state s_{i-1}, outputs the authentication message a_i and the state s_i, as follows:
  - Runs the signing algorithm $Sign$, on inputs the private key sk and the message i, in order to produce the signature $\sigma_i = Sign(sk, i)$;
  - Sets a_i to the pair (i, σ_i);
  - Outputs a_i and an empty state s_i.

- The verification algorithm Ver, on inputs a session counter i such that $1 \leq i \leq \ell$, an authentication message a_i, the public value pub, and the sequence of authentications $hist_{i-1}$ obtained by the first $i-1$ sessions, where $hist_0$ is the empty sequence, runs the verification algorithm $Vrfy$ on inputs the public key pk, the session counter i and the signature σ_i, and outputs the same output as $Vrfy(pk, i, \sigma_i)$.

Notice that the DS-EAC construction also supports an arbitrary polynomial number of authentication sessions, i.e., the construction works even in the case where the parameter ℓ is not fixed in advance.

A pictorial representation of the DS-EAC is shown in Figure 1.

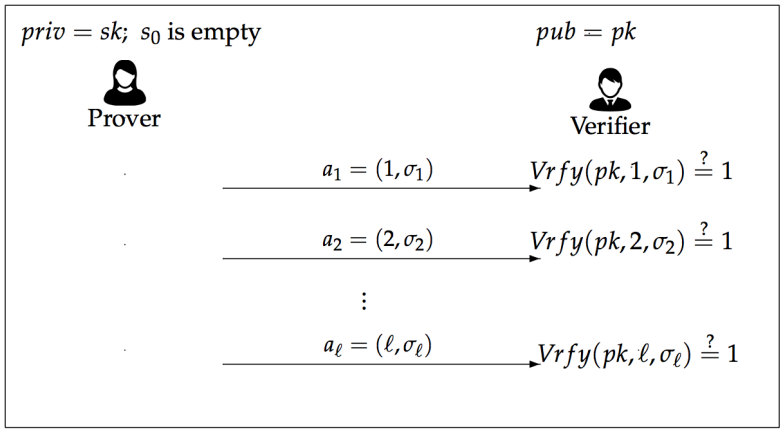

Figure 1. A Pictorial Representation of the DS-EAC.

We analyze the security of the DS-EAC against active static attacks. In particular, we are interested in establishing which security properties have to be satisfied by the underlying signature scheme in order for the resulting OM-UEA scheme to be secure against attacks carried out by active static adversaries. The next theorem shows that a weak form of unforgeability for the digital signature scheme, i.e., the existential unforgeability against a known-message attack, is sufficient to achieve the desired level of security for the DS-EAC.

Theorem 1. *If $\Sigma = (KGen, Sign, Vrfy)$ is a digital signature scheme that is existentially unforgeable under known-message attack, then the OM-UEA scheme for ℓ sessions $\Pi = (Gen, Auth, Ver)$ produced by the DS-EAC is secure in the sense of ℓ-*IMP-A-ST.

Proof. Let us suppose, for the sake of contradiction, that the OM-UEA scheme Π produced by the DS-EAC is not secure in the sense of ℓ-IMP-A-ST. Therefore, there exists a polynomial time adversary $\mathtt{ACT\text{-}STAT} = (\mathtt{ACT\text{-}STAT}_1, \mathtt{ACT\text{-}STAT}_2)$ whose advantage $\mathbf{Adv}^{\mathbf{Imp}}_{\mathtt{ACT\text{-}STAT},\Pi}(1^\tau)$ in the experiment $\mathbf{Imp}_{\mathtt{ACT\text{-}STAT},\Pi}(1^\tau)$ is non-negligible. We will show that there exists a polynomial time adversary A which, on input a public key pk along with a set Q of pairs consisting of a message and the corresponding signature, uses the adversary ACT-STAT to forge a valid signature on a message m which does not belong to Q.

Let Q be the set of pairs (m_h, σ_h), where $m_h = h$ and σ_h are signatures corresponding to m_h, for $h = 1, \ldots, i-1$. The adversary A, on inputs pk and Q, works as follows:

- First, A sets $pub = pk$;
- Then, A runs $\mathtt{ACT\text{-}STAT}_1$, which, taking as inputs the security parameter 1^τ and the maximum number of sessions ℓ, outputs the indexes i and j such that $1 \leq j < i \leq \ell$;
- Moreover, A constructs the sequence $hist_{i-1} = (a_1, a_2, \ldots, a_{i-1})$, by setting $a_h = (h, \sigma_h)$, for $h = 1, \ldots, i-1$. Notice that, for each $h = 1, \ldots, i-1$, the pair (h, σ_h) belongs to the set Q, which is an input to A;
- Afterwards, A runs $\mathtt{ACT\text{-}STAT}_2$ on inputs $(1^\tau, pub, i, j, hist_{i-1})$, in order to obtain the fake authentication messages $(a'_j, \ldots, a'_i)$;

- Finally, A outputs the pair (m, σ), where $m = i$ and $(i, \sigma) = a'_i$.

Since ACT-STAT is able to break the OM-UEA scheme $\Pi = (Gen, Auth, Ver)$ obtained by the DS-EAC, then $Ver(1^\tau, i, a'_i, pub, hist'_{i-1}) = 1$, where $hist'_{i-1} = (a_1, a_{j-1}, a'_j, \ldots, a'_{i-1})$. Thus, it also holds that $Vrfy(pk, i, a'_i) = 1$, i.e., the adversary A is able to build a forgery for the message i which does not belong to Q.

The view of adversary ACT-STAT, when executed as a subroutine by A, is the same as that in the impersonation experiment $\mathbf{Imp}_{\text{ACT-STAT},\Pi}(1^\tau)$. Thus, since the advantage of ACT-STAT in the experiment $\mathbf{Imp}_{\text{ACT-STAT},\Pi}(1^\tau)$ is non-negligible, the advantage of A in the signature-forgery experiment $\mathbf{Sig\text{-}Forge}_{A,\Sigma}(1^\tau)$ is also non-negligible. Contradiction. □

The DS-EAC construction, although simple, has the merit of showing that a weak form of unforgeability is sufficient to achieve security against active adversaries for OM-UEA schemes. Therefore, it is possible to instantiate it with existing concrete signature schemes to be chosen according to the characteristics of the application domain. In the next session, we show a lightweight instance suitable for applications in the IoT scenario.

A Lightweight Instance

OM-UEA schemes are particularly suitable to providing resistance to impersonation attacks in applications requiring continuous verification of user/device identities. This is of paramount importance, especially in the context of the Internet of Things (IoT), where continuous entity authentication is becoming essential. In such settings, given the limited computing power of the parties (especially the prover) and the often limited bandwidth channel, it is desirable to build one-message unilateral entity authentication schemes that require light computations.

In order to have a fast signing procedure and an optimal storage requirement on the signer's side, Yang et al. [2] proposed the LiS1 signature scheme, which is based on chameleon hash functions [26]. A chameleon hash function $\mathcal{CH}$ is a one-way hash function with trapdoor: if we do not know the associated trapdoor, it is hard to compute pre-images and collisions for the function. On the other hand, with knowledge of the trapdoor, collisions are efficiently computable. The main idea of the LiS1 scheme is to compute the signature verification key as a sequence of ℓ chameleon hash values corresponding to a set of dummy message/randomness pairs (m_i, r_i), for $1 \leq i \leq \ell$, where ℓ is the maximum number of messages that can be signed. In order to sign a message m'_i, which is different from each dummy message, the signer can easily compute a collision r'_i for the dummy pair (m_i, r_i), i.e., such that $\mathcal{CH}(m_i, r_i) = \mathcal{CH}(m'_i, r'_i)$. Such a value r'_i is then sent to the verifier, along with the message m'_i. The LiS1 scheme has been shown to be existentially unforgeable under a non-adaptive chosen message attack [2]; thus, when used in our DS-EAC, it gives rise to an OM-UEA scheme for ℓ-sessions, which is secure in the sense of ℓ-IMP-A-ST.

In order to reduce the size of the signing key in the LiS1 scheme, Yang et al. [2] proposed to fix all dummy messages m_i as a constant M, and to use a universal hash function $\mathcal{UHF}$ [27] to chain up all dummy randomness, i.e., each value r_i is computed as $r_i = \mathcal{UHF}(k, r_{i-1})$ for $1 \leq i \leq \ell$, where r_0 is chosen at random and k is the key of $\mathcal{UHF}$. With such a modification, the signer is only required to store the initial value r_0, along with the key k for the universal hash family, in addition to the message M and the trapdoor of the chameleon hash function. Moreover, in order to reduce the size of the signature verification key in the LiS1 schemes, Yang et al. also proposed using a Bloom filter [28] to efficiently represent the set of hash values stored in it. Such a data structure allows us to efficiently test whether an element belongs to a set but, due to its probabilistic nature, false positives are possible. With such a modification, the signature verification key in the LiS1 scheme only consists of the public key of the chameleon hash function along with the Bloom filter.

Yang et al. [2] proposed instantiating the chameleon hash function by using a slight variant of the original discrete logarithm-based chameleon hash function proposed in [26], whereas the universal hash function is instantiated with the multiply modular scheme

in [27]. The resulting scheme requires the signer to compute only two modular multiplications and three modular additions in order to sign a message. Moreover, the signing key in such a scheme is a bit string having a small size; indeed, it consists of five values chosen in Z_q^*.

4. A Construction Based on the Discrete-Logarithm Problem

In this section, we propose an OM-UEA scheme for ℓ-sessions whose security relies on the hardness of the discrete-logarithm problem. We refer to the proposed construction as the *Discrete-Logarithm Problem-based Entity Authentication Construction (DLP-EAC)*.

The Discrete-Logarithm Problem -based Entity Authentication Construction (DLP-EAC). Let p and q be two large primes such that q divides $p-1$; let G be a subgroup of Z_p^* having order q, and let g be a generator of G. We build an OM-UEA scheme $\Pi = (Gen, Auth, Ver)$ as follows:

- The parameter-generation algorithm Gen, on inputs the security parameter 1^τ and the maximum number of sessions ℓ, outputs the tuple $(pub, priv, s_0)$ constructed as follows:
  - Uniformly chooses at random $r_i \in Z_q$, for each $i = 1, \ldots, \ell$;
  - Computes $h_i = g^{r_i} \bmod p$, for each $i = 1, \ldots, \ell$;
  - Sets pub to the tuple $(h_1, h_2, \ldots, h_\ell)$;
  - Sets $priv$ to the tuple $(r_1, \ldots, r_\ell)$;
  - Sets the initial state s_0 to be empty.
- The authentication message-generation $Auth$, on inputs a session counter $1 \leq i \leq \ell$, the public information pub, the secret key $priv$ held by the prover, and the previous state s_{i-1}, outputs the authentication message $a_i = r_i$ and an empty state s_i.
- The verification algorithm Ver, on inputs a session counter i (where $1 \leq i \leq \ell$), an authentication message a_i, public information pub, and the sequence of authentication messages $hist_{i-1}$, which was generated by the previous $i-1$ authentication sessions, where $hist_0$ is the empty sequence, extracts the value h_i from pub and checks whether $h_i = g^{a_i} \bmod p$.

A pictorial representation of the DLP-EAC is shown in Figure 2.

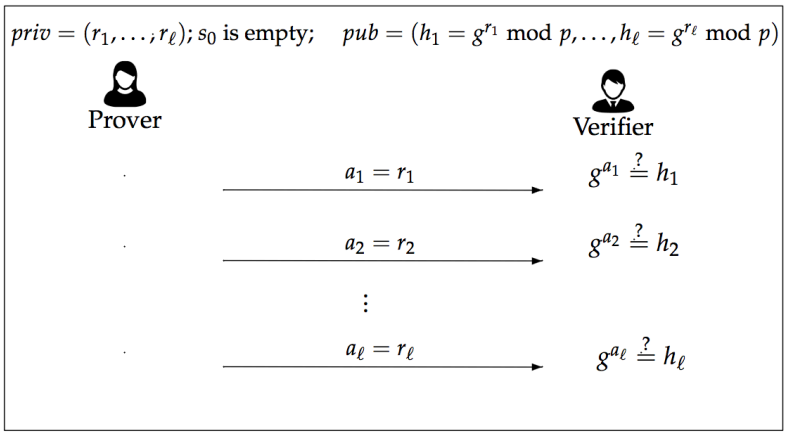

Figure 2. A Pictorial Representation of the DLP-EAC.

In the following, we prove that the DLP-EAC resists active static attacks.

Theorem 2. *Assuming the hardness of the discrete-logarithm problem, the OM-UEA scheme* Π *produced by the DLP-EAC is secure in the sense of* ℓ-IMP-A-ST.

Proof. Let us suppose, for the sake of contradiction, that the OM-UEA scheme $\Pi = (Gen, Auth, Ver)$ produced by DLP-EAC is not secure in the sense of ℓ-IMP-A-ST. There-

fore, there exists a polynomial time adversary $\texttt{ACT-STAT} = (\texttt{ACT-STAT}_1, \texttt{ACT-STAT}_2)$ whose advantage $\mathbf{Adv}^{\mathbf{Imp}}_{\texttt{ACT-STAT},\Pi}(1^\tau)$ in the experiment $\mathbf{Imp}_{\texttt{ACT-STAT},\Pi}(1^\tau)$ is non-negligible. We will prove that there exists a polynomial time adversary A that takes as input the tuple (G, q, g, h), where h is an element which has been uniformly chosen at random in G, and uses the adversary ACT-STAT to obtain a value $x \in Z_q$ such that $g^x \bmod p = h$.

The adversary A runs $\texttt{ACT-STAT}_1$, which, taking as inputs the security parameter 1^τ and the maximum number of sessions ℓ, outputs the indexes i and j such that $1 \leq j < i \leq \ell$. Afterwards, A first chooses at random a value $r_t \in Z_q$ for each $t = 1, \ldots, i-1$ and each $t = i+1, \ldots, \ell$ and then sets pub to be equal to the tuple $(h_1, \ldots, h_\ell) = (g^{r_1} \bmod p, \ldots, g^{r_{i-1}} \bmod p, h, g^{r_{i+1}} \bmod p, \ldots, g^{r_\ell} \bmod p)$ and $hist_{i-1}$ to be equal to the tuple $(r_1, \ldots, r_{i-1})$. On inputting 1^τ, pub, i, j, and $hist_{i-1}$, $\texttt{ACT-STAT}_2$ eventually outputs the tuple $(a'_j, \ldots, a'_i)$ of forgeries and the adversary A outputs a'_i. Since ACT-STAT is able to break the OM-UEA scheme generated by the DLP-EAC, then $Ver(1^\tau, i, a'_i, pub, hist_{i-1}) = 1$. Thus, it also holds that $g^{a'_i} \bmod p = h$, i.e., the adversary A is able to compute the discrete logarithm of h with respect to g.

It is easy to see that the view of adversary ACT-STAT, when executed as a subroutine by A, is the same as in the impersonation experiment $\mathbf{Imp}_{\texttt{ACT-STAT},\Pi}(1^\tau)$. As the advantage of adversary ACT-STAT in the impersonation experiment $\mathbf{Imp}_{\texttt{ACT-STAT},\Pi}(1^\tau)$ is non-negligible, it can be concluded that the advantage of A in the discrete-logarithm experiment $\mathbf{DLog}_{A,\mathcal{G}}(1^\tau)$ is non-negligible. □

Performance Analysis

In this section, we analyze the DLP-EAC with respect to memory and computational requirements.

It is easy to see that the private information for the prover consists of $\ell \cdot \log q$ bits, since $priv$ contains ℓ values which are randomly chosen in Z_q. In order to decrease the size of the private key in the DLP-EAC, we could use the same idea behind the LiS1 scheme: we chose at random an initial value r_0 in Z_q and then compute all values $r_1, \ldots, r_\ell$ by using a universal hash function $\mathcal{UHF}$ with a random key k, i.e., $r_i = \mathcal{UHF}(k, r_{i-1})$ for $1 \leq i \leq \ell$. With such a modification, the signer is only required to store the initial value r_0, along with the key k for the universal hash function. In particular, if we instantiate the universal hash function with the multiply modular scheme in [27], the size of the private key for the signer is equal to $3 \cdot \log q$ bits, since the key k consists of two values k_0 and k_1 chosen in Z_q. Moreover, to compute each $r_i = \mathcal{UHF}(k, r_{i-1})$, where $r_i = k_0 \cdot r_{i-1} + k_1 (\mathrm{mod} q)$ for $1 \leq i \leq \ell$, the prover only needs to perform one modular addition and one modular multiplication. As regards as the public information, its size equals $\ell \cdot \log p$ bits, since pub contains ℓ values in Z_p^*. As performed in the LiS1 scheme, we could use a Bloom filter [28] to efficiently represent the set of values in the public key, in order to reduce its size. In particular, if we use the Bloom filter construction proposed by Pagh et al. [29] to represent the set of ℓ values $g^{r_i} \bmod p$, for $i = 1, \ldots, \ell$, the size of the public key reduces to $1.44\epsilon\ell$ bits, where ϵ denotes the probability that a false positive occurs while testing the membership of an element to the Bloom filter.

5. Conclusions

In this paper, we have discussed OM-UEA schemes where the prover and verifier do not have any pre-shared secret, such as a password. In particular, we have shown that OM-UEA schemes can be implemented using digital signatures and that a weak form of unforgeability is sufficient to achieve security against active adversaries.

The efficiency of our construction is based on the fact that each authentication session requires only one message to be sent from the prover to the verifier; moreover, when the digital signature scheme is appropriately chosen, our construction provides a low computational effort for the prover and thus can be used to support continuous entity authentication of IoT devices. Indeed, in the proposed construction, the prover only

needs to perform two modular multiplications and three modular additions for each authentication message.

Afterwards, we have considered the problem of further reducing the computational complexity for the prover. In particular, we have proposed a different construction for OM-UEA schemes, whose security relies on the hardness of the discrete-logarithm problem. In such a construction, the prover only needs to perform one modular multiplication and one modular addition for each authentication message.

Author Contributions: Writing—original draft, A.D.S., A.L.F., M.F. and B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by EU-NGEU under the NRRP MUR program, projects "SEcurity and RIghts in the CyberSpace (SERICS)" (PE00000014) and "VITALITY Ecosystem, Spoke 1 MEGHALITIC".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. De Santis, A.; Ferrara, A.L.; Flores, M.; Masucci, B. Provably-Secure One-Message Unilateral Entity Authentication Schemes. *IEEE Trans. Dependable Secur. Comput.* **2022**, *submitted (under the 2nd round of review)*.
2. Yang, Z.; Jin, C.; Tian, Y.; Lai, J.; Zhou, J. LiS: Lightweight Signature Schemes for Continuous Message Authentication in Cyber-Physical Systems. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security—ASIACCS 2020, Taipei, Taiwan, 5–9 October 2020.
3. Heyszl, J.; Stumpf, F. Efficient One-Pass Entity Authentication based on ECC for Constrained Devices. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 13–14 June 2010.
4. Yavuz, A.A. An Efficient Real-Time Broadcast Authentication Scheme for Command and Control Messages. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 1733–1742. [CrossRef]
5. Bamasag, O.O.; Youcef-Toumi, K. Towards Continuous Authentication in Internet of Things Based on Secret Sharing Scheme. In Proceedings of the Workshop on Embedded Systems Security—WESS, Amsterdam, The Netherlands, 8 October 2015.
6. Hernandez-Ramos, I.L.; Pawlowski, M.P.; Jara, A.J.; Skarmeta, A.F.; Ladid, L. Towards a Lightweight and Authorization Framework for Smart Objects. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 690–702. [CrossRef]
7. Chen, D.; Zhang, N.; Qin, Z.; Mao, X.; Qin, Z.; Shen, X.; Li, X.Y. S2M: A Lightweight Acoustic Fingerprints-based Wireless Device Authentication Protocol. *IEEE Internet Things J.* **2016**, *4*, 88–100. [CrossRef]
8. Shahzad, M.; Singh, M.P. Continuous Authentication and Authorization for the Internet of Things. *IEEE Internet Comput.* **2017**, *21*, 86–90. [CrossRef]
9. Nespoli, P.; Zago, M.; Celdran, A.H.; Perez, M.G.; Marmol, F.G.; Garcia Clernente, F.J. A Dynamic Continuous Authentication Framework in IoT-Enabled Environments. In Proceedings of the 2018 Fifth International Conference on Internet of Things: Systems, Management and Security, Valencia, Spain, 15–18 October 2018.
10. Chuang, Y.H.; Lo, N.W.; Yang, C.Y.; Tang, S.W. A Lightweight Continuous Authentication Protocol for the Internet of Things. *Sensors* **2018**, *18*, 1104. [CrossRef] [PubMed]
11. Sathyadevan, C.; Achuthan, K.; Doss, R.; Pan, L. Protean Authentication Scheme—A Time-bound Dynamic Keygen Authentication Technique for IoT Edge Nodes in Outdoor Deployments. *IEEE Access* **2019**, *7*, 92419–92435. [CrossRef]
12. Shah, S.W.; Syed, N.F.; Shaghaghi, A.; Anwar, A.; Baig, Z.; Doss, R. Towards a Lightweight Continuous Authentication Protocol for Device-to-Device Communication. In Proceedings of the IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021.
13. Badhib, A.; Alshehri, S.; Cherif, A. A Robust Device-to-Device Continuous Authentication Protocol for the Internet of Things. *IEEE Access* **2021**, *9*, 124768–124792. [CrossRef]
14. Fayad, A.; Hammi, B.; Khatoun, R.; Serhrouchni, A. A Blockchain-based Lightweight Authentication Solution for IoT. In Proceedings of the 2019 3rd Cyber Security in Networking Conference (CSNet), Quito, Ecuador, 23–25 October 2019; pp. 28–34.
15. Biswal, A.K.; Maiti, P.; Bebarta, S.; Sahoo, B.; Turuk, A.K. Authenticating IoT Devices with Blockchain. In *Advanced Applications of Blockchain Technology*; Springer: Singapore, 2020.
16. Lau, C.H.; Alan, K.H.Y.; Yan, F. Blockchain-Based Authentication in IoT Networks. In Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 10–13 December 2018; pp. 1–8.
17. Hussain Al-Naji, F.; Zagrouba, R. A Survey on Continuous Authentication Methods in Internet of Things Environment. *Comput. Commun.* **2020**, *163*, 109–133. [CrossRef]
18. Shimon, E.; Goldreich, O.; Micali, S. On-Line/Off-Line Digital Signatures. *J. Cryptol.* **1996**, *9*, 35–67.
19. Shamir, A.; Tauman, Y. Improved Online/Offline Signature Schemes. In Proceedings of the 21st Annual International Cryptology Conference—Crypto 2001, Santa Barbara, CA, USA, 19–23 August 2001.
20. Yao, A.C.; Zhao, Y. Online/Offline Signatures for Low-Power Devices. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 283–294. [CrossRef]

21. Yavuz, A.A.; Ozmen, M.O. Ultra Lightweight Multiple-Time Digital Signature for the Internet of Things Devices. *IEEE Trans. Serv. Comput.* **2019**, *15*, 215–227. [CrossRef]
22. Lamport, L. *Constructing Digital Signatures from a One-Way Function* ; Technical Report CSL-98; Computer Science Laboratory SRI International: Menlo Park, CA, USA, 1979.
23. Reyzin, L.; Reyzin, N. Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying. In Proceedings of the 7th Australian Conference on Information Security and Privacy—ACISP'02, Melbourne, Australia, 3–5 July 2002; pp. 144–153.
24. ISO/IEC 11889-1:2009. Trusted Platform Module. Available online: https://www.iso.org/standard/50970.html (accessed on 1 May 2009).
25. Goldwasser, S.; Micali, S.; Rivest, R. A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. *SIAM J. Comput.* **1988**, *17*, 281–308. [CrossRef]
26. Krawczyk, H.; Rabin, T. Chameleon Signatures. In Proceedings of the Network and Distributed System Security Symposium—NDSS 2000, Diego, CA, USA, 2–4 February 2000.
27. Carter, L.; Wegman, M.N. Universal Classes of Hash Functions. In Proceedings of the ACM Symposium on Theory of Computing—STOC 1977, Boulder, CO, USA, 2–4 May 1977; pp. 106–112.
28. Bloom, B.H. Space/Time Trade-offa in Hash Coding with Allowable Errors. *Commun. ACM* **1970**, *13*, 422–426. [CrossRef]
29. Pagh, A.; Pagh, R.; Srinivasa Rao, S. An Optimal Bloom Filter with Replacement. In Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms—SODA 2005, Vancouver, BC, Canada, 23–25 January 2005; pp. 825–829.