Certificateless Multi-Party Authenticated Encryption Mitigating Ephemeral Key Leakage

Łukasz Krzywiecki
Department of Fundamentals of Computer Science
Wrocław University of Science and Technology
Wrocław, Poland
lukasz.krzywiecki@pwr.edu.pl

Hannes Salin
Department of Information and Communication Technology
Swedish Transport Administration
Borlänge, Sweden
hannes.salin@trafikverket.se

Mateusz Jachniak
Department of Fundamentals of Computer Science
Wrocław University of Science and Technology
Wrocław, Poland
236738@student.pwr.edu.pl

2021 IEEE 20th International Symposium on Network Computing and Applications (NCA) | 978-1-6654-9550-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/NCA53618.2021.9685848

***Abstract*—We propose two secure modifications of a multi-party authenticated encryption scheme with aggregation, mitigating ephemeral leakage attacks on narrowband Internet of Things devices and sensor equipment, used in different type of connected infrastructures. Our schemes are provably secure in a stronger security model where a set of nodes in a 5G-based architecture can produce authenticated and encrypted signcryption messages, aggregated into one verifiable cipher text. We provide benchmarks from a proof of concept implementation, showing the feasibility of our solutions.**

***Index Terms*—signcryption, cryptography, NB-IoT, multi-party authentication, 5G, certificateless authenticated encryption.**

I. Introduction

The rapid development of the Internet of Things (IoT) has led to many standardization attempts which are not necessarily compliant with each other. This is one of the main issues in the mobile device industry. As a result, the Narrowband Internet of Things (NB-IoT) standard was proposed [1], [2] for unifying systems. The standard is developed by 3rd Generation Partnership Project (3GPP) and the main benefits are that it enables efficient operations of cellular services, in particular designed for IoT devices requiring low cost, long battery life, increased coverage (good for building interiors, basements) and high connection density (thousands of devices in a small area). With the growing field of IoT-based infrastructures, the need for device authentication and data integrity is therefore acute. Regardless of architecture, any connected devices in the Cloud or internal secure networks, must provide these security requirements as a minimum.

A. Connected Infrastructure 5G Technology

Today's road and railway infrastructures are heavily dependent on connected devices such as IoT, Industrial IoT (IIoT), sensory devices, cameras and controlling units. In remote areas, both low powered long-range communication and efficient, local IoT cluster networking is needed. Also, with the growing advancement of Intelligent Transportation Systems (ITS) technology, more types of devices can be connected thus enabling more use cases in smart and secure transportation domains. In particular, the 5G network is of high importance; as a telecommunication technology standard, 5G combines efficiently with NB-IoT as the implementation of the communication protocol. The European Rail Traffic Management System (ERTMS) which currently is in the deployment phase in several European countries aims to build their communication stack on 5G. ERTMS is an initiative for the interoperability of railways, including harmonization of telecommunications for railway operations, improving the European infrastructure. Workplans from the European Commission states that one of the crucial game changers for deploying ERTMS and future digitalization for higher capacity and better performance is the Future Radio Mobile Communication System (FRMCS), replacing GSM-R and introducing 5G technologies [3]. Considering large-scale deployments of such new technology in infrastructure, cyber security is fundamental for safety and sustainability reasons. Vulnerable protocols could have a disastrous impact, potentially leading to casualties due to faulty or compromised controlling systems [4], [5]. Therefore, the need for efficient and secure protocols to deploy over the 5G protocol, e.g. the application layer, is high. For future advances in both ITS and connected railway infrastructures, such protocols are then an important area of research [6], [7].

978-1-6654-9550-9/21/$31.00 ©2021 IEEE

B. Ephemeral Key Leakage and Hardware Security Modules

In many protocols, an *ephemeral key* is used for a specific session and then discarded, thus mitigating replay attacks. When generating an ephemeral key it is highly important to use a cryptographically secure source of randomness. In practice, using pseudo-random generators, the risk of generating predictable bits is a real threat, thus allowing an attacker to access the key. For IoT and mobile devices, it is possible to have *Hardware Security Modules* (HSM) which are specifically manufactured hardware components, designated for secure storage of private keys and secure computations within the device. A *minimal function* $\mathbf{f}$ is such cryptographic function that can be executed within the HSM, e.g. encryptions or signatures using the secret key stored inside the HSM.

C. Problem Statement and Related Work

Many protocols have focused on the security in LTE networks and the 5G architecture, e.g. the lack of proper identity protection allowing DDoS attacks [8]. Authenticated encryption is a way to both encrypt a message and at the same time provide authenticity. A typical pattern is the *encrypt-then-MAC*, where a message authentication code is computed over the encrypted values sent to a decryptor. Another technique is *signcryption* where the produced cipher text and authentication value is computed in one logical step instead of two. Bit leakage was first addressed by Chari et al. [9], continued by Goubin and Patarin [10] and Alwen et al. [11]. Also, the problem of bit leakage from cryptographic keys was analyzed by Canetti et al. [12]. The problem with ephemeral key leakage in untrusted devices has been analyzed further, e.g. in [13]. For NB-IoT usage in a 5G architecture, a multi-party authenticated encryption algorithm with improved data transmission efficiency has been proposed [14]. However, despite the use of symmetric cryptography, data security is still compromised by the leakage of ephemeral values [15], [16]. In our modified scheme, we will show how to mitigate this problem. It is not difficult to imagine that these devices could potentially become the target of side-channel attacks, malicious hardware manufacturers that pre-install data leakage functionality, or other types of adversarial tampering. To ensure the integrity of the data flows, the protocols used by these devices should be as secure as possible.

D. Contribution

We present a multi-party authentication encryption scheme, addressing the security and privacy issues for NB-IoT connections over a 5G architecture. Compared to previous work, our scheme not only provides identity, anonymity and non-repudiation, but is also resistant to the leakage of ephemeral values. All previous properties are preserved, such as device validation verification, data breach checking and verification of generated cipher texts. The contribution of this paper is the following:

- We introduce a new security model for multi-party authentication, in which ephemeral keys can be exposed or set by an attacker.
- We show that a typical scheme, as proposed in [14], is not secure in the proposed security model.
- We propose two versions of the improved multi-party authentication scheme, asynchronous and synchronous, which are immune to ephemeral key leakage.
- We provide security proofs for our schemes in the proposed stronger security model.
- We provide comparison benchmarks, based on our proof-of-concept implementations.

E. Organization

The rest of the paper is organized as follows: in Sec. II we present the system settings and security requirements. In Sec. III, the formal model is presented and in Sec. IV we prove the original scheme unsecure in the new stronger model. In Sec. V the security analysis of the proposed schemes are described, and In Sec. VI we present our results from a proof of concept implementation. Sec. VII summarizes the paper.

II. System Settings and Security Requirements

A. Simplified 5G System Architecture

The simplified 5G architecture, depicted on Fig. 1, shows a subset of components involved in the analyzed Multi-Party Authentication Encryption (MPAE) for IoT devices (IOTD). IOTD devices in the *Endpoints* layer run *Core Network* functionalities via *Access Network* components layer. The Access and Mobility Management Function (AMF) handles key agreement and authentication requests, implements integrity protection algorithms, receives all connection and session related information from the user equipment and passes the session management requirements to the Session Management Function (SMF). The User Plane Function (UPF) is responsible for user-side configurations, allowing the data transfer component to be decentralized, connects mobile infrastructure and data network, and provides packet routing/forwarding.

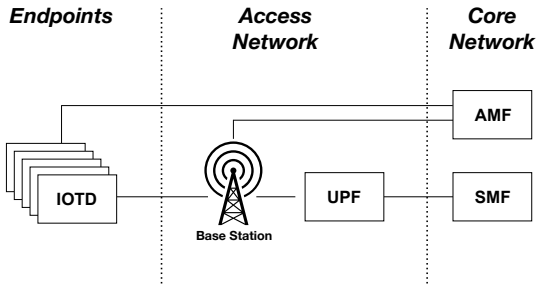

Fig. 1: Simplified 5G setup for IOTD.

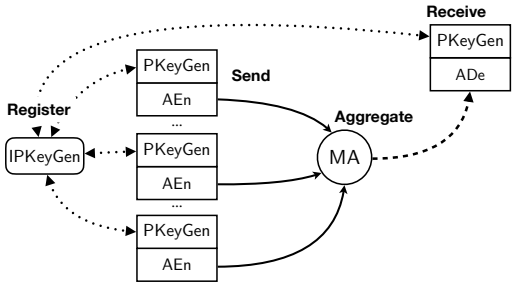

Fig. 2: The functionality of the MPAE scheme with IoT devices, a register node, an aggregation node and a receiver node.

B. Scheme Functionality and Security Requirements

The MPAE is defined on top of the architecture presented in Fig. 1. For a connected infrastructure where clusters of sensors need to cooperate, e.g. for a limited range of railway maintenance (cameras, radar, heat sensors etc.), each IOTD needs to send the sensor data to a track-side unit (dedicated collector for IOTD) for further processing. At this stage, all data need to be sent with intact integrity and privacy. These

IOTD devices have two different types of secret keys. The user key usk generated locally by a procedure PKeyGen, and psk obtained during the registration, where the key generation centre runs the procedure IPKeyGen. A registered device can produce signcrypted messages with usk and psk, together with a session-specific ephemeral key (randomness) and the recipients public key, running procedure AEn, as depicted in Figure 2. The aggregation of authentication values (via the MA procedure) can be implemented as a specific UPF or AMF sub-function in access or core layers, or as a dedicated function in one of the endpoint devices. The security of authentication requires that both secret keys are used, and the lack of at least one key prevents the adversarial forgery. This implies that secret keys are stored securely in separate HSMs, preferably from different vendors. Typically this should immune against forgery, even if one of the vendors is malicious, or an HSM is compromised due to production errors. However, this methodology assumes that the randomness is thoroughly protected since the security is broken if the adversary controls any randomness used in the protocol. Thus we also require the scheme to be *resistant to ephemeral key leakage*, i.e. the adversary should not impersonate the legitimate device even if it controls the randomness.

C. Device architecture

We consider architecture based upon two distinct HSMs from different vendors, for each secret key. The scheme should be still secure even one of the two HSMs is compromised, and additionally if the ephemeral value in less restricted code area is controlled by the attacker. The HSMs should validate inputs from unsafe areas in the device due to attacks on invalid curve points. The device architecture for our proposed schemes is depicted in Fig 3. HSM1, HSM2, realizing minimal functionalities **f1**, **f2** of exponent with secret keys. Note, the scheme construction from [14] is inherently vulnerable to ephemeral leakage.

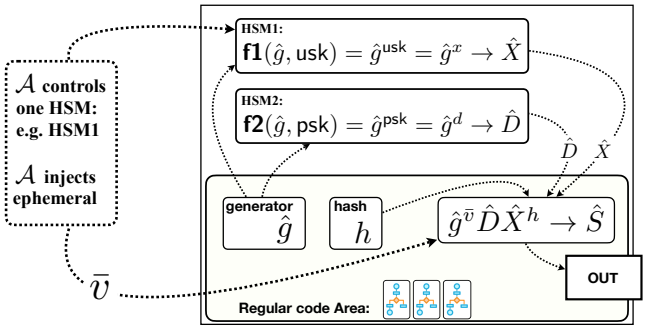

Fig. 3: Device architecture for AEn procedure.

D. Threat Description

The proposed authenticated encryption for IoT devices is based on a multi-party model. For instance, during the compromised execution of the message-relaying protocol, an IoT device has to encrypt a message m. The IoT device generates its secret key usk before registration. It also obtains a second key psk after registration. To perform the functionality described above, an ephemeral key v is used by the IoT device when authenticating a message. Our stronger model considers the following attacks:

- *Attack 1:* In the first attack scenario the adversary gets hold of usk, stored within the device's HSM as described in Section I-B, and injects the ephemeral session key $\bar{v}$. Given these values, the adversary tries to impersonate the device without the missing psk.
- *Attack 2:* A second *dual* attack is also considered. Similarly to the previous scenario, the adversary can get psk and controls $\bar{v}$. Given these values, the adversary tries to impersonate the device and authenticate a fresh message without the missing usk.

Note that a typical construction like [14], do not consider the leakage of ephemeral values v at all. Knowledge of v allows the adversaries to break the system without the additional secret keys. Our proposed constructions are immune against the considered attacks.

III. Formal model

We use the following notation: Let λ be a security parameter, $\epsilon(\lambda)$ denotes a negligible function of the security parameter. $\mathcal{H} : \{0,1\}^* \to \mathbb{A}$ denotes a secure hash function from binary strings into a set $\mathbb{A}$. $a_1, \ldots, a_n \leftarrow_\$ \mathbb{A}$ means that each a_i is chosen uniformly at random from $\mathbb{A}$. Let $\langle g \rangle = \mathbb{G} \leftarrow_\$ \mathcal{G}(1^\lambda)$ denotes that a group $\mathbb{G}$ with the generator g is constructed by the randomized group setup algorithm $\mathcal{G}$ taking the parameter λ. In the rest of the paper we refer to multiplicative groups $\langle g \rangle = \mathbb{G}$ of prime order q. Our scheme is based on the typical hardness assumptions for probabilistic polynomial time (PPT) algorithms for which the typical assumptions hold. Let $x \leftarrow_\$ \mathbb{Z}_q^*, y \leftarrow_\$ \mathbb{Z}_q^*, z \leftarrow_\$ \mathbb{Z}_q^*$, then:

Definition 1 (Discrete Logarithm Problem (DLP))**.** *For any PPT algorithm $\mathcal{A}_{\mathsf{DLP}}$ in $\langle g \rangle = \mathbb{G}$:*

$$\Pr[\mathcal{A}_{\mathsf{DLP}}(g^x) = x] \leq \epsilon_{\mathsf{DLP}}(\lambda),$$

where $\epsilon_{\mathsf{DLP}}(\lambda)$ is negligible.

Definition 2 (Computational Diffie-Hellman (CDH))**.** *For any PPT algorithm $\mathcal{A}_{\mathsf{CDH}}$ in $\langle g \rangle = \mathbb{G}$:*

$$\Pr[\mathcal{A}_{\mathsf{CDH}}(g^x, g^y) = g^{xy}] \leq \epsilon_{\mathsf{CDH}}(\lambda),$$

where $\epsilon_{\mathsf{CDH}}(\lambda)$ is negligible.

Definition 3 (Decisional Diffie-Hellman Oracle ($\mathcal{O}_{\mathsf{DDH}}$))**.** *The decisional Diffie-Hellman oracle ($\mathcal{O}_{\mathsf{DDH}}$) is an algorithm in $\langle g \rangle = \mathbb{G}$ for which:*

$$|\Pr[\mathcal{O}_{\mathsf{DDH}}(g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{O}_{\mathsf{DDH}}(g^x, g^y, g^z) = 1]|$$

is non-negligible.

Definition 4 (Gap Computational Diffie-Hellman (GDH))**.** *For any PPT algorithm $\mathcal{A}_{\mathsf{GDH}}$ having access to the oracle $\mathcal{O}_{\mathsf{GDH}}$:*

$$\Pr[\mathcal{A}_{\mathsf{CDH}}^{\mathcal{O}_{\mathsf{DDH}}}(g^x, g^y) = g^{xy}] \leq \epsilon_{\mathsf{DDH}}(\lambda),$$

where $\epsilon_{\mathsf{DDH}}(\lambda)$ is negligible.

Definition 5. *Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be groups with generators g_1, g_2, g_T respectively. Let $q = |\langle g_1 \rangle| = |\langle g_2 \rangle| = |\langle g_T \rangle|$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to G_T$ be a bilinear map with the following properties:*

1) *Bilinearity $\forall(a, b \in \mathbb{Z}_q, g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2)$: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$*
2) *Computability Computing $\hat{e}$ is efficient*
3) *Non-degeneracy $\exists(g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2) : \hat{e}(g_1, g_2) \neq 1$*

We then call $\hat{e}$ a pairing. *When $\mathbb{G}_1 = \mathbb{G}_2$ the pairing is called symmetric and we simply denote the group by $\mathbb{G}$.*

A. Multi-Party Authenticated Encryption

The original scheme [14] involves interaction between three types of parties: the Key Generate Center (KGC), NB-IoT devices (IOTD) and AMF. NB-IoT devices (IOTD) are indexed. The subset of indexes of IOTD devices that sends data is denoted as $\mathtt{ID} = \{i\}$, while the index of the receiver device is denoted by k. The KGC uses the Setup functions to generate basic parameters necessary for setting up a 5G communication channel between IOTD and AMF. It is also a trusted party of the protocol. AMF is the receiver party that verifies incoming data and decrypts the messages.

Definition 6. *A certificateless Multi-Party Authentication Encryption scheme* (MPAE) *consists of seven algorithms:* Setup, UKeyGen, IPKeyGen, PKeyGen, AEn, MA, *and* ADe.

$\mathsf{Setup}\left(1^{\lambda}\right) \to (\mathsf{par}, \mathsf{msk})$: *This algorithm is run by the* KGC. *Upon receiving a security parameter λ, it returns system public parameters* par *and a master secret key* msk. *The* par *are default parameters to all procedures of the scheme, thus we skip their explicit invocation.*

$\mathsf{UKeyGen}(i) \to (\mathsf{upk}_i, \mathsf{usk}_i)$: *The user-side key generation algorithm is run by the user itself. For a user (i.e.* IOTD, AMF*) with identity i this algorithm generates a user side public key upk_i and a user-side secret key usk_i.*

$\mathsf{IPKeyGen}(\mathsf{msk}, i, \mathsf{upk}_i) \to (\mathsf{ippk}_i,\ \mathsf{ipsk}_i)$ *: This algorithm is run by the* KGC *to generate initial-partial keys for a user with identity i. On input* msk, i *and* upk_i, *the algorithm returns an initial-partial public key ippk_i and the corresponding initial-partial secret key ipsk_i.*

$\mathsf{PKeyGen}(\mathsf{upk}_i, \mathsf{ippk}_i, \mathsf{ipsk}_i) \to (\mathsf{ppk}_i, \mathsf{psk}_i)$: *This algorithm is run by the user to transform initial partial keys to partial keys. On input $\mathsf{upk}_i, \mathsf{ippk}_i$ and ipsk_i, the algorithm returns a partial public key and the corresponding partial secret key.*

$\mathsf{AEn}(\mathsf{PK}_i, \mathsf{SK}_i, \mathsf{PK}_k, m_i) \to \mathtt{ACT}_i$: *The authenticated aggregate encryption algorithm is performed by $\{\mathtt{IOTD}_i\}_{i\in\mathtt{ID}}$. Suppose $\mathtt{IOTD}_i$ takes as inputs* par, *its public key $\mathsf{PK}_i = (\mathsf{upk}_i, \mathsf{ppk}_i)$ and corresponding secret key $\mathsf{SK}_i = (\mathsf{usk}_i, \mathsf{psk}_i)$, the public key PK_k of a target and a message m_i, the algorithm generates an authenticated encryption ciphertext $\mathtt{ACT}_i$*

$\mathsf{MA}(\{\mathtt{ACT}_i\}_{i\in\mathtt{ID}}) \to \mathtt{ACT}$: *The multi-authenticated encryption algorithm is performed by* AMF. *It inputs a set of authenticated encryption ciphertexts $\{\mathtt{ACT}_i\}_{i\in\mathtt{ID}}$, and returns a multi-party aggregate authenticated encryption ciphertext* ACT.

$\mathsf{ADe}(\mathsf{SK}_k, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathtt{ACT}) \to m$ *or* $\perp$: *The multi-party aggregate authenticated decryption algorithm is run by* AMF. *It takes as inputs $\mathsf{SK}_k, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}$ and* ACT, *then outputs $m = \{m_i\}_{i\in\mathtt{ID}}$, if it was a valid aggregated multi-party authenticated encryption ciphertext, otherwise it outputs $\perp$.*

We require the scheme to be *correct*, i.e. messages authenticated via secret keys (of senders) and encrypted via a public key (of a recipient) should be positively verified and successfully decrypted.

Definition 7 (Correctness)**.** MPAE *scheme is correct if for any set of indexes $\Omega \subset \mathbb{Z}^*$, any subset $\mathtt{ID} \subset \Omega$, any index of a recipient $k \in \Omega$, and any messages $\{m_i\}_{i\in\mathtt{ID}} \subset \{0,1\}^l$:*

$$\Pr\left[\begin{array}{l}(\mathsf{par}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda),\\ \textit{For each } i \in \Omega:\\ \quad (\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKeyGen}(i),\\ \quad (\mathsf{ippk}_i, \mathsf{ipsk}_i) \leftarrow \mathsf{IPKeyGen}(\mathsf{msk}, i, \mathsf{upk}_i),\\ \quad (\mathsf{SK}_i, \mathsf{PK}_i) \leftarrow \mathsf{PKeyGen}(\mathsf{upk}_i, \mathsf{ippk}_i, \mathsf{ipsk}_i),\\ \textit{For each } i \in \mathtt{ID}:\\ \quad \mathtt{ACT}_i \leftarrow \mathsf{AEn}(\mathsf{PK}_i, \mathsf{SK}_i, \mathsf{PK}_k, m_i),\\ \mathtt{ACT} \leftarrow \mathsf{MA}(\{\mathtt{ACT}_i\}_{i\in\mathtt{ID}}),\\ \{m_i\}_{i\in\mathtt{ID}} \leftarrow \mathsf{ADe}(\mathsf{SK}_k, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathtt{ACT})\end{array}\right] = 1.$$

The definition below models the secrecy of messages encrypted to the recipient whose private decryption key is unknown to the adversary. We do not tweak that definition, and recall it just for the paper completeness.

Definition 8 (Indistinguishability under adaptive chosen ciphertext attack)**.** *Let* $\mathsf{MPAE} = (\mathsf{Setup}, \mathsf{UKeyGen}, \mathsf{IPKeyGen}, \mathsf{PKeyGen}, \mathsf{AEn}, \mathsf{MA}, \mathsf{ADe})$ *be a certificateless Multi-Party Authentication Encryption scheme. The following security experiment $\mathtt{Exp}^{\lambda,\ell_1,\ell_2}_{\mathsf{ESS}}(\mathcal{A}, \mathsf{MPAE})$ is defined as:*

`Init stage` : *The challenger generates common parameters* $\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, *including the device indexes Ω. For each $i \in \Omega$ it generates keys:*

$$\begin{array}{l}(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKeyGen}(\mathsf{par}, i),\\ (\mathsf{ippk}_i, \mathsf{ipsk}_i) \leftarrow \mathsf{IPKeyGen}(\mathsf{msk}, i, \mathsf{upk}_i),\\ (\mathsf{SK}_i, \mathsf{PK}_i) \leftarrow \mathsf{PKeyGen}(\mathsf{upk}_i, \mathsf{ippk}_i, \mathsf{ipsk}_i).\end{array}$$

$\mathcal{A}$ chooses a set of indexes of sending devices $\mathtt{ID} \subset \Omega$ and the receiving device index $k \in \Omega, k \notin \mathtt{ID}$. $\mathcal{A}$ is given all generated data, except the master secret msk *and secret keys of the receiving device k:* ipsk_k, $\mathsf{SK}_k = (\mathsf{usk}_k, \mathsf{psk}_k)$.

`Oracle AEn` : *The oracle $\mathcal{O}_{\mathsf{AEn}}$ accepts parameters* par, *public key PK_i, public key of the recipient, PK_k, and a message m_i, and outputs $\mathtt{ACT}_i = (s_i, V_i, C_i)$ which is verifiable and decryptable. Note that having the secret keys of all sending devices $\mathcal{A}$ can produce $\mathtt{ACT}_i = (s_i, V_i, C_i)$ itself.*

`Oracle ADe` : *The oracle $\mathcal{O}_{\mathsf{ADe}}$ accepts parameters* par, *index of the recipient k, the public keys $\{\mathsf{PK}_i\}_{i\in\mathtt{ID}}$*

Original Scheme MPAE [14]	**Modified Schemes**: MPAE-1 and MPAE-2
$\mathsf{Setup}(\lambda)$: $(\mathbb{G}, g, q) \leftarrow \mathsf{Setup}(1^\lambda)$ $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q^*$, $\mathcal{H}_l : \{0,1\}^* \to \{0,1\}^l$ $a \leftarrow_\$ \mathbb{Z}_q^*$, $A = g^a$ $\mathsf{par} = (\mathbb{G}, g, q, A, \mathcal{H})$ **return** $\mathsf{par}, \mathsf{msk} = a$	$\mathsf{Setup}(\lambda)$: $(\mathbb{G}, \mathbb{G}_T, g, g_T, q, \hat{e}) \leftarrow \mathsf{Setup}(1^\lambda)$ $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q^*$, $\mathcal{H}_l : \{0,1\}^* \to \{0,1\}^l$, $\mathcal{H}_g : \{0,1\}^* \to \mathbb{G}$ $a \leftarrow_\$ \mathbb{Z}_q^*$, $A = g^a$ $\mathsf{par} = (\mathbb{G}, \mathbb{G}_T, g, g_T, q, A, \hat{e}, \mathcal{H}, \mathcal{H}_g)$ **return** $\mathsf{par}, \mathsf{msk} = a$
$\mathsf{UKeyGen}(\mathsf{par}, i)$: $x_i \leftarrow_\$ \mathbb{Z}_q^*$, store x_i in HSM1 of i, compute $X_i = \mathbf{f1}(g) = g^{x_i}$ **return** $\mathsf{upk}_i = X_i, \mathsf{usk}_i = x_i$	
$\mathsf{IPKeyGen}(\mathsf{par}, a, i, \mathsf{upk}_i)$: $r_i \leftarrow_\$ \mathbb{Z}_q^*, R_i = g^{r_i}$, $X_i \leftarrow \mathsf{upk}_i$, $\mathsf{n}_i \leftarrow_\$ \{0,1\}^l$, $h_{1,i} = \mathcal{H}(\mathsf{n}_i, X_i, R_i)$, $u_i = r_i + a \cdot h_{1,i} + \mathcal{H}((X_i)^a)$ **return** $\mathsf{ippk}_i = (\mathsf{n}_i, R_i), \mathsf{ipsk}_i = u_i$	
$\mathsf{PKeyGen}(\mathsf{par}, \mathsf{upk}_i, \mathsf{ippk}_i, \mathsf{ipsk}_i)$: $u_i \leftarrow \mathsf{ipsk}_i$, $d_i = u_i - \mathcal{H}(\mathbf{f1}_i(A)) = u_i - \mathcal{H}(A^{x_i})$, $\mathsf{ppk}_i = (\mathsf{n}_i, R_i), \mathsf{psk}_i = d_i$ **return** $\mathsf{SK}_i = (\mathsf{usk}_i, \mathsf{psk}_i)$, $\mathsf{PK}_i = (\mathsf{upk}_i, \mathsf{ppk}_i)$	

TABLE I: The modified schemes are on the right.

Original Scheme MPAE [14]	**Modified Scheme** MPAE-1	**Modified Scheme** MPAE-2
$\mathsf{AEn}(\mathsf{par}, \mathsf{PK}_i, \mathsf{SK}_i, \mathsf{PK}_k, m_i)$: $v_i \leftarrow_\$ \mathbb{Z}_q^*, V_i = g^{v_i}$, $(x_i, d_i) \leftarrow \mathsf{SK}_i$, $(X_i, R_i, \mathsf{n}_i) \leftarrow \mathsf{PK}_i$, $(X_k, R_k, \mathsf{n}_k) \leftarrow \mathsf{PK}_k$ $h_{1,k} = \mathcal{H}(\mathsf{n}_k, X_k, R_k)$, $Z_i = (X_k \cdot R_k \cdot A^{h_{1,k}})^{v_i}$, $h_{2,i} = \mathcal{H}_l(\mathsf{n}_k, V_i, Z_i)$, $C_i = h_{2,i} \oplus m_i$, $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$		
$s_i = v_i + d_i + x_i \cdot h_{3,i}$ **return** $\mathtt{ACT}_i = (s_i, V_i, C_i)$	$\hat{g}_i = \mathcal{H}_g(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$ $\hat{S}_i = \hat{g}_i^{v_i} \mathbf{f2}_i(\hat{g}_i) \mathbf{f1}_i(\hat{g}_i)^{h_{3,i}} = \hat{g}_i^{v_i + d_i + x_i \cdot h_{3,i}}$ **return** $\mathtt{ACT}_i = (\hat{S}_i, V_i, C_i)$	$\hat{g} = \mathcal{H}_g(com)$ $\hat{S}_i = \hat{g}^{v_i} \mathbf{f2}_i(\hat{g}) \mathbf{f1}_i(\hat{g})^{h_{3,i}} = \hat{g}^{v_i + d_i + x_i \cdot h_{3,i}}$ **return** $\mathtt{ACT}_i = (\hat{S}_i, V_i, C_i)$
$\mathsf{MA}(\mathsf{par}, \{\mathtt{ACT}_i\}_{i \in \mathtt{ID}})$: $s_i \leftarrow \mathtt{ACT}_i$, $s = \sum_{i \in \mathtt{ID}} s_i$ **return** $\mathtt{ACT} = \{s, \{V_i, C_i\}_{i \in \mathtt{ID}}\}$	$\mathsf{MA}(\mathsf{par}, \{\mathtt{ACT}_i\}_{i \in \mathtt{ID}})$: $\hat{S}_i \leftarrow \mathtt{ACT}_i$, $S = \prod_{i \in \mathtt{ID}} \hat{S}_i$ **return** $\mathtt{ACT} = \{S, \{V_i, C_i\}_{i \in \mathtt{ID}}\}$	
$\mathsf{ADe}\ (\mathsf{par}, \mathsf{SK}_k, \{\mathsf{PK}_i\}_{i \in \mathtt{ID}}, \mathtt{ACT})$: $(x_k, d_k) \leftarrow \mathsf{SK}_k$ **For each** $i \in \mathtt{ID}$ $(X_i, R_i, \mathsf{n}_i) \leftarrow \mathsf{PK}_i$, $Z_i = V_i^{(x_k + d_k)}$, $h_{1,i} = \mathcal{H}(\mathsf{n}_i, X_i, R_i)$, $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$. $h_1 = \sum_{i \in \mathtt{ID}} h_{1,i}$ $V = \prod_{i \in \mathtt{ID}} V_i$, $R = \prod_{i \in \mathtt{ID}} R_i$ **Accept iff** $g^s == V \cdot R \cdot A^{h1} \cdot \prod_{i \in \mathtt{ID}} X_i^{h_{3,i}}$ **For each** $i \in \mathtt{ID}$ $h_{2,i} = \mathcal{H}_l(\mathsf{n}_k, V_i, Z_i)$, $m_i = h_{2,i} \oplus C_i$. **return** $\{m_i\}_{i \in \mathtt{ID}}$	$\mathsf{ADe}\ (\mathsf{par}, \mathsf{SK}_k, \{\mathsf{PK}_i\}_{i \in \mathtt{ID}}, \mathtt{ACT})$: $(x_k, d_k) \leftarrow \mathsf{SK}_k$ **For each** $i \in \mathtt{ID}$ $(X_i, R_i, \mathsf{n}_i) \leftarrow \mathsf{PK}_i$, $Z_i = V_i^{(x_k + d_k)}$, $h_{1,i} = \mathcal{H}(\mathsf{n}_i, X_i, R_i)$, $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$, $\hat{g}_i = \mathcal{H}_g(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$. **Accept iff** $\hat{e}(S, g) ==$ $== \prod_{i \in \mathtt{ID}} \hat{e}(\hat{g}_i, V_i \cdot R_i \cdot A^{h_{1,i}} \cdot X_i^{h_{3,i}})$ **For each** $i \in \mathtt{ID}$ $h_{2,i} = \mathcal{H}_l(\mathsf{n}_k, V_i, Z_i)$, $m_i = h_{2,i} \oplus C_i$. **return** $\{m_i\}_{i \in \mathtt{ID}}$	$\mathsf{ADe}\ (\mathsf{par}, \mathsf{SK}_k, \{\mathsf{PK}_i\}_{i \in \mathtt{ID}}, \mathtt{ACT})$: $(x_k, d_k) \leftarrow \mathsf{SK}_k$ **For each** $i \in \mathtt{ID}$ $(X_i, R_i, \mathsf{n}_i) \leftarrow \mathsf{PK}_i$, $Z_i = V_i^{(x_k + d_k)}$, $h_{1,i} = \mathcal{H}(\mathsf{n}_i, X_i, R_i)$, $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$. $h_1 = \sum_{i \in \mathtt{ID}} h_{1,i}$, $\hat{g} = \mathcal{H}_g(com)$ $V = \prod_{i \in \mathtt{ID}} V_i$, $R = \prod_{i \in \mathtt{ID}} R_i$ **Accept iff** $\hat{e}(S, g) ==$ $== \hat{e}(\hat{g}, V \cdot R \cdot A^{h_1} \cdot \prod_{i \in \mathtt{ID}} X_i^{h_{3,i}})$ **For each** $i \in \mathtt{ID}$ $h_{2,i} = \mathcal{H}_l(\mathsf{n}_k, V_i, Z_i)$, $m_i = h_{2,i} \oplus C_i$. **return** $\{m_i\}_{i \in \mathtt{ID}}$

TABLE II: The modified schemes are in the middle and the right column.

of senders, the aggregated and authenticated ciphertext $\mathtt{ACT} = \mathsf{MA}(\{\mathtt{ACT}_i\}_{i \in \mathtt{ID}})$. *It outputs verified and decrypted messages* $\{m_i\}_{i \in \mathtt{ID}}$.

`Query stage 1` : *The adversary may issue* ℓ_1 *queries to oracles with inputs of its choice.*

`Challenge stage` : $\mathcal{A}$ *outputs two messages* $m_{i,0}$, $m_{i,1}$. *The challenger chooses* $b \in_\$ \{0,1\}$ *at random and computes the challenge* $\mathtt{ACT}_{i,b} \leftarrow \mathsf{AEn}(\mathsf{PK}_i, \mathsf{SK}_i, \mathsf{PK}_k, m_i)$.

`Query stage 2` : *The adversary may issue* ℓ_2 *queries to oracles with inputs of its choice, provided that* $\{\mathtt{ACT}_i\}_{i \in \mathtt{ID}}$ *for* $\mathcal{O}_{\mathsf{ADe}}$ *do not contain* $\mathtt{ACT}_{i,b}$.

`Response stage` : *The adversary outputs a bit* b' *indicating which message was encrypted in the challenge stage.*

The adversary advantage in the experiment

$$\mathbf{Adv}(\mathrm{Exp}_{\mathsf{ESS}}^{\lambda,\ell_1,\ell_2}(\mathcal{A}, \mathsf{MPAE})) = |\Pr[b' = b] - 1/2|$$

is the probability of the event that $\mathcal{A}$ *correctly guess which message was encrypted in the query stage. We say the scheme* MPAE *provide message secrecy if the advantage of the adversary is negligible:* $\mathbf{Adv}(\mathrm{Exp}_{\mathsf{ESS}}^{\lambda,\ell_1,\ell_2}(\mathcal{A}, \mathsf{MPAE})) \leq \epsilon(\lambda, \ell_1, \ell_2)$.

B. A Stronger Authentication Model for MPAE

We now present a stronger security model where an impersonation attack is mounted given a malicious setup of the randomness and key leakage of one of the private keys.

Definition 9 (Ephemeral Setup Impersonation (ESI))**.** *Let* $\mathsf{MPAE} = (\mathsf{Setup}, \mathsf{UKeyGen}, \mathsf{IPKeyGen}, \mathsf{PKeyGen}, \mathsf{AEn}, \mathsf{MA}, \mathsf{ADe})$ *be a Multi-Party Authentication Encryption scheme. We define the following security experiment* $\mathtt{Exp}^{\lambda,\ell}_{\mathsf{ESI}}(\mathcal{A}, \mathsf{MPAE})$*:*

`Init stage` : *The challenger generates common parameters* $\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$*, including the device indexes* Ω*. For each* $i \in \Omega$ *it generates keys:*

$(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKeyGen}(i)$,
$(\mathsf{ippk}_i, \mathsf{ipsk}_i) \leftarrow \mathsf{IPKeyGen}(\mathsf{msk}, i, \mathsf{upk}_i)$,
$(\mathsf{SK}_i, \mathsf{PK}_i) \leftarrow \mathsf{PKeyGen}(\mathsf{upk}_i, \mathsf{ippk}_i, \mathsf{ipsk}_i)$.

It chooses a set of indexes of sending devices $\mathtt{ID} \subset \Omega$ *and the receiving device* k*. The adversary* $\mathcal{A}$ *chooses one index* $\bar{\imath} \in \mathtt{ID}$*. Let* $\bar{\mathtt{ID}} = \mathtt{ID} \setminus \{\bar{\imath}\}$*.* $\mathcal{A}$ *is given all generated public data, including the public keys of the devices.*

`Oracle AEn` : *The oracle* $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$ *accepts an ephemeral value* $\bar{v}$*, parameters* par*, public key* PK_i*, public key of the recipient,* PK_k*, and a message* m_i*, then outputs* $\mathtt{ACT}_i$ *which is verifiable and decryptable. The adversary issues* ℓ *number of queries to the oracle and each time setting the ephemeral* $\bar{v}$ *the oracle uses to produce* $\mathtt{ACT}_i$*. Let* $\mathcal{M} = \{m_i\}$ *denote the set of* ℓ *messages the oracles process.*

`Forgery Game I` : $\mathcal{A}$ *is given* $\mathsf{usk}_{\bar{\imath}}$ *and return tuple:*
$\{\mathtt{ACT}^*_i\}_{i\in\mathtt{ID}} \leftarrow \mathcal{A}^{\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}}(\mathsf{usk}_{\bar{\imath}}, \{\mathsf{PK}_i\}_{i\in\bar{\mathtt{ID}}}, \mathsf{PK}_k)$.

The advantage $\mathbf{Adv}^1(\mathtt{Exp}^{\lambda,\ell}_{\mathsf{ESI}}(\mathcal{A}, \mathsf{MPAE}))$ *is defined as the probability of the event that* $\mathcal{A}^{\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}}(\mathsf{usk}_{\bar{\imath}}, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathsf{PK}_k)$ *outputs a verifiable aggregated ciphertext* $\mathtt{ACT}^*_{\bar{\imath}}$ *decryptable to some* $m^*_{\bar{\imath}}$ *which was not queried to* $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$*:*

$$\Pr\left[\begin{array}{l} \{\mathtt{ACT}^*_i\}_{i\in\mathtt{ID}} \leftarrow \mathcal{A}^{\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}}(\mathsf{usk}_{\bar{\imath}}, \{\mathsf{PK}_i\}_{i\in\bar{\mathtt{ID}}}, \mathsf{PK}_k), \\ \mathtt{ACT}^* \leftarrow \mathsf{MA}(\{\mathtt{ACT}^*_i\}_{i\in\mathtt{ID}}), \\ \{m^*_i\}_{i\in\mathtt{ID}} \leftarrow \mathsf{ADe}(\mathsf{SK}_k, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathtt{ACT}^*), \\ m^*_{\bar{\imath}} \in \{m^*_i\}_{i\in\mathtt{ID}},\ \ m^*_{\bar{\imath}} \notin \mathcal{M}. \end{array}\right].$$

`Forgery Game II` : $\mathcal{A}$ *is given* $\mathsf{psk}_{\bar{\imath}}$ *and return tuple:*
$\{\mathtt{ACT}^*_i\}_{i\in\mathtt{ID}} \leftarrow \mathcal{A}^{\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}}(\mathsf{psk}_{\bar{\imath}}, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathsf{PK}_k)$.

The advantage $\mathbf{Adv}^2(\mathtt{Exp}^{\lambda,\ell}_{\mathsf{ESI}}(\mathcal{A}, \mathsf{MPAE}))$ *is defined as the probability of the event that* $\mathcal{A}^{\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}}(\mathsf{psk}_{\bar{\imath}}, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathsf{PK}_k)$ *outputs a verifiable aggregated ciphertext* $\mathtt{ACT}^*$ *decryptable to some* $m^*_{\bar{\imath}}$ *which was not queried to* $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$*:*

$$\Pr\left[\begin{array}{l} \{\mathtt{ACT}^*_i\}_{i\in\mathtt{ID}} \leftarrow \mathcal{A}^{\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}}(\mathsf{psk}_{\bar{\imath}}, \{\mathsf{PK}_i\}_{i\in\bar{\mathtt{ID}}}, \mathsf{PK}_k), \\ \mathtt{ACT}^* \leftarrow \mathsf{MA}(\{\mathtt{ACT}^*_i\}_{i\in\mathtt{ID}}), \\ \{m^*_i\}_{i\in\mathtt{ID}} \leftarrow \mathsf{ADe}(\mathsf{SK}_k, \{\mathsf{PK}_i\}_{i\in\mathtt{ID}}, \mathtt{ACT}^*), \\ m^*_{\bar{\imath}} \in \{m^*_i\}_{i\in\mathtt{ID}},\ \ m^*_{\bar{\imath}} \notin \mathcal{M}. \end{array}\right].$$

We say that the scheme MPAE *is secure if the advantages of the adversary in both forgery games are negligible:*

$$\mathbf{Adv}^1(\mathtt{Exp}^{\lambda,\ell}_{\mathsf{ESI}}(\mathcal{A}, \mathsf{MPAE})) \le \epsilon(\lambda, \ell),$$
$$\mathbf{Adv}^2(\mathtt{Exp}^{\lambda,\ell}_{\mathsf{ESI}}(\mathcal{A}, \mathsf{MPAE})) \le \epsilon(\lambda, \ell).$$

IV. Improved MPAE Schemes Secure Against ESI

The original scheme is recalled in the first (left) column of Tables I and II. The proposed modified versions are presented in the second (middle) and third (right) columns. The devices use distinct HSMs: $\mathbf{f1}_i(\hat{g}) = \hat{g}_i^{x_i} = \hat{X}_i$, and $\mathbf{f2}_i(\hat{g}) = \hat{g}_i^{d_i} = \hat{D}_i$ to exponent with secret keys. In our first proposition, i.e. MPAE-1, the value $s_i = v_i + d_i + x_i \cdot h_{3,i}$ from the AEn procedure is moved to the exponent of a fresh generator. Thus we have $\hat{S}_i = \hat{g}_i^{v_i}\mathbf{f2}_i(\hat{g})\mathbf{f1}_i(\hat{g})^{h_{3,i}} = \hat{g}_i^{v_i}\hat{D}_i\hat{X}_i^{h_{3,i}} = \hat{g}_i^{v_i+d_i+x_i\cdot h_{3,i}}$. The verification in the ADe procedure uses a pairing function $\hat{e}$ to verify the equality of exponents. The first proposed construction MPAE-1 is asynchronous as the original scheme, so it requires $n+1$ pairing operations during verification. Since the pairing function is the most time-consuming operation, we propose another synchronous MPAE-2 construction. In this version, all messages are sent in *intervals* defined by thresholded timestamps (e.g. per whole minutes) or by challenges broadcasted from the aggregation device. We denote those timeframes via unique com bit sequences. It is used as the input for the hash $\mathcal{H}_g(com)$ to get the same fresh generator $\hat{g}$ for each sending device in the interval related to com.

In MPAE-2 we only allow for one signing per device in each com. Otherwise a simple repetition attack could be mounted. The adversary gets two valid $\hat{S}_i = \hat{g}^{v_i+d_i+x_i\cdot h_{3,i}}$ and $\hat{S}'_i = \hat{g}^{v'_i+d_i+x_i\cdot h'_{3,i}}$ for two messages m, m' in a single com. It then computes $\hat{S}_i/\hat{S}_i = \hat{g}^{(v_i-v'_i)+x_i(h_{3,i}-h'_{3,i})}$ and $\hat{g}^{x_i} = ((\hat{S}_i/\hat{S}_i)/(\hat{g}^{(v_i-v'_i)}))^{(1/(h_{3,i}-h'_{3,i}))}$. Next, it computes $\hat{g}^{d_i} = \hat{S}_i/g^{v_i}(g^{x_i})^{h_{3,i}}$. Now the attacker could produce a fresh $\bar{v}_i$ and use it to encrypt a fresh m^*_i for the recipient k, as in the AEn procedure resulting with C^*_i. It uses that value in $\bar{h}_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C^*_i, X_i, R_i, Z_i)$. Now it can produce a verifiable forgery $\hat{S}^*_i = \hat{g}^{\bar{v}_i}\hat{g}^{d_i}(\hat{g}^{x_i})^{\bar{h}_{3,i}}$ for the message m^*_i encrypted to C^*_i.

A. Ephemeral Leakage Attack on the Original scheme

In Theorem 1 we state that the original construction from [14] is not secure in our model.

Theorem 1. *The original scheme* MPAE *in the left column of Tab. II is not secure in our new stronger model, as of Def. 9.*

Proof. After the system is initialized, the adversary $\mathcal{A}$ selects two ephemerals $v_{\bar{\imath}}, v'_{\bar{\imath}} \leftarrow_{\$} \mathbb{Z}_q$ and use them to query $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$ twice for arbitrary messages m, m' and a recipient device k, obtaining:

$$\begin{aligned}(s_{\bar{\imath}}, V_{\bar{\imath}}, C_{\bar{\imath}}) &= \mathtt{ACT}_{\bar{\imath}} \leftarrow \mathcal{O}^{\bar{v}}_{\mathsf{AEn}}(\mathsf{PK}_{\bar{\imath}}, \mathsf{PK}_k, m),\\ (s'_{\bar{\imath}}, V'_{\bar{\imath}}, C'_{\bar{\imath}}) &= \mathtt{ACT}'_{\bar{\imath}} \leftarrow \mathcal{O}^{\bar{v}'}_{\mathsf{AEn}}(\mathsf{PK}_{\bar{\imath}}, \mathsf{PK}_k, m').\end{aligned}$$

Next, it computes: $h_{1,k} = \mathcal{H}(\mathsf{n}_k, X_k, R_k)$, and values:

$$Z_{\bar{\imath}} = (X_k \cdot R_k \cdot A^{h_{1,k}})^{v_{\bar{\imath}}}, \quad h_{3,\bar{\imath}} = \mathcal{H}(V_{\bar{\imath}}, \mathsf{n}_{\bar{\imath}}, C_{\bar{\imath}}, X_{\bar{\imath}}, R_i, Z_{\bar{\imath}}),$$
$$Z'_{\bar{\imath}} = (X_k \cdot R_k \cdot A^{h_{1,k}})^{v'_{\bar{\imath}}}, \quad h'_{3,\bar{\imath}} = \mathcal{H}(V'_i, \mathsf{n}_{\bar{\imath}}, C'_{\bar{\imath}}, X_{\bar{\imath}}, R_{\bar{\imath}}, Z'_{\bar{\imath}}).$$

These allows to build the system of equations:

$$(1):\ \ s_{\bar{\imath}} = v_{\bar{\imath}} + d_i + x_i \cdot h_{3,\bar{\imath}}, \quad (2): s'_{\bar{\imath}} = v'_{\bar{\imath}} + d_{\bar{\imath}} + x_i \cdot h'_{3,\bar{\imath}}.$$

Solving this system the adversary obtains:

$$x_i = ((s_{\bar{\imath}} - s'_{\bar{\imath}}) - (v_{\bar{\imath}} - v'_{\bar{\imath}}))\,/(h_{3,\bar{\imath}} - h'_{3,\bar{\imath}}).$$

Having $x_{\bar{\imath}}$, the adversary $\mathcal{A}$ can easily compute the value $d_{\bar{\imath}}$:

$$d_{\bar{\imath}} = s_{\bar{\imath}} - v_{\bar{\imath}} - x_{\bar{\imath}} \cdot h_{3,\bar{\imath}},$$

which is the secret of $\bar{\imath}$-th device, allowing impersonation. □

V. Security Analysis of Proposed Schemes

A. Correctness of The Proposed Schemes

Theorem 2. *The scheme* MPAE-1 *proposed in the middle column of Tab. II is correct.*

Proof. The value of Z_i computed by the verifier k, equals Z_i computed by the i-th sender. Indeed:

$$Z_i = V_i^{x_k+d_k} = g^{v_i\cdot(x_k+r_k+a\cdot h_{1,k})} = (X_k \cdot R_k \cdot A^{h_{1,k}})^{v_i}.$$

In the verification we have: $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$, and therefore:

$$\begin{aligned}\hat{e}(S,g) &= \hat{e}(\prod_{i\in\mathtt{ID}} \hat{S}_i, g) = \prod_{i\in\mathtt{ID}} \hat{e}(\hat{g}_i^{v_i+d_i+x_i\cdot h_{3,i}}, g)\\ &= \prod_{i\in\mathtt{ID}} \hat{e}(\hat{g}_i, g^{v_i+d_i+x_i\cdot h_{3,i}})\\ &= \prod_{i\in\mathtt{ID}} \hat{e}(\hat{g}_i, V_i \cdot R_i \cdot A^{h_{1,i}} \cdot X_i^{h_{3,i}}).\end{aligned}$$ □

Theorem 3. *The scheme* MPAE-2 *proposed in the right column of Tab. II is correct.*

Proof. Again, as in the previous proof, the value of Z_i computed by the verifier k equals Z_i computed by the i-th sender. Indeed:

$$Z_i = V_i^{x_k+d_k} = g^{v_i\cdot(x_k+r_k+a\cdot h_{1,k})} = (X_k \cdot R_k \cdot A^{h_{1,k}})^{v_i}.$$

We have $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$, and therefore:

$$\begin{aligned}\hat{e}(S,g) &= \hat{e}(\prod_{i\in\mathtt{ID}} \hat{S}_i, g) = \hat{e}(\hat{g}^{\sum_{i\in\mathtt{ID}} v_i+d_i+x_i h_{3,i}}, g)\\ &= \hat{e}(\hat{g}_i, g^{\sum_{i\in\mathtt{ID}} v_i+d_i+x_i h_{3,i}})\\ &= \hat{e}(\hat{g}_i, \prod_{i\in\mathtt{ID}} V_i \cdot \prod_{i\in\mathtt{ID}} R_i \cdot A^{\sum_{i\in\mathtt{ID}} h_{1,i}} \cdot \prod_{i\in\mathtt{ID}} X_i^{h_{3,i}})\\ &= \hat{e}(\hat{g}_i, V \cdot R \cdot A^{h_1} \cdot \prod_{i\in\mathtt{ID}} X_i^{h_{3,i}}).\end{aligned}$$ □

B. Unforgeability of The New Schemes Under Impersonation Attacks

Theorem 4. *The scheme proposed in the middle column of Tab. II is secure in the sense of Definition 9.*

Sketch of the proof. `Init` : Let (g, g^α, g^β) be an instance of the GDH problem in $\mathsf{par} = (\mathbb{G}, \mathbb{G}_T, g, g_T, q, \hat{e})$.

`Forgery I Setup` : We generate all keys except for the device $\bar{\imath}$, for which we setup: $\mathsf{upk}_{\bar{\imath}} = X_{\bar{\imath}} = g^{x_{\bar{\imath}}} = g^\alpha$. Thus the unknown secret key $\mathsf{usk}_{\bar{\imath}} = x_{\bar{\imath}}$ equals the unknown α. We provide the adversary $\mathcal{A}$ access to $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$, and hash $\mathcal{O}_{\{\mathcal{H}\}}$ oracles, and all public values, as in the security game.

`Forgery II Setup` : We setup the system s.t. $\mathsf{mpk} = A = g^a = g^\alpha$. Thus the unknown $\mathsf{msk} = a$ equals the unknown α. We generate all secret keys $\mathsf{usk}_i = x_i$ at random, and compute $\mathsf{upk}_i = X_i$. We compute $r_i \leftarrow_\$ \mathbb{Z}_q^*, R_i = g^{r_i}$, $\mathsf{n}_i \leftarrow_\$ \{0,1\}^l$, $h_{1,i} = \mathcal{H}(\mathsf{n}_i, X_i, R_i)$ as in the IPKeyGen procedure. However, we do not compute u_i as we do not know the secret $a = \alpha$. Also all $\mathsf{psk}_i = d_i$ are unknown. We provide the adversary $\mathcal{A}$ with access to $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$, and hash $\mathcal{O}_{\{\mathcal{H}\}}$ oracles, and all public values, as in the security game.

`Serving` $\mathcal{O}_{\mathcal{H}_g}$ `Oracle` : We allow ℓ fresh inputs to the $\mathcal{O}_{\mathcal{H}_g}$ oracle. We choose the random index $j \leftarrow_\$ \{1, \ldots, \ell\}$, which denotes the j-th invocation of $\mathcal{O}_{\mathcal{H}_g}$, for which we assume the forgery will happen.

- On the i-th, $(i \neq j)$ fresh input $V, \mathsf{n}, C, X, R, Z$, we compute $\omega \leftarrow_\$ \mathbb{Z}_q^*$, and register the value $\hat{g} = \mathcal{H}_g(V, \mathsf{n}, C, X, R, Z)$ in the ROM table for $\mathcal{H}_g$. We return $\hat{g}$ as the output.
- On the j-th fresh input $V, \mathsf{n}, C, X, R, Z$, we set $\hat{g} = (g^\beta)$ and register that value in the ROM table for $\mathcal{H}_g$. We return $\hat{g}$ as the output.

`Serving` $\mathcal{O}^{\bar{v}}_{\mathsf{AEn}}$ `Oracle` : On input $\bar{v}$ we compute $\bar{V}_i = g^{\bar{v}}$, $h_{1,k} = \mathcal{H}(\mathsf{n}_k, X_k, R_k)$, $Z_i = (X_k R_k A^{h_{1,k}})^{\bar{v}}$, $h_{2,i} = \mathcal{H}_l(\mathsf{n}_k, V_i, Z_i)$, and $C_i = h_{2,i} \oplus m_i$, $h_{3,i} = \mathcal{H}(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$. We serve the call $\mathcal{H}_g(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$, and if $(V_i, \mathsf{n}_i, C_i, X_i, R_i, Z_i)$ was not j-th fresh input to $\mathcal{O}_{\mathcal{H}_g}$, we return $\hat{g}_i = g^{\omega_i}$ for some ω_i registered in the ROM table. We compute $\hat{S}_i = \hat{g}_i^{v_i+d_i+x_i\cdot h_{3,i}}$ as $(V_i R_i A^{h_{1,i}} X_i^{h_{3,i}})^{\omega_i}$, and return $\mathtt{ACT}_i = (\hat{S}_i, V_i, C_i)$. The verification in ADe holds as: $\hat{e}(S,g) = \hat{e}(\prod_{i\in\mathtt{ID}} \hat{S}_i, g) = \prod_{i\in\mathtt{ID}} \hat{e}(g^{\omega_i(v_i+d_i+x_i h_{3,i})}, g)$ which equals $\prod_{i\in\mathtt{ID}} \hat{e}(g^{\omega_i}, g^{(v_i+d_i+x_i h_{3,i})})$, which equals $\prod_{i\in\mathtt{ID}} \hat{e}(\hat{g}_i, V_i R_i A^{h_{1,i}} X_i^{h_{3,i}})$.

`Processing Forgery Game I` : According to the *Forking Lemma* for the hash $h_{3,\bar{\imath}} = \mathcal{H}(V_{\bar{\imath}}, \mathsf{n}_{\bar{\imath}}, C_{\bar{\imath}}, X_{\bar{\imath}}, R_{\bar{\imath}}, Z_{\bar{\imath}})$ the attacker returns two valid forgeries $\mathtt{ACT}^*_{\bar{\imath}}, \mathtt{ACT}^{*'}_{\bar{\imath}}$ for the respective tuples $(v_{\bar{\imath}}, h_{3,\bar{\imath}}, \hat{S}_{\bar{\imath}})$, $(v_{\bar{\imath}}, h'_{3,\bar{\imath}}, \hat{S}'_{\bar{\imath}})$ with the same randomness $v_{\bar{\imath}}$. We compute $\hat{S}_{\bar{\imath}}/\hat{S}'_{\bar{\imath}} = \hat{g}_{\bar{\imath}}^{v_{\bar{\imath}}+d_{\bar{\imath}}+x_{\bar{\imath}}h_{3,i}}/\hat{g}_{\bar{\imath}}^{v_{\bar{\imath}}+d_{\bar{\imath}}+x_{\bar{\imath}}h'_{3,\bar{\imath}}}$ equal to $\hat{g}_i^{x_{\bar{\imath}}(h_{3,\bar{\imath}}-h'_{3,\bar{\imath}})}$. With the non-negligible probability $1/\ell$, related to j-th fresh input to $\mathcal{O}_{\mathcal{H}_g}$, then $\hat{g}_{\bar{\imath}}$ equals g^β in both tuples. Thus we have $\hat{S}_{\bar{\imath}}/\hat{S}'_{\bar{\imath}} = g^{\beta x_{\bar{\imath}}(h_{3,\bar{\imath}}-h'_{3,\bar{\imath}})}$. Since we set $X_{\bar{\imath}} = g^{x_{\bar{\imath}}} = g^\alpha$, we can compute $g^{\alpha\beta} = (\hat{S}_{\bar{\imath}}/\hat{S}'_{\bar{\imath}})^{(1/(h_{3,\bar{\imath}}-h'_{3,\bar{\imath}}))}$, breaking the given instance of GDH.

`Processing Forgery Game II` : The successful attacker returns a verifiable and decryptable $\mathtt{ACT}^*_{\bar{\imath}} = (\hat{S}_{\bar{\imath}}, V_{\bar{\imath}}, C_{\bar{\imath}})$. With the non-negligible probability $1/\ell$, it is related to j-th fresh input to $\mathcal{O}_{\mathcal{H}_g}$ - so $\hat{g}_{\bar{\imath}}$ equals g^β. Thus we have $\hat{S}_{\bar{\imath}} = \hat{g}_{\bar{\imath}}^{v_{\bar{\imath}}+d_{\bar{\imath}}+x_{\bar{\imath}}h_{3,i}} = (g^\beta)^{(v_{\bar{\imath}}+(r_{\bar{\imath}}+a\cdot h_{1,\bar{\imath}})+x_{\bar{\imath}}h_{3,i})}$. Since we set $\mathsf{msk} = A = g^a = g^\alpha$ we can compute $g^{\alpha\beta} = (\hat{S}_{\bar{\imath}}/(g^\beta)^{(v_{\bar{\imath}}+r_{\bar{\imath}}+x_{\bar{\imath}}h_{3,i})})^{1/h_{1,\bar{\imath}}}$ for $h_{1,\bar{\imath}} = \mathcal{H}(\mathsf{n}_{\bar{\imath}}, X_{\bar{\imath}}, R_{\bar{\imath}})$, breaking the given instance of GDH. □

TABLE III: Procedure performance and scalability analysis, all timings are measured in milliseconds.

Procedure	MPAE	MPAE − 1	MPAE − 2
Setup	[0, 1, 0, 0] 3.9131	[0, 1, 0, 0] 6.1523	[0, 1, 0, 0] 6.1680
UKeyGen	[0, n, 0, 0] 1.2198	[0, n, 0, 0] 1.2215	[0, n, 0, 0] 1.2302
IPKeyGen	[0, n, 0, 0] 2.4871	[0, n, 0, 0] 2.3537	[0, n, 0, 0] 2.3547
PKeyGen	[0, n, 0, 0] 1.1860	[0, n, 0, 0] 1.1177	[0, n, 0, 0] 1.1242
AEn	[2, $3n$, 0, 0] 8.6258	[2, $4n$, 1, 0] 11.7533	[2, $4n$, 1, 0] 11.7580
MA, $n = 1$	[0, 0, 0, 0] 0.0012	[n, 0, 0, 0] 0.0013	[n, 0, 0, 0] 0.0013
ADe, $n = 1$	[$2 + 3n$, $2n$, 0, 0] 0.0096	[$3n$, $3n$, n, $n + 1$] 0.0114	[$3n + 2$, $3n$, 1, 2] 0.0109
MA, $n = 10$	[0, 0, 0, 0] 0.0037	[n, 0, 0, 0] 0.0512	[n, 0, 0, 0] 0.0512
ADe, $n = 10$	[$2 + 3n$, $2n$, 0, 0] 0.0543	[$3n$, $3n$, n, $n + 1$] 0.1042	[$3n + 2$, $3n$, 1, 2] 0.0496
MA, $n = 100$	[0, 0, 0, 0] 0.0204	[n, 0, 0, 0] 0.5563	[n, 0, 0, 0] 0.5564
ADe, $n = 100$	[$2 + 3n$, $2n$, 0, 0] 0.4566	[$3n$, $3n$, n, $n + 1$] 1.1097	[$3n + 2$, $3n$, 1, 2] 0.4562

Theorem 5. *The scheme proposed in the right column of Tab. II is secure in the sense of Definition 9.*

Sketch of the proof. Essentially similar to the proof of Theorem 4, changing the definition of the oracle $\mathcal{O}_{\mathcal{H}_g}$, which with input com is unique for a single time frame, returning the same $\hat{g}$ for all devices. And devices are allowed to sign and authenticate only one message per single frame com. □

VI. Implementation and performance

Our proposed schemes are based on symmetric pairings. However, a scheme based on asymmetric pairings would still be secure under a similar security analysis. All testing was performed on Ubuntu 18.04.5 LTS with an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz. The implementation was done in Python using the Charm crypto library [17] with the SS512 curve for pairings. Table III shows the number of operations of each procedure in brackets, i.e. [`G:mul`, `G:exp`, `G:hashTo`, `Pairing`] where n is the number of `IOTD` senders. Table IV shows the total number of fundamental operations for each scheme.

TABLE IV: Comparison of overall complexity.

Operation	Time (ms)	MPAE	MPAE − 1	MPAE − 2
`G:mul`	0.0033	$3n + 4$	$4n + 2$	$3n + 4$
`G:exp`	1.1089	$8n + 1$	$10n + 1$	$10n + 1$
`G:hashTo`	2.5223	0	$n + 1$	2
`Pairing`	0.6855	0	n	2

VII. Conclusion

We have considered a multi-party authenticated encryption scheme that fits well for NB-IoT systems over 5G architectures. We analyze the previous proposed scheme [14] in the new stronger security model with ephemeral key leakage and prove it to be unsecure. Subsequently we propose two modifications of that scheme, synchronized and asynchronized, which are provable secure in our stronger model. Our proof of concept implementation shows the feasibility and scalability of the schemes.

Acknowledgment

The research was partially financed from the Fundamental Research Fund nr 8201003902 of the Wrocław University of Science and Technology.

References

[1] "3GPP Low Power Wide Area Technologies," https://www.gsma.com/iot/resources/3gpp-low-power-wide-area-technologies-white-paper/.

[2] S. Tabbane, "IoT standards Part II: 33GPP Standards GPP Standards Training on Planning Internet Of Things (IoTs) Networks," 2018.

[3] E. C. Matthias Ruete, "ERTMS: First Work Plan of the European Coordinator," 2020.

[4] I. Lopez and M. Aguado, "Cyber security analysis of the european train control system," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 110–116, 2015.

[5] R. Bloomfield, R. Bloomfield, I. Gashi, and R. Stroud, "How secure is ertms?" in *Computer Safety, Reliability, and Security*, F. Ortmeier and P. Daniel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 247–258.

[6] D. A. Hahn, A. Munir, and V. Behzadan, "Security and privacy issues in intelligent transportation systems: Classification and challenges." *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 1, pp. 181–196, 2021.

[7] A. Lamssaggad, N. Benamar, A. S. Hafid, and M. Msahli, "A survey on the current security landscape of intelligent transportation systems," *IEEE Access*, vol. 9, pp. 9180–9208, 2021.

[8] J. M. Vidal, A. L. S. Orozco, and L. J. G. Villalba, "Mitigation of DDoS Attacks in 5G Networks: a Bio-inspired Approach," *National Academies of Sciences, Engineering, and Medicine*, 2017.

[9] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 398–412.

[10] L. Goubin and J. Patarin, "DES and Differential Power Analysis The "Duplication" Method," in *Cryptographic Hardware and Embedded Systems*, Ç. K. Koç and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 158–172.

[11] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Advances in Cryptology - CRYPTO 2009*, S. Halevi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 36–54.

[12] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai, "Exposure-resilient functions and all-or-nothing transforms," in *EUROCRYPT*, 2000.

[13] Łukasz Krzywiecki, A. Bobowski, M. Słowik, M. Słowik, and P. Kozieł, "Schnorr-like identification scheme resistant to malicious subliminal setting of ephemeral secret," *Computer Networks*, vol. 179, p. 107346, 2020.

[14] Y. Zhang, F. Ren, A. Wu, T. Zhang, J. Cao, and D. Zheng, "Certificateless Multi-Party Authenticated Encryption for NB-IoT Terminals in 5G Networks," 2019.

[15] M. Manulis and K. Suzuki, "Modeling Leakage of Ephemeral Secrets in Tripartite/Group Key Exchange," 2019.

[16] C.-L. Liu, W.-J. Tsai, T.-Y. Chang, and T.-M. Liu, "Ephemeral-Secret-Leakage Secure ID-Based Three-Party Authenticated Key Agreement Protocol for Mobile Distributed Computing Environments," 2018.

[17] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.