Journal Pre-proof

SPrivAD: A Secure and Privacy-Preserving Mutually Dependent Authentication and Data Access Scheme for Smart Communities

Abubakar Sadiq Sani, Elisa Bertino, Dong Yuan, Ke Meng, Zhao Yang Dong

PII: S0167-4048(22)00009-8
DOI: https://doi.org/10.1016/j.cose.2022.102610
Reference: COSE 102610

To appear in: *Computers & Security*

Received date: 3 August 2020
Revised date: 21 December 2021
Accepted date: 10 January 2022

Please cite this article as: Abubakar Sadiq Sani, Elisa Bertino, Dong Yuan, Ke Meng, Zhao Yang Dong, SPrivAD: A Secure and Privacy-Preserving Mutually Dependent Authentication and Data Access Scheme for Smart Communities, *Computers & Security* (2022), doi: https://doi.org/10.1016/j.cose.2022.102610

SPrivAD: A Secure and Privacy-Preserving Mutually Dependent Authentication and Data Access Scheme for Smart Communities

Abubakar Sadiq Sani[a], Elisa Bertino[b], Dong Yuan[c], Ke Meng[d], Zhao Yang Dong[d]

[a] *University of Greenwich, London, SE10 9LS, UK*
[b] *Purdue University, West Lafayette, IN 47907, USA*
[c] *The University of Sydney, Sydney, NSW 2006, Australia*
[d] *University of New South Wales, Sydney, NSW 2052, Australia*

Abstract

Recent studies show that attackers evade authentication by exploiting valid credentials and crafting authentication request messages to compromise assets and illegitimately access data in smart communities such as smart campuses and smart cities. In addition, attackers can send large numbers of authentication and data access requests to spread malware across the smart communities' network and cause Distributed Denial of Service (DDoS) attacks. This paper proposes SPrivAD, a secure and privacy-preserving mutually dependent authentication and data access solution by which smart communities' assets such as users, devices, and apps can authenticate each other before allowing data access. SPrivAD uses an Inter-Attribute-based Zero Knowledge Proof of Knowledge (IA-ZKPK) protocol based on computational attributes of cryptographic operations, and cryptographic identities of the assets to perform Mutually Dependent Multi-Factor Authentication and Data Access (MDMFA). The computational attributes such as message size and number of executed steps of cryptographic operations are features derived from the knowledge of cryptographic operations between the assets. Our approach for deriving a unique, deactivatable, and revocable cryptographic identity is based on the secrets of an asset in a modified Elliptic Curve Pedersen Commitment Scheme (EC-PCS) with security and privacy guarantees. We implement a prototype of SPrivAD and evaluate it with respect to its security, privacy, and performance. The results show that it is secure, privacy-preserving, and efficient for mutually dependent authenti-

cation and data access in smart communities. Furthermore, we design and analyse a new attack, Smart Communities Authentication Bypass Attack (SCABA), on real-world authentication and secure access schemes such as Ruckus Cloudpath Enrollment System and Duo Multi-Factor Authentication (MFA). This type of attack exploits valid credentials of smart communities' assets. We show that SPrivAD mitigates SCABA.

Keywords: Smart Communities, Authentication, Data Access, Security, Privacy

1. Introduction

Assurance on the verification of an asset's identity and security and privacy of the identity are key requirements for authentication and data access in smart communities, which refer to Internet of Things (IoT) applications and interconnect users, devices, and apps to create intelligent services, automate services delivery, and enhance operational efficiency. Public key authentication is a stronger authentication method compared to traditional symmetric key and password-based authentication methods. However, public key authentication has both advantages, such as using private keys for stronger identity verification, and disadvantages, such as distribution of public keys and storage of private keys on assets, and reuse of private keys. These disadvantages can cause exploitation of sensitive information which may lead to authentication evasion and illegitimate data access, which can further cause malware spread across smart communities' assets. Therefore, an authentication and data access mechanism suitable for smart communities needs to address those shortcomings. In this work, the key requirement is thus to develop mutually dependent techniques supporting the generation of strong cryptographic keys from cryptographic identities, which are unique, deactivatable, and revocable and used during authentication. The cryptographic keys can then be used for granting data access in smart communities.

Many smart communities Authentication and Identity Management (AIM) solution providers, such as Cisco [1], Amazon [2], and Ruckus Networks [3], are increasingly adapting and deploying public key authentication. However, existing public key-based remote authentication systems have security and privacy concerns. Different authentication features such as identities or cryptographic keys are used by solution providers to implement their proprietary public key authentication mechanism, which requires smart communities' as-

sets to perform an asset registration process. Upon an asset registration, the public key authentication system records an asset's digital certificate (that is used to prove the identity of the asset) and stores it in some database. The digital certificate is validated when the asset needs to authenticate. Security of the database and certificate are thus important for the public key authentication. If the database and/or certificate are compromised, smart communities' assets may be exposed to attackers that can evade authentication and illegitimately access data.

Providing asset information to too many AIM solution providers increases security and privacy risks. To address such risks, a trusted authority (or certificate provider) could be deployed to register the asset information. When an asset needs to authenticate to a solution provider, the solution provider uses the asset information signed and issued by the trusted authority as well as the trusted authority information to authenticate the asset. Thus, the solution provider relies on the trusted authority for the asset authentication so that the asset does not have to register or reveal his/her information at the solution provider, thereby providing better asset information protection. However, such authentication solution raises other types of security, privacy, and performance concerns as follows: i) because the solution provider relies on the trusted authority for authentication transaction, it can lead to single point of failure, DDoS attack, and attacks to escalate attacker privileges to administrative level (say, the trusted authority level); ii) because the solution provider needs to validate the asset and trusted authority information for each authentication transaction, sensitive information about the asset's transactions can be gathered by the solution provider, thereby exposing the privacy of the asset; and iii) because the solution provider uses computational resources to access and validate asset and trusted authority information, it increases the communication and computational overheads, which can make it difficult to meet our 20 *msec* latency target of smart communities' applications such as renewable and smart grids [4], which enable communities to utilise local renewable energy sources to effectively capture clean energy. As today commercial products [5] support public key authentication based on the trusted authority approach, such as security, privacy, and performance concerns are not addressed.

The goal of this paper is to propose SPrivAD, a secure, privacy-preserving, and efficient scheme for authentication and data access between smart communities' assets based on their cryptographic identities and secrets. SPrivAD does not have the drawbacks that we have discussed, namely: i) it does not

require the involvement of solution providers or a trusted authority in authentication transaction as it uses registration and identity providers ($RIDs$) to register or enrol assets before authentication and data access note that the $RIDs$ can be regarded as trusted servers with synchronised distributed databases to ensure that the databases maintain identical information and are always available for data storage (in an encrypted manner) and verification; ii) it does not require storing sensitive information at the $RIDs$ or solution providers; and iii) it meets the 20 $msec$ latency target. SPrivAD uses an Inter-Attribute-based Zero Knowledge Proof of Knowledge (IA-ZKPK) protocol to design a Mutually Dependent Multi-Factor Authentication and Data Access (MDMFA) protocol, which leverages the security and privacy properties of the ZKPK protocol [6]. It also addresses the ZKPK challenge of reflection attack [7], which is carried out to impersonate an honest asset, and other challenges of public key authentication carried out from smart communities' assets by: i) deriving a unique, deactivatable, and revocable cryptographic identity from every asset's secrets; ii) including a mutually dependent computational attributes in the authentication protocol to support mitigation of Smart Communities Authentication Bypass Attack (SCABA), reflection attack, and crafting of authentication requests messages; iii) including a key establishment mechanism for the authentication protocol to support secure communication, identity verification, and granting data access; and iv) employing lightweight cryptographic algorithms and operations that meet the 20 $msec$ latency target. The main contributions of our work are as follows.

- We propose a Mutually Authenticated Registration (MAR) protocol that registers every asset to SPrivAD by modifying the Elliptic Curve Pedersen Commitment Scheme (EC-PCS) [8] to provide security and privacy guarantees.
- We propose an MDMFA protocol between two assets using the IA-ZKPK protocol by which the secrets of the commitment scheme, cryptographic identities of the assets, and computational attributes of cryptographic operations are used for authentication and deriving a unique shared secret key for securely granting data access.
- We analyse the security and privacy of SPrivAD. Furthermore, we validate the security of SPrivAD using the Automated Validation of Internet Security Protocol and Application (AVISPA) tool [9].

- We implement a prototype of SPrivAD and show that it is efficient and meets our latency target.
- We identify SCABA as a threat for smart communities' assets and its absence is not guaranteed by real-world authentication and secure access schemes such as Ruckus Cloudpath Enrollment System [10] and Duo Multi-Factor Authentication scheme [11] for the smart communities.
- We provide security analyses of the Ruckus Cloudpath Enrollment System and Duo Multi-Factor Authentication (MFA) scheme in smart campus scenarios and find some weaknesses in their authentication procedures. We show that using SPrivAD, slight variations of the schemes offer stronger authentication and secure access benefits.

The rest of the paper is organised as follows. Section 2 introduces the main building blocks used in our scheme. Section 3 presents a network architecture of smart communities and an attack model. Section 4 presents SPrivAD. Section 5 presents the security and privacy analyses of our scheme. Section 6 presents the security validation of our scheme. Section 7 presents the implementation and experimental results for our scheme. Section 8 introduces SCABA. Section 9 presents case studies. Section 10 discusses related work. Section 11 concludes the paper and outlines future work.

2. Background

In this section, we introduce the main building blocks in SPrivAD which consists of the EC-PCS and ZKPK protocol.

2.1. Elliptic Curve Pedersen Commitment Scheme

The EC-PCS is an efficient implementation of the Pedersen Commitment Scheme (PCS) [12] which uses Elliptic Curve Cryptography (ECC) [13] based on the elliptic curve discrete logarithm assumption. In this scheme, a committer commits to secrets such that it is hard for a verifier to open the commitment. The description of the EC-PCS presented in the following steps:

- Setup: Let F_p be a group of elliptic curve points, where p is a large prime of 128 *bits*. Let Z_p be an integer group of order p. Let $G \in F_p$

be a random generator point of order n and $H \in F_p$ be a chosen generator point of n such as it is computationally hard to find $H = x_H.G$, where $x_H \in Z_p$ is a random secret (see [12] for more details). A trusted authority, say an RID, publishes the elliptic curve domain parameters (p, a, b, G, H, n, h), where a and b are curve parameters and h is a cofactor.

- Commit: The committer creates a commitment C of $x \in Z_p$ by randomly choosing $r \in Z_p$ and computing $C(x, r) = x.G+r.H$. Please note that $C(x, r)$ represents a dedicated value created by the committer.
- Reveal: To confirm the authenticity of C, the committer reveals x and r and the verifier checks if $C = x.G + r.H$.

The EC-PCS and PCS have similar properties as follows: i) Perfectly hiding, i.e., every possible random secret x is equally committed in C; and ii) Computationally binding, i.e., a random secret $x^{'}$ cannot open the commitment since $x^{'} \notin x$, unless one can solve the Elliptic Curve Discrete Logarithm Problem (ECDLP). In secure implementations of ECC, it is difficult to guess x_H from $H = x_H.G$ as it is to guess x_H from g^{x_H}, where g is an integer. Thus, this is called the ECDLP.

In this paper, we modify the EC-PCS to support the derivation of a unique, deactivatable, and revocable cryptographic identity for an asset without transmitting and revealing the actual secrets of the asset thereby providing security and privacy guarantees to the secrets.

2.2. Zero Knowledge Proof of Knowledge Protocol

The ZKPK protocol [6] is a protocol that allows a prover to prove his/her knowledge of a secret to a verifier without revealing the secret or allowing the verifier to obtain the actual secret. To prove the knowledge of the secrets in the EC-PCS, we use the ZKPK protocol to hide the secrets from which the EC-PCS is computed, where the RID chooses and publishes G_p - a subgroup of Z_p, g - a generator of G_p, and $q = g^o \ mod \ p$ (where $'o'$ is a secret) - an element of G_p. The description of this protocol between a prover A and verifier B is presented in the following steps:

- $A \rightarrow B$: A chooses random secrets $s, t \in Z_p$ and sends $d = g^s q^t \in G_p$ to B.

- $B \rightarrow A$: B sends a random challenge $e \in Z_p$ to A.
- $A \rightarrow B$: A sends $\alpha = s + e.x$ and $\beta = t + e.r$ to B.
- B: B accepts α and β if $g^{\alpha}q^{\beta} = d.C^{e}$.

The ZKPK protocol has three properties as follows: i) Completeness, i.e., the protocol succeeds with overwhelming probability if both A and B are honest; ii) Zero Knowledge, i.e., the proof does not leak any information about the secrets; and iii) Soundness, i.e., the protocol does not allow the prover to prove a false statement.

In this paper, we enhance the efficiency of the ZKPK protocol and make it easier to use in the smart communities by introducing the IA-ZKPK protocols, which uses ECC for efficiency and integrates the computational attributes of assets to primarily support multi-factor authentication. The description of the IA-ZKPK protocol is presented in the following steps:

- $A \rightarrow B$: A chooses random secrets $s, t \in Z_p$, computes $d = s.G + t.H$, derives computational attributes S_{Msg_A} and ST_A, and sends d, S_{Msg_A}, and ST_A to B, where S_{Msg_A} is size of a message Msg_A and ST_A is number of executed steps of cryptographic operations by A.
- $B \rightarrow A$: B obtains S_{Msg_B} and ST_B, verifies if $S_{Msg_B} = S_{Msg_A}$ and $ST_B = ST_A$, selects a random challenge $e \in Z_p$, and sends S_{Msg_B}, ST_B, and e to B if the verifications succeed. Note that: i) A and B have knowledge of cryptographic operations; and ii) without loss of generality, $Msg_A = Msg_B$ and $ST_A = ST_B$ in our scheme (see Section 4 for more details).
- $A \rightarrow B$: A verifies if $S_{Msg_A} = S_{Msg_B}$ and $ST_A = ST_B$, and sends $\alpha = s + e.x$ and $\beta = t + e.r$ to B if the verifications succeed, where α, β are authentication values computed to support authentication.
- B: B accepts α and β if $\alpha.G + \beta.H = d + e.C$.

The notations used in this work are listed in Table 1.

3. Network Architecture and Threat Model

In this section, we present a network architecture of smart communities and an attack model.

Table 1: Notations and Meanings

Notations	Meanings
p	Large prime of 128 *bits*
a, b	Curve parameters
G	Random generator point
H	Chosen generator point
n	Order of G and H
h	Cofactor
ID_A	Identity of asset A
x, r, s, t	Random secrets
C, d	Cryptographic Commitments
pv_{rid}	Private key of provider RID
Q_{rid}	Public key of provider RID
$Sig_{pv_{rid}}(.)$	ECDSA digital signature created by provider RID
T_{info}	Discrete clock information
S_{Msg_A}	Size of a message from asset A
ST_A	Number of executed steps by asset A
k_{pk}	Preshared key
ATF	Artifacts
T	Timestamp
RN	Random number
vv	Verification value
e	Random challenge
α, β	Authentication values
k	Shared secret session key
$Hash(k, .)$	Secure Hash Function (SHA-2)
$F(.)$	Pseudo-Random Function (PRF)
$Enc_k(.)$	Advanced Encryption Standard (AES)
$MAC_k(.)$	Keyed-Hash Message Authentication Code (HMAC)

3.1. Network Architecture

Every domain such as campus and commercial centre in smart communities consists of a collection of sensing devices that monitor and collect information within the smart communities. Figure 1 represents a simple network architecture of smart communities in this work. The figure shows that through the Wide Area Network (WAN), the smart community domains use

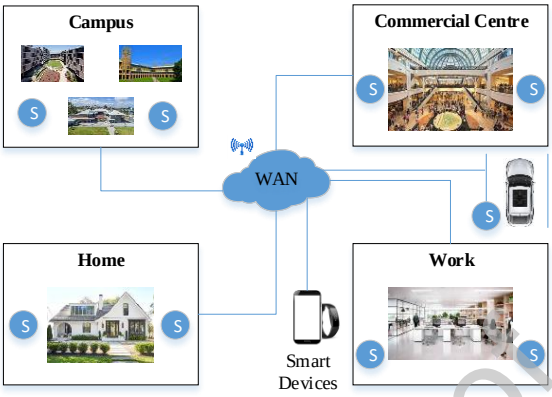

Figure 1: A simple network architecture of smart communities.

their sensing devices (denoted as S) to gather and exchange data. Other smart devices such as smartphones and wearables can also receive, monitor, and control day-to-day activities in the smart community. It is very important to note that the communications between all the devices take place via public channels. In this case, the devices are vulnerable to security attacks as the data exchanged and communication channels between the devices can be manipulated by attackers. Thus, it is necessary to authenticate every device (or asset) before having data access in smart communities. Furthermore, it should be noted that our network architecture focuses on the domains provided in Figure 1 and other unknown domains which cannot be anticipated may disrupt authentication and data access in the smart communities, thus, we assume that the network architecture cannot be utilised for the unknown domains to avoid disruption in the execution of our proposed scheme in this work.

3.2. Attack Model

We consider the well-known Dolev-Yao attack model [18] for describing the knowledge and capabilities of the attacker in this work. In the Dolev-Yao attack model, communications between smart communities' assets are performed over public channels, where an adversary can eavesdrop, intercept, modify, replay, inject, and delete data being transmitted. The attacker can impersonate genuine assets in smart communities. Furthermore, authentication can be evaded by the attacker if the valid credentials of the assets are exploited by the adversary.

4. SPrivAD

SPrivAD involves two entities, namely: i) RID, which represents the entity that registers and issues a cryptographic identity to every asset and further provides a discrete clock that increments in rounds or epochs and can be utilised by the assets for setting current time in the smart communities; and ii) asset, which represents the entity that is authenticating using the issued cryptographic identity as well as authenticates another asset before allowing legitimate access to its data. SPrivAD consists of two main phases, namely: i) registration phase, which represents the phase by which every asset obtains its cryptographic identity from the RID; and ii) mutually dependent authentication and data access phase (MDA), which represents the phase by which the asset proves its identity to another asset. We model the RID and assets as honest entities as long as they follow the protocols in the registration and MDA phases (see below).

4.1. Registration Phase

In this registration phase, an asset is issued a cryptographic identity ID by the RID, which digitally signs a cryptographic commitment provided by the asset to derive the ID via the MAR protocol (see below). At the end of this phase, the asset stores the ID and its random secrets in an encrypted format (to provide security and privacy guarantees) within a distributed database and further uses the stored information during the authentication phase. Before the registration phase, every AIM solution provider provides a list of its assets to the RID, which issues a unique name to every asset and stores the name in a set $Assets$ within the database, which also contains the identity of the RID, ID_{RID}. For assets that have been blocked from being registered, the RID stores their names (say, A and B for assets A and B, respectively) in a set $Assets_{blocked}$ within the database. We assume that A and B cannot be forged since they are provided by the RID.

4.1.1. MAR Protocol

We modify the EC-PCS to design the MAR protocol ($MAReg$), which is executed between a smart communities' asset (A) and RID (ID_{RID}) and the steps are provided in Figure 2. In this protocol, we introduce cryptographic algorithms such as 160 *bits* ECDSA digital signature $Sig_{pv_{rid}}(.)$, 128 *bits* Advanced Encryption Standard (AES) $Enc_k(.)$, 256 *bits* Secure Hash Function (SHA-2) $Hash(.)$, and 160 *bits* keyed-Hash Message Authentication Code

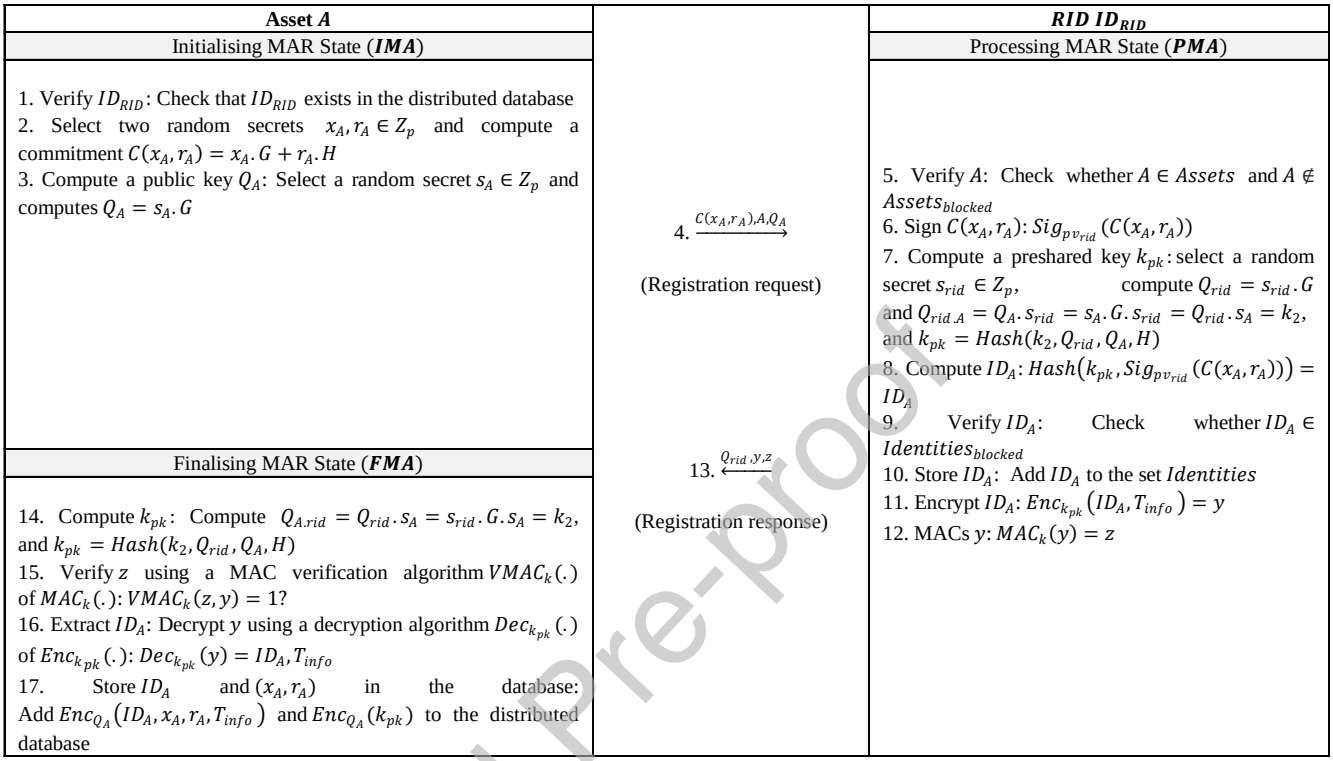

Figure 2: Mutually Authenticated Registration Protocol ($MAReg$).

(HMAC) $MAC_k(.)$ to provide security and privacy guarantees to the secrets provided by the asset and ID issued to the asset, where pv_{rid} is a private key of RID and k is a shared secret session key. Please note that the cryptographic algorithms support the construction of $MAReg$. For example, $MAC_k(.)$, $Enc_k(.)$, and $Hash(.)$ support asset authentication, data confidentiality, and data integrity, respectively. $MAReg$ uses three states as follows: i) initialising MAR state (IMA), which represents the state of sending an asset registration request to the RID; ii) processing MAR state (PMA), which represents the state of registering and issuing an ID to the asset; and iii) finalising MAR state (FMA), which represents the state of verifying ID and other information such as shared secret session key.

Upon the first initialization of $MAReg$ by an asset, the RID activates $MAReg$ and makes available the database for verification of ID_{RID}. When

RID receives an asset registration request that leverages the "perfectly hiding" property of the EC-PCS to hide the random secrets (x_A, r_A) and creates a commitment $C(x_A, r_A)$ in the IMA. RID verifies A, signs $C(x_A, r_A)$ using $Sig_{pv_{rid}}(.)$, creates a preshared key k_{pk} using $Hash(k_2, .)$, hashes $Sig_{pv_{rid}}$ $(C(x_A, r_A))$ and k_{pk} as $Hash(k_{pk}, Sig_{pv_{rid}}(C(x_A, r_A)))$, which becomes the cryptographic identity ID_A of A, where k_2 is an elliptic point curve, stores ID_A in a set $Identities$ within the database, and returns ID_A and other information to A at the end of PMA. Then, A computes k_{pk}, verifies received z, and securely stores $(ID_A, x_A, r_A, T_{info})$ and $Enc_{Q_A}(k_{pk})$ (in the database) in the FMA as depicted in Figure 2, where T_{info} represents information about the discrete clock.

4.1.2. Deactivation of Cryptographic Identity

According to Figure 2, if the RID cannot verify A in the $MAReg$, it will not register A. Additionally, if A cannot verify the received information from the RID (as a result of a violation of any steps such as computing a preshared key k in the PMA), it will request the RID to deactivate the issued ID_A by providing the information (such as Q_{rid} or (y, z)) that cannot be verified. If the information provided are genuine, RID stores ID_A in a set $Identities_{revoked}$, which represents a set containing issued identities that have been revoked. Thus, A can request for a new ID by executing $MAReg$, which is a repeatable procedure since every ID is unique, deactivatable, and revocable.

4.1.3. Benefits of the MAR Protocol

$MAReg$ achieves the following main goals to ensure the security and privacy of smart communities' assets: i) secrets are only known to the owner of the secrets and thus they are not revealed to the RID during registration, thereby guaranteeing the secrecy of the secrets and mitigating the risk of an adversary stealing the secrets; ii) a unique, deactivatable, and revocable cryptographic ID can be derived for every asset; iii) sensitive information are not transmitted in plaintext during the registration; and iv) secrets and cryptographic keys are not stored at the RID or assets.

4.2. MDA Phase

In this phase, two assets prove the ownership of their cryptographic identities and computational attributes of several cryptographic operations to

each other before deriving a shared secret session key for granting data access. This is achieved by proving in zero knowledge, the random secrets in the cryptographic identity of every asset and then proving the message size and number of executed steps of several cryptographic operations in this phase.

4.2.1. MDMFA Protocol

The MDMFA protocol ($MDMFA$), as shown in Figure 3, is executed between two assets, say A and B. The states of the protocol include initialising MDMFA state (IMD), processing MDMFA state (PMD), finalising MDMFA state (FMD), and key establishment and data access state (KED). Using the knowledge of cryptographic operations, the computational attributes, such as message size and number of executed steps are derived in $MDMFA$. The computational attributes are authentication factors that depend on each other and can be verified during authentication to support $MDMFA$ and mitigate authentication bypass in a timely manner. Hence, the attributes are required by A and B for authentication and both A and B cannot bypass the execution of cryptographic operations in the $MDMFA$. Without loss of generality, we note that every genuine asset has knowledge of cryptographic operations and maximum acceptable time of all computations in $MDMFA$.

In a state of $MDMFA$, S represents the message size while ST represents the number of executed steps by an asset. For example, a message that contains only an ID is 256 $bits$ and the number of executed steps of only this computation is one (1). Note that: i) the number of executed steps in a state at A is equal to the number of executed steps in the corresponding state at B to support the accuracy of ST in $MDMFA$; and ii) $MDMFA$ uses the states and computational attributes of cryptographic operations to mitigate the spread of malware (before an attacker's authentication requests are successfully accepted by the assets) and DDoS attacks during authentication by requiring assets to initialise and finish authentication with the support of computational attributes which are required for granting data access.

Upon the first initialization of $MDMFA$ by an asset, the RID activates $MDMFA$ and stores the default computation size of cryptographic operations, such as $Enc(.)$ (128 $bits$), $Hash(.)$ (256 $bits$), and $160-bit$ Pseudo-Random Function (PRF) $F(.)$, and artifacts such as ID (256 $bits$), timestamp T (32 $bits$), random number RN (32 $bits$), and every secret x in Z_p (80 $bits$) in a set $CryptoSize$ in the database. All registered assets can use this set for verifying the computation size of cryptographic operations and attributes. In

Asset A		Asset B
Initialising MDMFA State (IMD)		
1. Verify ID_B: Check that $ID_B \in Identities$ and $ID_B \notin Identities_{blocked}$ 2. Compute an MDMFA commitment d_A: $d_A = s_A.G + t_A.H$, where $s_A, t_A \in Z_p$ are random secrets 3. Compute a public key Q_{AP}: select a random secret $m_A \in Z_p$ and compute $Q_{AP} = m_A.G$ 4. Set $Msg_A(1) = (ID_A, ID_B, d_A, Q_{AP})$ size $S_{Msg_A(1)} \coloneqq S_{Msg_A(1)}$, where $S_{Msg_A(1)}$ represents the size of the message $Msg_A(1)$. 5. Set IMD_A state: $ST_{IMD_A} \coloneqq 5 = steps\ 1, 2, 3, 4, and\ 5$ which represents the number of executed steps by A in IMD at time T_{IMD_A}	6. $\xrightarrow{Msg_A(1), S_{Msg_A(1)}, ST_{IMD_A}, T_{IMD_A}}$	7. Verify ID_A: Check that $ID_A \in Identities$ and $ID_A \notin Identities_{blocked}$ 8. Compute a public key Q_{BP}: select a random secret $m_B \in Z_p$ and compute $Q_{BP} = m_B.G$ 9. Compute a preshared key k_{pk}: $k_{pk_1} = Q_{AP}.m_B = m_A.G.m_B = Q_{BP}.m_A$ and $k_{pk} = Hash(k_{pk_1}, Q_{AP}, Q_{BP})$ 10. Self-obtain $S_{Msg_A(1).B}$: Calculate $S_{Msg_A(1).B} = (ID_A, ID_B, k_{pk}, Q_{BP})$ and verify if $S_{Msg_A(1).B} = S_{Msg_A(1)}$ (via knowledge of cryptographic operations). 11. Self-obtain ST_{IMD_B}: Calculate $ST_{IMD_B} = 5 = steps\ 7, 8, 9, 10,$ and 11 and verify if $ST_{IMD_B} = ST_{IMD_A}$ at time T_{IMD_B}. Max. time to perform steps 7, 8, 9, 10, and 11 = T_{B_1} 12. Select a random challenge $e_B \in Z_p$
Processing MDMFA State (PMD)		
16. Compute k_{pk}: $k_{pk_2} = Q_{BP}.m_A = m_B.G.m_A = Q_{AP}.m_B$ and $k_{pk} = Hash(k_{pk_1}, Q_{AP}, Q_{BP})$ 17. Self-obtain ST_{PMD_A}: Calculate $ST_{PMD_A} = 2 = ST_{PMD_B}$ at time T_{PMD_A} 18. Obtain $Msg_B(1)$: Verify if $VMAC_{k_{pk}}(Msg_B(3), Msg_B(2)) = 1$ using k_{pk}; $Dec_{k_{pk}}(Msg_B(2)) = Msg_B(1)$	15. $\xleftarrow{Msg_B(4)}$	13. Create messages: $Msg_B(1) = (ID_A, ID_B, e_B, Q_{BP}, T_{IMD_B}, S_{Msg_A(1).B}, ST_{IMD_B}, ATF)$, where ATF are some artifacts that contain details of cryptographic operations used by ID_B. $Msg_B(2) = Enc_{k_{pk}}(Msg_B(1))$; $Msg_B(3) = MAC_{k_{pk}}(Msg_B(2))$ $Msg_B(4) = (Msg_B(3), Msg_B(2), Q_{BP})$ 14. Set PMD_B state: $ST_{PMD_B} \coloneqq 2$ at time T_{PMD_B}
Finalising MDMFA State (FMD)		
19. Verify if $\lvert T_{IMD_B} - T_{IMD_A} \rvert \leq T_{B_1}$ 20. Compute incremental preshared key k_{pk2}: $k_{pk2} = Hash(k_{pk}, ST_{IMD_B}, S_{Msg_A(1).B})$ Maximum time to perform step 20 = T_{A_1} 21. Set $Msg_A(2) = (ST_{PMD_A}, T_{PMD_A}, ATF)$ size $S_{Msg_A(2)} \coloneqq S_{Msg_A(2)}$ 22. Compute authentication values α_A and β_A: $\alpha_A = s_A + e_B.x_A$ and $\beta_A = t_A + e_B.r_A$ 23. Create messages: $Msg_A(3) = (Msg_A(2), \alpha_A, \beta_A)$; $Msg_A(4) = Enc_{k_{pk2}}(Msg_A(3))$; $Msg_A(5) = MAC_{k_{pk2}}(Msg_A(4))$; $Msg_A(6) = Msg_A(5), Msg_A(4))$	24. $\xrightarrow{Msg_A(6)}$	25. Compute incremental preshared key k_{pk2}: $k_{pk2} = Hash(k_{pk}, ST_{IMD_B}, S_{Msg_A(1).B})$ 26. Obtain $Msg_A(3)$: Verify if $VMAC_{k_{pk2}}(Msg_A(5), Msg_A(4)) = 1$; $Dec_{k_{pk2}}(Msg_A(4)) = Msg_A(3)$; $Msg_A(3) = (Msg_A(2), \alpha_A, \beta_A)$; $Msg_A(2) = (ST_{PMD_A}, T_{PMD_A}, ATF)$ 27. Verify if $ST_{PMD_B} = ST_{PMD_A}$ 28. Self-obtain $S_{Msg_A(2).B}$: Calculate $S_{Msg_A(2).B} = (ST_{PMD_B}, T_{PMD_B}, ATF)$ and verify if $S_{Msg_A(2).B} = S_{Msg_A(2)}$ 29. Verify if $\lvert T_{PMD_A} - T_{PMD_B} \rvert \leq T_{A_1}$ 30. Compute a verification value: vv: $\alpha_A.G + \beta_A.H$
Key Establishment and Data Access State (KED)		
31a. Derive a shared secret session key $k_{AB2} = (d_A + e_B.C)(k_{pk2})$ $k_{AB2} = ((s_A.G + t_A.H) + e_B.x_A.G + r_A.H)(k_{pk2})$ $k_{AB} = F(k_{AB2}, (S_{Msg_A(1).B}, ST_{PMD_A}, T_{PMD_A}), ID_A, ID_B)$ 32. Data access request using k_{AB} – Create messages: $Msg_A(7) = Enc_{k_{AB}}(data\ access\ request)$ $Msg_A(8) = MAC_{k_{AB}}(Msg_A(3))$	33. $\xrightarrow{Msg_A(8), Msg_A(7)}$ 35. $\xleftarrow{MACed\ data\ access\ result\ with\ a\ timestamp}$	31b. Derive a shared secret session key $k_{BA2} = (\alpha_A.G + \beta_A.H)(k_{pk2})$ $[k_{BA2} = (G(s_A + e_B.x_A) + H(t_A + e_B.r_A))(k_{pk2})]$ $k_{BA} = F(k_{AB2}, (S_{Msg_A(1).B}, ST_{PMD_A}, T_{PMD_A}), ID_A, ID_B)$ 34. Grant data access using k_{BA}: Verify if $VMAC_{k_{BA}}(Msg_A(8), Msg_A(7)) = 1$? (to check whether A is the owner of ID_A). Then, $Dec_{k_{BA}}(Msg_A(7)) = data\ access\ request$

Figure 3: Mutually Dependent Multi-Factor Authentication and Data Access Protocol ($MDMFA$).

$MDMFA$, A and B use their issued ID_A and ID_B, respectively, and computational attributes (based on the IA-ZKPK protocol) to authenticate each other and generate a shared secret session key k_{AB} to secure data access and further mitigate reflection attacks that are carried out on ZKPK protocol by an attacker or a malicious asset. In the reflection attack, the adversary tricks an asset into responding to its challenge and revealing the secret shared in the ZKPK protocol. Three major solutions have been proposed to mitigate this attack, including: (i) allowing an asset that initiates authentication to prove its identity before proceeding with the authentication [14], (ii) using different keys or types of challenge for the assets during authentication [14], and (iii) inserting an identifier in every response to a challenge to mitigate reflection attack [7]. However, even if these solutions were deployed, reflection attacks would still arise as follows:

- Reflection attack in the scenario where the initiator asset proves its identity before proceeding with authentication.

  This solution works for ZKPK protocols that use a single factor during authentication but may not be able to a mitigate reflection attack in the protocols that use multi factors during authentication. To see this, consider a scenario where an initiator asset and a responder asset want to use the ZKPK protocol for authentication based on a single factor. An adversary with username and password of the initiator asset can still carry out the reflection attack after using the username and password. Once the single factor has been compromised, the adversary can still trick the responder asset into revealing the secret in the ZKPK protocol. Therefore, it is preferable to use multi factors for the ZKPK protocol.

- Reflection attack in the scenario where different keys or types of challenge are utilized in the ZKPK protocol.

  Because using different keys or type of challenge introduces additional processes for either computing the keys or type of challenge, security of the keys or types of challenge is not guaranteed in the communication path as well as the sensitive information about the computing the keys or type of challenge is vulnerable to active man-in-the-middle (MITM) attack, which can be carried out by either an adversary or malicious asset to intercept and alter communications. This could lead to a reflection attack in which the adversary or malicious asset can reveal the secret in the ZKPK protocol after compromising the communications.

Therefore, it is preferable to have a secure in-built key establishment mechanism in the ZKPK protocol to help in securing messages and mitigating the active MITM attack.

- Inserting an identifier in every response to a challenge to mitigate reflection attack.

  In this solution, an active adversary or malicious asset can still replace the identifier and use it along with the response to carry out a reflection attack in the ZKPK protocol. Therefore, a ZKPK protocol with a capability of mitigating the replay of a response with an unknown identifier will help in protecting the protocol's secrets. Also, the verification of asset authenticity via secret keys that are supported by identifiers and only known to the assets can mitigate the reflection attack and further prevent impersonation attack on the ZKPK protocol.

The solution to protect against those reflection attacks associated with the ZKPK protocol is to integrate mutually dependent multi-factor authentication and data exchange mechanism with authenticated encryption to serve three purposes, namely: i) it establishes a multi-factor authentication for enhancing identity verification in the protocol; ii) it establishes a preshared key and an incremental preshared key for securing messages during communication and further mitigating active MITM attacks (note that the introduction of an incremental preshared key prevents the reuse of a preshared key whose sensitive information may have been leaked during communication); and iii) it helps in preventing replacement of identifier and replay of a response by allowing assets to verify received messages and computational attributes of the messages.

Remarks: An attacker can eavesdrop messages and based on the size and sequence can determine the number of steps that have been executed or count the messages that are exchanged by the assets. Additionally, an attacker can buy the same devices used by the parties and extract information about the size of the messages and steps. As we have already mentioned, preshared keys are used for securing messages (in an encrypted and/or MACed manners) during communication and the computational attributes support the MDMFA protocol and mitigate authentication bypass in a timely manner, i.e., the assets can securely verify the time interval of received and created messages against the maximum acceptable time of creating the messages, which rely on the computational attributes and preshared keys. Thus, the

success probability that the adversary can determine the size of the messages (in a timely manner), number of messages in an encrypted message, number of steps executed by the assets, or extract information about the size of the messages and steps in the MDMFA protocol is negligible.

According to the steps in $MDMFA$ as depicted in Figure 3, A sends an authentication and data access request message $Msg_A(1)$ along with the message size of $Msg_A(1)$, i.e., $S_{Msg_A(1)}$, the number of steps executed by A, i.e., ST_{IMD_A}, and timestamp T_{IMD_A} to B, which verifies the identity of A (ID_A) in step 7, verifies $S_{Msg_A(1)}$ in step 10, and verifies ST_{IMD_A} in step 11 for multi-factor identity verification. In the key establishment and data access state, A and B derive a shared secret session key k_{AB} using their cryptographic identities and secrets known to them. A and B then use k_{AB} to request and grant data access, respectively. Additionally, B uses k_{AB} to verify if A is the owner of ID_A. In the final step, a MACed data access result with a timestamp is issued to A. Note that k_{AB} as well as the data access expire based on the expiration of the timestamp issued by B. Once an existing data access expires, A may request a new authentication and data access by initiating $MDMFA$ with B and then B may grant another data access based on Figure 3. As a result of the $MDMFA$, a computationally bounded adversary's success probability to bypass authentication after crafting authentication request messages is negligible since the $MDMFA$ provides mutually dependent authentication with the support of the computational attributes of its cryptographic operations.

Remarks: i) to verify that the reflection attacks have not taken place during the $MDMFA$, A and B use multi factors such as ID, Q, and k_{pk} to perform authentication, preshared key k_{pk} and the incremental preshared key k_{pk2} to secure communications, and self-obtain computational attributes such as S and ST to support the protection of secrets such as k_{pk}, k_{pk2}, and k_{AB}; ii) the secure communications in $MDMFA$ mitigate active MITM attack and achieves forward secrecy and a new state-based forward secrecy, which provides assurances that session key in every new state in $MDMFA$ will not be compromised even if session key in a previous state is compromised, and thus, any future compromise of an asset's ID or secrets will not compromise the past preshared keys and shared secret session keys thereby preserving the secrecy of past communications in $MDMFA$; and iii) $MDMFA$ mitigates impersonation attack by a malicious asset or an adversary through the verification of asset identity and derivation of shared secret keys.

4.2.2. Revocation of a Cryptographic Identity

An asset A with ID_A can initiate revocation of another asset's ID (say, asset B with ID_B) based on the violation or misuse of cryptographic operations during the execution $MDMFA$/violation of any steps in the states of the $MDMFA$. In this case, ID_A tells RID to replace ID_B. If all the information provided by ID_A are genuine and $ID_A, ID_B \in Identities$, RID marks ID_B as revoked, stores $ID_B \in Identities_{blocked}$, and sends a revocation and replacement message $(Revoked(ID_B), RequestforReplacement(ID_B))$ to ID_A and ID_B. Asset B can request for a new identity using $MAReg$ before executing $MDMFA$ with any asset.

SPrivAD achieves the following main goals: i) a unique, deactivatable, and revocable cryptographic identity is derived from the asset's secrets; ii) the RID is not involved at authentication; and iii) it mitigates reflection attacks.

5. Security and Privacy Analyses

In this section, we formally define SPrivAD and analyse its security and privacy properties.

5.1. Modelling SPrivAD

Let asset A, asset B, and RID be the main entities in SPrivAD. Let DDb denote a distributed database. Before authentication and data access takes place, A and B register with the RID. After the registration is completed, A can authenticate with B in order to obtain data access.

Definition 1: We define two procedures of SPrivAD: $MAReg$ and $MDMFA$.

- *MAReg:* this is a protocol between A and RID as well as between B and RID across three states IMA, PMA, and FMA. If the registration between A and RID is successful, A obtains ID_A. Similarly, B obtains ID_B if its registration with RID is successful. Both ID_A and ID_B are securely stored in the DDb.

- *MDMFA:* this is a protocol between A and B across four states IMD, PMD, FMD and KED, where A initialises authentication and data access with B.

Definition 2: SPrivAD is a scheme with the following properties: Secure, Complete, Privacy-preserving, and Sound.

- *Secure:* A computationally bounded adversary's success probability in breaking the $MDMFA$ procedure and impersonating an honest asset is negligible.
- *Complete:* Two honest assets can successfully authenticate each other across the four states of the $MDMFA$.
- *Privacy-preserving:* Sensitive information is neither leaked nor revealed from the records of $MAReg$ and $MDMFA$. As long as A and B follow the $MDMFA$, RID does not learn about their communications.
- *Sound:* $MDMFA$ does not allow an honest asset to derive or prove false information. Furthermore, no other asset or adversary learns about the shared secret session key and access granted.

5.2. Analysing Security and Privacy Properties of SPrivAD

In this section, we analyse the security and privacy properties of SPrivAD which involves breaking and bypassing SPrivAD and leaking sensitive information from it, respectively.

5.2.1. Security Proof of SPrivAD

This proof involves the following: i) showing that breaking SPrivAD implies that the ECDLP can be solved since SPrivAD is constructed based on the IA-ZKPK protocol, which uses ECC; and ii) showing that bypassing SPrivAD implies that an attacker or dishonest asset has all the valid credentials and random secrets of an entity since SPrivAD is built to mitigate authentication bypass (and provide secure data access). Thus, if the ECDLP is hard and all the valid credentials and random secrets are not exposed, the adversary's success probability to break or bypass SPrivAD is negligible and the properties of SPrivAD as per Definitions 1, 2, 3, and 4 are satisfied.

In this proof, we define a function $RSECRET$ that derives two unique random secrets of 80 *bits* each, and a set $VCREDENTIALS$ that contains valid credentials in SPrivAD.

Theorem 1. *If $RSECRET$ is a random oracle for deriving two unique random secrets, $VCREDENTIALS$ is a set of valid credentials, and there*

exists an adversary I that successfully authenticates and gets data access with the MDMFA procedure with non-negligible probability, then I has a knowledge extractor that can extract secrets and solve the ECDLP, and also has all the valid credentials to bypass authentication with non-negligible probability.

Proof: Assume that such an adversary I exists, and another adversary J is given access to $RSECRET$, the Pedersen commitment of x and r, $VCREDENTIALS$, and $Hash(.)$. J sends ID_J and randomly chosen e and $x.G + r.H$ to I. Then, I is expected to validate ID_J and output IA-ZKPK of x and r against e.

We now simulate $RSECRET$ and $VCREDENTIALS$ and then present three cases of our security proof. To simulate $RSECRET$, J creates a set of tuples $RSSET$ and initialises it by selecting two random secrets (x_J, r_J) of length 80 *bits* each. J adds (x_J, r_J) to $RSSET$. To simulate $VCREDENTIALS$, J adds ID_J to $IDSET$.

- *Case 1:* When I queries $RSECRET$ on a secret x_J, J does the following: If there is a tuple (x_J, r_J) already in $RSSET$, it responds with $rs2$. Otherwise, it selects a random secret $r_J^{'}$, adds $(x_J, r_J^{'})$ to $RSSET$, and outputs $r_J^{'}$ to I. Similarly, when I queries $VCREDENTIALS$ on an identity ID_I, if there exists ID_I already in $IDSET$, it responds with ID_I. This simulation is similar to when J is not involved in the protocol. Thus, J succeeds in the simulation.

- *Case 2:* When I queries $Hash(.)$ with k_I and SC_I, J does the following: If $k_I = k_J$ and $SC_I = SC_J$, J outputs $fail$ with negligible probability because I does not know the preshared key of J and random secrets in SC_J. Otherwise, J queries $Hash(.)$ with k_I and SC_I, receives ID_I and sends it to I. Furthermore, when I queries $IDSET$ with ID_I, J verifies if $ID_I = ID_J$. If the verification succeeds, J outputs $fail$ with negligible probability because ID_I and ID_J are unique. Thus, authentication bypass is mitigated.

- *Case 3:* If J does not output $fail$ and I can create ZKPK of x and r and use ID and computational attributes, then by the properties of ZKPK, x and r can be produced by I using a knowledge extractor and ID can be exploited by I. In this case, J solves the ECDLP using the knowledge extractor and bypasses authentication using the valid credentials. This shows that if I succeeds, J also succeeds. However,

I does not exist if the ECDLP is hard and bypassing authentication is difficult.

5.2.2. Privacy Proof of SPrivAD

This proof shows that no sensitive information is leaked from the following four cases.

- *Case 1:* The privacy of ID directly follows the perfectly hiding property of EC-PCS as per Section 2.1.
- *Case 2:* The privacy of the transcripts of $MAReg$ and $MDMFA$ follows from the $MAReg$ that sensitive information of ID are not leaked or revealed during registration, from the zero-knowledge property of ZKPK and constructed IA-ZKPK used in the $MDMFA$ such that A and B reveals nothing during the execution of $MDMFA$, from the fact that no sensitive information is exchanged during the $MAReg$ and $MDMFA$, and from the setting that sensitive information is not stored on the entities even if the $MAReg$ and $MDMFA$ are broken with negligible probability.
- *Case 3:* The fact that no sensitive information is revealed during $MAReg$ and $MDMFA$, the transcripts of deactivating ID (during $MAReg$) and revoking ID (during $MDMFA$) cannot reveal any sensitive information.
- *Case 4:* The fact that we have adopted mutually dependent authentication and access control, which avoids going through the RID for any asset authentication and data access, the privacy of communications between A and B is guaranteed.

We can conclude that SPrivAD satisfies the four properties mentioned in Definition 2, and thus it is secure, complete, privacy-preserving, and sound with overwhelming probability. Please note that formal proof supporting our Theorem 1 is provided in Section 6.

6. Formal Security Validation using AVISPA

In this section, we simulate and validate the security of SPrivAD using the AVISPA tool [9], which is a well-known formal security verification tool

that has been successfully used for automated security analysis of cryptographic protocols in the smart communities [15], [16]. It uses the High-Level Protocol Specification Language (HLPSL) [17] for implementing the protocols and follows the Dolev-Yao attack model [18]. In our simulation, we use the On-the-fly Model-Checker (OFMC) [19] and Constraint Logic based Attack Searcher (CL-AtSe) [20] backend model checkers in AVISPA for fast detection of attacks on SPrivAD and proving the correctness of SPrivAD. These backends verify against replay and man-in-the-middle attacks. More details about our simulation are provided as follows:

- We specify the registration and MDA phases that are decomposed into three basic roles: asset A, asset B, and RID ID_{RID} (see Section 4), where the initial state of every asset is 0.
- We describe the composition roles that comprise sessions of the protocol, and environment representing the construction of the sessions with modelling of intruder knowledge according to SPrivAD as presented in Section 4 and analysed in Section 5. In the construction of the sessions, all random secrets and valid credentials in the registration and MDA phases are supported by the function $RSECRET$ and the set $VCREDENTIALS$ as presented in Theorem 1. EC-PCS, ZKPK, and IA-ZKPK are modelled in the construction based on SPrivAD (see Section 4).
- We model secrecy and authentication goals in SPrivAD (see Section 4).

Figure 4 depicts the simulation results from the OFMC and CL-AtSe backends. The results show that SPrivAD is resilient against replay and man-in-the-middle attacks with the bounded number of sessions and the derived shared secret session key and data access granted are safe from the Dolev-Yao attack model. Thus, the shared secret session key can be used for secure data access in the smart communities. The simulation results in Figure 4 have the following sections:

- SUMMARY: This indicates that SPrivAD is safe. In this case, SPrivAD properties are satisfied.
- DETAILS: This indicates the conditions where SPrivAD is safe. In this case, we have bounded number of sessions associated with SPrivAD.

OFMC Backend	CL-AtSe Backend
% OFMC	%CL-AtSe
SUMMARY	SUMMARY
SAFE	SAFE
DETAILS	DETAILS
BOUNDED_NUMBER_OF_SESSIONS	BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL	TYPED_MODEL
/papers/protocols/SPrivAD.if	PROTOCOL
GOAL	/papers/protocols/SPrivAD.if
as_specified	GOAL
BACKEND	As Specified
OFMC	BACKEND
COMMENTS	CL-AtSe
STATISTICS	STATISTICS
parseTime: 0.00s	Analysed : 2 states
searchTime: 0.01s	Reachable : 0 states
visitedNodes: 4 nodes	Translation: 0.01 seconds
depth: 2 plies	Computation: 0.00 seconds

Figure 4: Simulation results using AVISPA's OFMC and CL-AtSe backends.

- BACKEND: This indicates the backend used in analysing SPrivAD. In this case, we have the OFMC and CL-AtSe backends.
- GOAL: This indicates the goal of analysing SPrivAD as specified in Sections 4 and 5.
- PROTOCOL: This indicates the name of the protocol, i.e., SPrivAD.
- STATISTICS: This indicates the data on the analysis. For example, Reachable of 0 states shows that SPrivAD is secure as no attack was reachable.

Thus, no attack was found in the execution of SPrivAD.

7. Implementation and Experiments

In this section, we present the architecture of a prototype of SPrivAD and our experiments.

7.1. Architecture

The components of the prototype and the execution of $MAReg$ and $MDMFA$ are provided in Figure 5. Details of the components and communications among them are as follows.

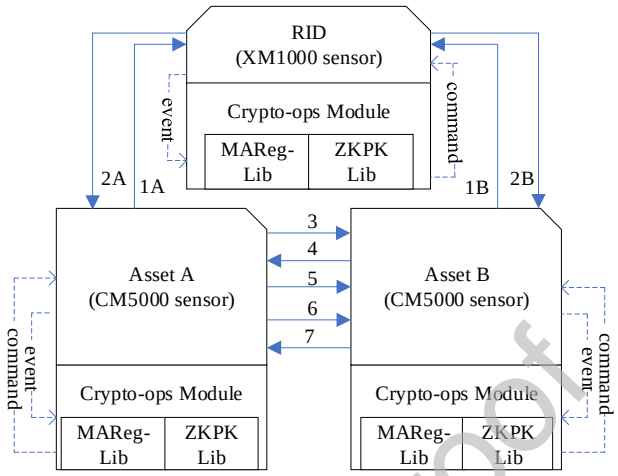

Figure 5: Architecture of SPrivAD.

7.1.1. Components

The components that represent the two main entities described in Section 4 are Tmote Sky sensor nodes [21], which are low power wireless sensor nodes that are widely used for data collection and simulating smart communities (see, e.g., [22]). The sensor nodes include CM5000 sensors and XM1000 sensors which are capable of monitoring and delivering critical information such as physical and environmental conditions in the smart communities. We use one of the CM5000 sensors as the RID to perform registration of two XM100 sensors, which are the assets that authenticate each other for data access.

We have modified the TinyECC library [23] to construct two libraries, SC-Crypto Lib and IA-ZKPK Lib, that contain different building blocks of the protocols and which can be used by the components. SC-MAReg Lib and IA-ZKPK Lib are used for the execution of $MAReg$ and $MDMFA$, respectively. They are stored in a Crypto-ops module that is attached to all the components for the implementation of cryptographic operations (via events) and return of implementation results (via commands) due to the limited physical resources such as 10 KB data memory. Furthermore, the CM5000 is equipped with a distributed database in a MacBook Pro machine.

7.1.2. Communications

The communications between the entities during the MAR and MDMFA protocols are illustrated in Figure 5. We use a series of numbered arrows to depict the flow of communications in the protocols. Steps 1A and 2A represent the MAR protocol executed between the asset A and RID. At the end of the MAR protocol, A and B obtain ID_A and ID_B, respectively. When A wants to initiate authentication and data access with B, it first uses the Crypto-ops module to implement cryptographic operations and then send a request to B in step 3 to obtain initial authentication and data access factors such as public key Q_{BP}, random challenge e_B, and ID_B. These factors are given to A by B in step 4 after using the Crypto-ops module. A then processes the response it received from B and sends a finalizing authentication and data access request to B in step 5. A and B can now derive a shared secret key k_{AB}. In step 6, A initiates a data access request with B using k_{AB}. If the zero-knowledge proof and attributes' computations are successful, B grants the request using k_{AB} in step 7. Note that steps 3 to 7 represent the MDMFA protocol as depicted in Figure 3.

7.2. Experiments

The main goal of our experiments is to evaluate the performance of SPrivAD using the MAR and MDMFA protocols as follows: i) evaluating the processing times of cryptographic operations in the protocols states to ensure the capabilities of components in handling requests; ii) evaluating the security of the experiments by measuring the execution times of the main steps of the protocols to identify any potential bottlenecks such as inability to compute a unique identity; and iii) measuring the End-to-End Delay (EED) of the protocols to show their impacts on smart communities' applications. We set up our experiments by connecting our sensors to a MacBook Pro machine with Ubuntu 16.04 TLS, 2.3GHz dual-core Intel Core i5 processor, 256 SSD, and 8GB memory. We use an operating system, TinyOS [24], to compile all the cryptographic operations, which are written in nesC language [25].

7.2.1. Evaluation of the States of the Protocols

We present the experimental evaluation of the states of the MAR and MDMFA protocols using the number of bits obtained from the protocols and then measured their processing times. The number of bits represents the size of messages, while the processing time represents the execution time

Table 2: Breakdown of Number of Bits and Approximate Processing Times of States of The MAR and MDMFA Protocols

MAR Protocol		
States	Number of Bits (in *bits*)	Processing Time (in *sec*)
Initialising MAR (IMA)	352	$\approx$ 2.082
Processing MAR (PMA)	1,440	$\approx$ 1.621
Finalising MAR (FMA)	1,728	$\approx$ 0.0326
MDMFA Protocol		
States	Number of Bits (in bits)	Processing Time (in sec)
Initialising MDMFA (IMD)	1,088	$\approx$ 3.1233
Processing MDMFA (PMD)	3,088	$\approx$ 0.0493
Finalising MDMFA (FMD)	2,400	$\approx$ 0.0765
Key Establishment and Data Access (KED)	1,152	$\approx$ 0.0520

of cryptographic operations. We measured the processing times of steps related to the states by executing the cryptographic algorithms in the states. The breakdown of the number of bits and processing times of the states is presented in Table 2.

From our results in Table 2, it can be observed that: i) the IMA and IMD states have the most demanding processing times in the MAR and MDMFA protocols, respectively, due the use of a large number of 160 *bits* EC point multiplication, which takes approximately 1.041 *sec* to be implemented on our CM5000 and XM1000 sensors. Performance benchmark [26] indicates that cryptographic protocols in IoT require cryptographic operations that provide at least asset authentication, data confidentiality, and data integrity since the performance requirements for these operations depend on specific software and hardware platforms. In our experiments, SPrivAD satisfies the above performance requirements by using $MAC_k(.)$, $Enc_k(.)$, and $Hash(.)$ for asset authentication, data confidentiality, and data integrity, respectively. Without loss of generality, we leave an investigation of performance benchmark based on processing times for future work since the main scope of this work is to provide secure and privacy-preserving authentication and data

Table 3: Simulation Parameters

Parameter	Description
Platform	Ubuntu 16.04 TLS
Network Scenarios	1 for $MAReg$ and 2 for $MDMFA$
Number of components	Two (2) for every scenario
Size of data packet S_p	$1,184$ $bits$ for scenario 1 and $3,824$ $bits$ for scenario 2
Packetization delay T_p	0.00704 $msec$ for scenario 1 and 0.03968 $msec$ for scenario 2
De-packetization delay T_d	0.01664 $msec$ for scenario 1 and 0.03680 $msec$ for scenario 2
Communication medium	Wi-Fi
Channel Model	P2P at 50 $Mbps$
Transport Layer	UDP

access solution for the smart communities.

Furthermore, we compare SPrivAD with existing related schemes of Wazid et al. [34], Kumar et al. [35], and Kyusuk et al. [36] in terms of the number of bits. Please note that: i) SPrivAD requires $2F(.) + Enc(.) + Dec(.) + 2HMAC(.) + 8RN = 1,512$ $bits$; ii) Wazid et al. [34] requires $22Hash(.) + 2Enc(.) + 2Dec(.) = 6,144$ $bits$; iii) Kumar et al. [35] requires $2Hash(.) + 3HMAC(.) + Enc(.) + Dec(.) = 1,248$ $bits$; and iv) Kyusuk et al. [36] requires $5Hash(.) + 7HMAC(.) + 4Enc(.) + 4Dec(.) = 6,144$ $bits$. It can be seen that SPrivAD requires the lowest number of bits, and thus, it is more efficient than the existing related schemes.

7.2.2. EED

This represents the average time taken by the messages to arrive at the responder component from the initiator component. We simulated the EED of the MAR and MDMFA protocols using the widely accepted network simulation tool, Network Simulator 3 (NS-3) [27], on Ubuntu 16.04 TLS platform to show the impact of the protocols.

The EED can be formulated as $\sum_{i=1}^{n} T_{p_i} + \sum_{j=1}^{m} T_{g_j} + \sum_{k=1}^{n} T_{d_k}$, where T_{p_i} is the packetization delay (i.e., the time taken to prepare the packet), T_{g_j} is the propagation delay (i.e., the time taken by the network to deliver the

packet), and T_{d_k} is the de-packetization delay. Details of the parameters used in our NS3 simulation are provided in Table 3. Apart from these parameters, other standard parameters such as flow monitor (for measuring network performance) in NS3 are used in our simulation. The bit lengths of the random number/timestamp, name, identity, digital signature, commitment, and hash function are taken as 32 $bits$, 32 $bits$, 256 $bits$, 160 $bits$, 160 $bits$, and 256 $bits$, respectively. In the MAR protocol, i.e., scenario 1, two different types of messages are used, namely, $(C(x_A, r_A), A, Q_A)$ and (Q_{rid}, y, z) which are of sizes 352 $bits$ and 832 $bits$, respectively. On the other hand, five different types of messages, namely, $(Msg_A(1), S_{Msg_A}(1), ST_{IMD_A}, T_{IMD_A})$, $Msg_B(4)$, $Msg_A(6)$, $(Msg_A(8), Msg_A(7))$, and $(MACed\ data)$ of sizes 928 $bits$, $1,424$ $bits$, 640 $bits$, 416 $bits$, and 416 $bits$, respectively, have been used in the MDMFA protocol, i.e., scenario 2 (see Figure 3 for more details on the messages). The EED values are given as ≈ 7.4238 $msec$ and ≈ 9.8476 $msec$ for scenarios 1 and 2, respectively. These results show that the EED value increases as the number of transmitted/received packets increase across the network. Furthermore, the EED values are below our maximum 20 $msec$ latency target.

8. SCABA

In this section, we present SCABA and propose a protection uing SPrivAD.

8.1. Definition of SCABA

We define SCABA and relate it to authentication and secure access protocols in smart communities, and propose a solution using SPrivAD. We say that an asset A is honest, i.e., did not bypass authentication, if and only if all of A's actions conform to the execution of SPrivAD.

Definition 5. *A SCABA is an attack in which a dishonest asset A or an attacker exploits one or more valid credentials of an asset B to provide a responder asset F with the valid credentials of B, thereby bypassing authentication to access data in the smart communities.*

A protocol is then said to be vulnerable to SCABA if it allows A to bypass authentication. In the context of authentication and secure access protocols in smart communities, some valid credentials about an asset are passwords and security tokens or touch identity; hence SCABA involves convincing F that A is B. In a typical SCABA on an authentication and data access

protocol, a dishonest asset A convinces a responder asset F that B has executed an authentication and data access protocol with F, whereas the protocol has been executed by A. This is performed without the cooperation of B. This type of attack can be carried out by allowing A to complete the authentication and data access protocol using the valid credentials of B.

Example 1 (SCABA on Authentication in Ruckus Cloudpath Enrollment System). *Figure 6 depicts a SCABA on the authentication in Ruckus Cloudpath Enrollment System, which uses both Protected Extensible Authentication Protocol (PEAP) and Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) IEEE802.1X authentication scheme to let users register their devices through a self-service portal and gain access to network resources using their existing login credentials. The EAP-TLS relies on TLS to establish a secure communication channel. Note that: i) when the PEAP method is selected during authentication, the RADIUS server issues a certificate, which is verified using the Certificate Authority (CA) certificate on the asset, and the asset is required to provide username and password in order to authenticate itself to the server; ii) authentication using the EAP-TLS method requires both the RADIUS server and asset to issue their certificates to each other in this case, the server certificate is verified using the CA certificate on the user, while the certificate from the asset is verified using the CA certificate on the server; iii) since CAs are in charge of issuing certificates, these authorities can be compromised by attackers and they present a single point of failure; iv) certificates are vulnerable to MITM attacks as they can be intercepted during communication; and v) storing certificates on assets make the assets prime targets for the attackers as the certificates are not adequately protected. Thus, we consider certificates as valid credentials that can also be exploited by attackers in this work.*

In the SCABA as shown in Figure 6, the authenticator F thinks he is communicating with B, where B is an honest asset. When the dishonest asset A tries to prove its identity using ID_B, F, allows the protocol to proceed as normal. Note that P has no evidence that the EAP-Response identity packet it received was indeed sent by A since A is using B's valid credentials such as passwords and keys. Upon receiving the RADIUS Access-Accept packet from the authentication server in step 9, F will falsely conclude that B sent the EAP Response packet. Thus, F assumes that A is B, even though in reality, this protocol was performed by A thereby bypassing authentication to access resources such as data.

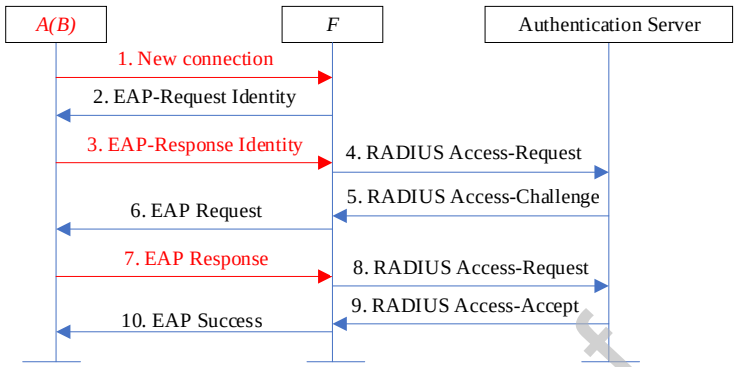

Figure 6: SCABA on Authentication in Ruckus Cloudpath Enrollment System.

Furthermore, we present another example scenario in which SCABA pose a threat to Duo MFA scheme that is widely used for personal accounts authentication in smart communities.

Example 2 (SCABA on Duo MFA scheme). *Consider the real-world scenario depicted in Figure 7, in which several people are situated in a secure smart communities facility. In this facility, authorized personnel can access data using Duo MFA scheme. As an added security mechanism, users can choose any authentication methods that are allowed in the smart communities. These methods, such as duo push (i.e., username and password), call me, and enter a passcode use login request, phone call-back, and passcode (generated by a hardware token or provided by an administrator), respectively, as authentication factors for identity verification. These factors are used every time a user wants to access Duo-protected resources from their devices.*

Assume that an attacker I has managed to get hold of a user B valid credentials and/or device. I can exploit the credentials to execute SCABA. The Duo MFA scheme now believes that I is B and then I is granted access to the Duo-protected resources thereby making the scheme vulnerable to SCABA.

8.2. Protecting Against SCABA

We have seen some authentication and secure access protocols that are vulnerable to SCABA, and we now present how to repair them. In SCABA, a dishonest asset or an attacker exploits an honest asset's valid credentials

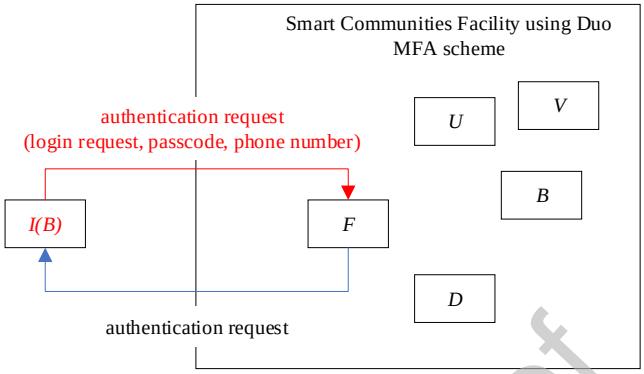

Figure 7: Real-world scenario for SCABA on Duo MFA scheme.

during authentication. The attacker can construct his messages and does not have to follow the execution of the protocols, which can be exploited by the attacker at any step. Therefore, in the solution we propose, we ensure that the secrets of an asset are neither transmitted nor revealed to another asset (say, asset F) so that the attacker cannot exploit the secrets of the asset in its communication with F.

Using SPrivAD, we discuss our three possible integrated solutions as follows: i) deriving a unique, deactivatable, and revocable identity using cryptographic commitment with random secrets during registration; ii) deriving a shared secret session key using the (features of the) identity and computational attributes of cryptographic operations; and iii) verifying the identity using the shared secret session key.

Solution 1: Deriving a unique, deactivatable, and revocable identity this solution ensures that the identity of every asset is distinguishable by explicitly including random secrets in a cryptographic commitment that is created by the asset and used in deriving the identity, which can be deactivated and revoked if any cryptographic operations is misused during registration and authentication, respectively. An example of deriving a unique, deactivatable, and revocable identity (for an asset A by RID) is via $MAReg$, which derives an identity in its processing MAR state.

- Asset A Compute $C(x_A, r_A) : C(x_A, r_A) = x_A.G + r_A.H$, where x_A, r_A are random secrets and $C(x_A, r_A)$ is a cryptographic commitment
- RID Derive ID_A: $Hash(k_{pk}, Sig_{pv_{rid}}(C(x_A, r_A))) = ID_A$, where ID_A

is an identity of A used during authentication.

Solution 2: Deriving a shared secret session key This solution uses the identity features (such as cryptographic commitment), and computational attributes of cryptographic operations to derive a shared secret session key during authentication. This solution relies on the fact that sensitive information about the key is secure during authentication and only honest assets can compute the shared secret session key. Thus, only the assets executing a protocol can derive the shared secret session key and then use it to construct messages. An example of deriving a shared secret session key is via $MDMFA$.

- Asset A Derive $k_{AB} : k_{AB2} = (d_A + e_B.C)(k_{pk2})$ and $k_{AB} = F(k_{AB2}, (S_{Msg_A(1).B}, ST_{PMD_A}, T_{PMD_A}), ID_A, ID_B)$, where k_{AB} is a shared secret session key, d_A is an MDMFA commitment from A, e_B is a random challenge, C is a cryptographic commitment of A during its registration, k_{pk2} is an incremental preshared key, ST_{PMD_A} is the number of several executed steps in PMD at A, T_{PMD_A} is a timestamp of setting ST_{PMD_A}, $S_{Msg_A(1).B}$ is the message size of $Msg_A(1)$ in IMD at B, and ID_B is an identity of B.

- Asset B Derive $k_{AB} : k_{BA2} = (\alpha_A.G + \beta_A.H)(k_{pk2})$ and $k_{BA} = F(k_{AB2}, (S_{Msg_A(1).B}, ST_{PMD_A}, T_{PMD_A}), ID_A, ID_B)$, where α_A and β_A are authentication values from A and $k_{BA} = k_{AB}$.

Solution 3: Verifying the identity The third solution uses the shared secret session key k_{BA} to verify the identity of an asset (via $Mac_{k_{BA}}(.)$) before granting data access after authentication. This solution relies on the fact that k_{BA} is used by a responder asset B to verify and show that an initiator asset A is an owner of ID_A. Note that if any cryptographic operations during authentication are misused, ID_A can be revoked (see Section 4.2 for more details about revoking a cryptographic identity). Until a shared secret session key is computed, the ownership of ID_A cannot be fully proven, and an attacker or dishonest asset does not have the (unrevealed) secrets of A. Thus, B can check that the claimed identity (ID_A) of A corresponds to its verification using $Mac_{k_{BA}}(.)$ since k_{AB} at A is equal to k_{BA} at B.

Hence, these solutions can be integrated and applied to repair the Ruckus Cloudpath Enrollment System and Duo MFA scheme for SCABA mitigation.

9. Case Studies

In this section, we carry out some case studies to strengthen the security of Ruckus Cloudpath Enrollment System and Duo MFA schemes in smart campus scenarios.

9.1. Ruckus Cloudpath Enrollment System

The Ruckus Cloudpath Enrollment System uses an in-built and comprehensive CA that allows users to create and manage their public key infrastructure, which yields every user's login credentials. The potential drawbacks of this protocol are as follows: i) impersonating the user in the case of using PEAP for authentication and secure access since the user can be impersonated after its password is exploited; ii) using existing login credentials for authentication and secure access exposes the system to security attacks such as phishing attack; iii) limitations of the CA as presented in Section 7; and iv) leaking sensitive information; and v) breaking the enrolment and authentication procedures.

According to a case study for the Ruckus Cloudpath Enrollment System for providing user authentication in Evangel University [28], Ruckus-patented Dynamic Pre-shared Key (DPSK) technology was used by smart devices for authentication and secure access via a unique pre-shared key generated by a Ruckus controller. The generation of pre-shared keys for the devices by the controller leads to pre-shared key secrecy concern thereby leaking sensitive information about a user's authentication and data access. Thus, the security and privacy of the users cannot be guaranteed. In this paper, we use the $MAReg$ and $MDMFA$ in SPrivAD to strengthen the security of Ruckus Cloudpath Enrollment System.

Let CA be a CA certificate and $CERTS$ be a set of certificates stored by every user and server in the Ruckus Cloudpath Enrollment System using EAP-TLS. The following theorem states that authentication and secure access via EAP-TLS in the system is vulnerable to compromised and privacy attacks.

Theorem 2. *If CA is a CA certificate, $ASReq$ is an authentication and secure access request, S is a server with a set of certificates $CERTS_S$, U is a user with a set of certificates $CERTS_U$, and both S and U issue their certificates $CERT_{S.1}$ and $CERT_{U.1}$ respectively, to each other for authentication and secure access, we say that $CERT_{U.1}$ and $CERT_U$ can be compromised by an adversary I and further leak sensitive information about U and CA in the Ruckus Cloudpath Enrollment System with overwhelming probability.*

Proof: We assume that such adversary I exists and has access to $CERTS_U$ since I can exploit U. Then, I sends $CERT_{U.1}$ to S in order to be authenticated and granted access. Then, S uses CA in $CERTS_S$ to verify $CERT_{U.1}$. As $CERT_{U.1}$ is stored in $CERT_U$, S authenticates and grants access to I. It is straightforward to see that I has compromised $CERT_{U.1}$ and $CERT_U$ and sensitive information about U and CA have been leaked and thus the authentication and secure access procedure has been broken. We can now say that authentication and secure access via EAP-TLS in Ruckus Cloudpath Enrollment System does not satisfy the secure, privacy-preserving, complete, and sound properties of SPrivAD.

To fix the above problems, we propose a variant of the authentication and secure access via EAP-TLS in the Ruckus Cloudpath Enrollment System, i.e., SPrivAD-based Ruckus Cloudpath Enrollment System, which uses SPrivAD's $MAReg$ and $MDMFA$ procedures in the Ruckus Cloudpath Enrollment System so that we can replace the use of certificates and introduce EC-PCS and IA-ZKPK protocol since $MAReg$ guarantees that random secrets of a user are not revealed during registration to mitigate the problem of issuing sensitive information (such as CA certificate) and $MDMFA$ guarantees that the user cannot be successfully compromised (since it does not store its random secrets on its memory). The following theorem states that our variant of the authentication and secure access via EAP-TLS in the Ruckus Cloudpath Enrollment System mitigates compromised and privacy attacks.

Theorem 3. *Let $MAReg$ and $MDMFA$ be the MAR and MDMFA procedures, respectively, modelling the SPrivAD-based Ruckus Cloudpath Enrollment System that allows a user and server to authenticate each other. Let U, S, and I be a user, server, and adversary, respectively, where U and S register with $MAReg$, U, S, and I authenticate with $MDMFA$, and $MAReg$ and $MDMFA$ use a set $RSECRET$ as a random oracle. Then, I has a negligible success probability in compromising U, leaking any sensitive information, and breaking $MAReg$ and $MDMFA$.*

Proof: This statement follows easily from Theorem 1 and the fact that U and S are successfully registered separately via $MAReg$ and authenticate each other with $MDMFA$, and I cannot violate the procedures with overwhelming probability.

While we opted for an authentication and secure access via EAP-TLS in the Ruckus Cloudpath Enrollment System for simplicity, it is trivial to analyse authentication and secure access via PEAP in the system as such

analysis requires only the server to issue a certificate while the user provides a username and password, which can be exploited according to SCABA.

9.2. Duo MFA scheme

The Duo MFA scheme uses two-factor authentication to verify the identity of users before granting access to applications and data at the duo resources component in the smart campus. Its authentication factors such as duo push, phone number, or passcode are used for identity verification. The potential drawbacks of this protocol are as follows: i) impersonating the duo resources component; ii) leaking sensitive information; and iii) breaking the MFA procedure. As the identity of the duo resources component is not verified by the scheme and users can enter sensitive information, these could lead to impersonation and privacy attacks, respectively. In the impersonation attack, an adversary can impersonate an honest user and deceive other users. In a privacy attack, the users can unknowingly provide their passwords to the adversary.

According to two case studies of Duo MFA scheme for mitigating the impact of users' weak passwords at the University of Michigan Departmental Computing Organization [29] as well as providing trusted MFA solution to improve detection of potential compromised accounts at Duke University [30], Duo MFA scheme was deployed between the users and systems (such as portals, servers, or web applications) before access to the systems is granted. Lack of authenticating the systems by the users before the users provide their secret information makes the scheme vulnerable to impersonation attack at the systems' ends and privacy attack at the users' ends. Thus, the security and privacy of the systems and users cannot be guaranteed. In this paper, we use the $MDMFA$ in SPrivAD to strengthen the security and privacy of Duo MFA scheme.

Let $IDEN$ be a set of identities which is used for verifying the identity of every system and $USERS$ be a set of users for verifying the identity and attributes of every user. Let $AReq$ be an authentication and data access request. The following theorem states that the Duo MFA scheme is vulnerable to impersonation and privacy attacks at every system and user, respectively.

Theorem 4. *If $IDEN$ is a set of systems' identities, $USERS$ is a set of users' identities and attributes, $AReq$ is an authentication and data access request, U is a user, and there exists a system S that authenticates and grants data access to U, we say that S can be impersonated by an adversary I as*

$I(S)$ and U can leak sensitive information to I in Duo MFA scheme with overwhelming probability.

Proof: Assume that such an adversary I exists and is given access to $USERS$ since it can impersonate S. U sends $AReq$ to $I(S)$ for identity verification. $I(S)$ can now verify U since it has access to $USERS$. As U has no access to $IDEN$ since the Duo MFA scheme only considers S authenticating U, it can be seen that U has leaked its sensitive information to I, who impersonated S, and the authentication procedure has been broken via impersonation attack by I and privacy attack at U. Thus, the Duo MFA scheme does not satisfy the secure, privacy-preserving, and complete properties of SPrivAD.

To fix the above problem, we propose our variant of the Duo MFA scheme, i.e., SPrivAD-based Duo MFA scheme, which uses SPrivAD's $MDMFA$ in the Duo MFA scheme so that we can analyse the Duo MFA scheme using IA-ZKPK protocol and since $MDMFA$ provides mutual multi-factor authentication and data access that is dependent on the attributes of two users and prevents leakage of sensitive information, system impersonation, and breaking of authentication procedure. The following theorem states that our variant of the Duo MFA scheme provides mutual multi-factor authentication and data access.

Theorem 5. *Let $MDMFA$ be the mutual multi-factor authentication and data access procedure modelling the SPrivAD-based Duo MFA scheme that allows mutual authentication. Let U, S, and I be user, system, and adversary, respectively which authenticate with $MDMFA$. Then, I's success probability in leaking any sensitive information, impersonating S, and breaking $MDMFA$ is negligible.*

Proof: The proof of this theorem follows easily from Theorem 1 and the fact that U and S successfully authenticate each other before data access with $MDMFA$, which cannot be violated by I with overwhelming probability.

10. Related Work

Providing a secure authentication and data access mechanism in smart communities is a rising area of research. Public-key authentication is a well-known authentication method used for such a mechanism. A public key authentication mechanism relies on cryptographic algorithms to generate a key pair, i.e., a public key Q and its corresponding private key d. Q is made

public while d is kept as a secret usually stored on an asset. These keys are used as authentication factors during authentication and data access via public key authentication. However, there are issues when using a public-key authentication mechanism to provide authentication and data access, namely exploitation of stored sensitive information and reuse of private keys and passcodes. The reason is that in public-key authentication the storage and reuse of private keys pose security and privacy issues due to asset compromised attacks and leakage of sensitive information, respectively. Addressing such security and privacy issues requires additional measures such as multi-factor authentication and key derivation.

Several proposed schemes to authentication and data access have focused on providing security of assets in the smart communities (see, e.g., [10], [11], [15], [31], [32], [33], [34], [37], [38]). The schemes address the authentication problem of verifying if an asset is who he/she claims to be. On the other hand, the privacy of the asset is not well supported during authentication and data access. Zheng et al. [15] proposed a mutual authentication protocol for the smart campus. However, the protocol neither provides confidentiality nor achieve mutual authentication as an encryption algorithm is required for confidentiality and verification of identity is needed for mutual authentication. Hence the protocol is vulnerable to compromised attacks, impersonation attacks, and leakage of sensitive information (such as a timestamp) that support the protocol execution. Unlike the protocol, SPrivAD provides confidentiality, uses IA-ZKPK protocol to support multi-factor authentication, and prevents compromised attacks, impersonation attacks, and sensitive information leakage during authentication and data access in the smart campus.

Safkhani et al. [31] discovered that Zheng et al.'s protocol [15] is vulnerable to replay attack and then addressed this weakness by proposing a secure authentication protocol for information system and smart campus. However, Safkhani et al.'s protocol [31]] is vulnerable to impersonation and compromised attacks as follows: i) a reader can be compromised since no secret information of the reader is required during authentication; and ii) a compromised reader can initiate the protocol with a genuine tag and server and carry out impersonation attack since the protocol does not support MDMFA between two nodes. In [31], assurance on the verification of the identity is not considered during authentication. Furthermore, sensitive information that supports the protocol is leaked during the protocol execution and the compromised reader can carry out an active MITM attack to intercept and

alter communications in the protocol. Note that the protocols [15], [31] do not provide SCABA mitigation. Thus, our scheme uses the $MAReg$ and $MDMFA$ procedures to address the weaknesses of the protocols and provide SCABA mitigation.

Ye et al. [32] presented an IoT system-level authentication scheme that can be implemented in a smart campus. In [28], two nodes rely on a preshared credential to compute a shared secret key for securing data during authentication. While the preshared credential introduces extra computational and communication overheads in the scheme, it can also be intercepted by an adversary or be made available to a compromised or malicious node. In this case, the sensitive information that supports the shared secret key computation and execution of the scheme has been leaked and mutual authentication has been compromised. Thus, the scheme is vulnerable to compromised and impersonation attacks, and an attacker can interrupt and alter communications in the scheme thereby carrying out an active man-in-the-middle attack on the scheme. Furthermore, the scheme does not mitigate SCABA. In our scheme, we use the MDMFA procedure to address the preshared credential and shared secret key weaknesses and provide SCABA mitigation.

Li et al. [33] proposed a lightweight mutual authentication protocol based on a public-key encryption scheme for IoT and its applications such as smart campus. The protocol stores and utilises existing private keys for mutual authentication make the protocol vulnerable to compromised attacks and leakage of private keys, and the storage capabilities of the resource-constrained IoT devices may not be adequate for storing the keys. In our scheme, we overcome these security, privacy, and storage weaknesses by providing MDMFA procedure, preventing the revealing of sensitive information, and preventing the storage of sensitive information on the devices, respectively. Furthermore, the protocol does not mitigate SCABA when applied to the smart communities and thus we provide SCABA mitigation in our scheme via the MDMFA procedure.

Wazid et al. [34] proposed a secure remote user authentication scheme for a smart home network to enable only authorised users to have access to smart devices, where smart home is an application in smart communities. However, the security of the scheme relies on a tamper-resistant device, which makes the scheme suffer from compromised, impersonation, and privacy attacks as an attacker can at any time successfully break the security of the device. Furthermore, since the authority in the scheme is responsible for generating a unique secret key and storing the key and other information into the memory

of smart devices, this leads to a lack of secret key secrecy and the attacker can further compromise these devices to extract the stored secret keys. Note that the scheme [34] does not mitigate SCABA and thus our scheme can be applied to overcome the weaknesses in [34].

Saud [37] proposed a user authentication scheme for smart e-governance applications in smart cities. However, the scheme relies on a trusted authority that issues confidential values needed to login into the smart cities. Ali et al. [38] proposed a lightweight authentication mechanism that enhances the security of smart city surveillance. However, the mechanism relies on a server during authentication. This reliance presents a single point of failure in the mechanism. Thus, the scheme [37] and mechanism [38] neither prevent leakage of sensitive information nor mitigate SCABA. Our scheme can be applied to enhance the privacy of sensitive information and mitigate SCABA in [37] and [38].

Ruckus Networks [39] provided Ruckus Cloudpath Enrollment System [10] for authentication and secure access. However, the system relies on certificates issued by CA and/or passwords which has security, privacy, and performance concerns such as compromised attack and single point of failure, leakage of sensitive information, and computational and communication overheads associated with issuing and revoking certificates, respectively. Furthermore, the system does not cover SCABA prevention. Our scheme uses the features of the MDMFA procedure (as presented in Section 4.2) to prevent SCABA and introduces a variant of the system to prevent compromised attacks and sensitive information leakage and further restricts the use of certificates to mitigate the performance concern of the system.

Duo [40] designed an MFA scheme [11] that easily enrols, and grants users access to resources in a smart campus. However, the scheme does not provide mutual authentication and it is vulnerable to impersonation attacks. Additionally, while the scheme neither prevent sensitive information leakage nor prevents SCABA, our scheme uses the features of the MDMFA procedure (as presented in Section 4.2) to prevent SCABA and introduces a variant of the Duo MFA scheme with the MDMFA procedure that applies IA-ZKPK protocol to provide mutual (multi-factor) authentication and generates a shared secret session key to overcome the above security and privacy challenges. A summary of the functionalities and limitations of all the related schemes is provided in Table 4. The various notations used in this table are as follows: F_1, F_2, F_3, F_4, F_5, and F_6 denote whether a scheme mitigates compromised attack, mitigates impersonation attack, mitigates active man-in-the-middle

Table 4: Functionalities and Limitations of Related Schemes

Schemes	F_1	F_2	F_3	F_4	F_5	F_6
[10]	×	×	✓	✓	×	×
[11]	×	×	✓	×	×	×
[15]	×	×	✓	×	×	×
[31]	×	×	×	×	×	×
[32]	×	×	×	×	×	×
[33]	×	✓	✓	✓	×	×
[34]	×	×	✓	✓	×	×
[37]	×	✓	✓	✓	×	×
[38]	×	✓	✓	✓	×	×
Ours	✓	✓	✓	✓	✓	✓

attack, supports secure mutual authentication, mitigates SCABA, and mitigates privacy attack, respectively. It can be observed that our scheme offers more security and privacy benefits compared to the other related schemes.

11. Conclusion and Future Work

In this paper, we presented an authentication and data access scheme, SPrivAD, in which smart communities' assets authenticate each other before data access using their cryptographic identities. Our scheme mainly focuses on security and privacy-preserving authentication and data access in order to overcome drawbacks of the existing public key authentication solutions in the smart communities. At the same time, our scheme provides strong and mutually dependent assurance about the ownership of a cryptographic identity. Three significant contributions of our scheme are: i) using EC-PCS in a new MAR protocol to securely derive a unique, deactivatable, and revocable cryptographic identity, which is based on the unrevealed secrets of an asset; ii) introducing IA-ZKPK protocol that uses computational attributes of cryptographic operations to support mutual authentication and mitigation of reflection attacks on ZKPK protocol in a new MDMFA protocol, which provides mutually dependent authentication and secure data access between smart communities' assets; and iii) introducing and protecting SCABA on authentication and secure access schemes in the smart communities.

We evaluated the performance of our scheme by performing some experiments and implementing a prototype of the scheme on Tmote Sky sensors to prove the feasibility of the scheme in terms of robustness in security and performance. The results show that our scheme meets the 20 *msec* latency target of smart communities' applications. The security and privacy analyses of our scheme show that it is secure and privacy-preserving, respectively. We illustrated the usefulness of our scheme in two real-world authentication and secure access schemes, Ruckus Cloudpath Enrollment System and Duo MFA, in the smart campus environments. We uncovered some weaknesses of the schemes and applied our scheme to provide some security and privacy guarantees. In future work, we will extend our scheme to support asset anonymity and facilitate security risk assessment and implement it as a service in the smart communities.

References

[1] Cisco. Cisco Smart Cities. [Online]. Available: https://www.cisco.com/c/en_au/solutions/industries/smart-connected-communities.html

[2] Amazon. Amazon Smart Communities. [Online]. Available: https://www.amazon.com/smart-home-communities/b?ie=UTF8&node=17726333011

[3] Ruckus Networks. Smart Cities. [Online]. Available: https://www.ruckuswireless.com/solutions/smart-cities

[4] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, et al., "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," IEEE Communications Magazine, vol. 55, pp. 70-78, 2017.

[5] Microchip Technology Inc. Trusted and Secure Authentication with ATECC508A for AWS IoT. [Online]. Available: https://www.microchip.com/design-centers/security-ics/cryptoauthentication/cloud-authentication/aws-iot-atecc508a

[6] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in Advances in Cryptology-CRYPTO'86, 1986, pp. 186-194.

[7] CAPEC. CAPEC-90: Reflection Attack in Authentication Protocol. [Online]. Available: https://capec.mitre.org/data/definitions/90.html

[8] B. Frana, "Homomorphic mini-blockchain scheme," ed: April, 2015.

[9] AVISPA. Automation Validation of Internet Security Protocols and Applications. [Online]. Available: http://www.avispa-project.org/

[10] Ruckus Networks. Cloudpath Enrollment System. [Online]. Available: https://www.ruckuswireless.com/products/software-and-saas/cloudpath

[11] Duo. Multi-Factor Authentication (MFA). [Online]. Available: https://duo.com/product/multi-factor-authentication-mfa

[12] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Annual International Cryptology Conference, 1991, pp. 129-140.

[13] A. H. Koblitza, N. Koblitzb, and A. Menezes, "Elliptic Curve Cryptography: The Serpentine Corse of a Paradigm Shift," Journal of Number Theory, vol. 131, pp. 781814, May 2011.

[14] CWE. CWE-301: Reflection Attack in an Authentication Protocol. [Online]. Available: https://cwe.mitre.org/data/definitions/301.html

[15] L. Zheng, C. Song, N. Cao, Z. Li, W. Zhou, J. Chen, et al., "A new mutual authentication protocol in mobile RFID for smart campus," IEEE Access, vol. 6, pp. 60996-61005, 2018.

[16] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," IEEE Transactions on Vehicular Technology, vol. 67, no. 5, pp. 4359-4373, 2017.

[17] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in Proceedings of APPSEM 2005 workshop, 2005, pp. 1-17.

[18] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Transactions on information theory, vol. 29, no. 2, pp. 198-208, 1983.

[19] D. Basin, S. Mdersheim, and L. Vigan, "OFMC: A symbolic model checker for security protocols," International Journal of Information Security, vol. 4, no. 3, pp. 181-208, 2005.

[20] M. Turuani, "The CL-Atse protocol analyser," in International Conference on Rewriting Techniques and Applications, Springer, 2006, pp. 277-286.

[21] Moteiv Corporation. (2006). Tmote Sky: Ultra low power IEEE 802.15. 4 compliant wireless sensor module.

[22] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed, and S. Aslam, "Hierarchical and flat-based hybrid naming scheme in content-centric networks of things," IEEE Internet of Things Journal, vol. 5, pp. 1070-1080, 2018.

[23] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in Proceedings of the 7th international conference on Information processing in sensor networks, 2008, pp. 245-256.

[24] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," ACM SIGOPS operating systems review, vol. 34, pp. 93-104, 2000.

[25] D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to networked embedded systems," ACM Sigplan Notices, vol. 49, pp. 41-51, 2014.

[26] Synopsys. Right-Sizing Your Cryptographic Processing Solution.

[27] NS-3. Consortium. The Network Simulator 3. [Online]. Available: https://www.nsnam.org/

[28] Ruckus Networks. Case Study: Evangel University - 802.11AC Wave 2 Hits Higher Education. [Online]. Available: `https://webresources.ruckuswireless.com/pdf/case-studies/cs-evangel-univeristy.pdf`

[29] Duo. Case Study: University of Michigan Departmental Computing Organization. [Online]. Available: https://duo.com/use-cases/case-studies/university-of-michigan-departmental-computing-organization

[30] Duo. Case Study: Duke University. [Online]. Available: https://duo.com/use-cases/case-studies/duke-university

[31] M. Safkhani and A. Vasilakos, "A New Secure Authentication Protocol for Telecare Medicine Information System and Smart Campus," IEEE Access, vol. 7, pp. 23514-23526, 2019.

[32] F. Ye, Y. Sun, and A. Rettig, "Authentication and Access Control for an IoT Green Roof Monitoring System," in 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017, pp. 251-256.

[33] N. Li, D. Liu, and S. Nepal, "Lightweight mutual authentication for IoT and its applications," IEEE Transactions on Sustainable Computing, vol. 2, pp. 359-370, 2017.

[34] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," IEEE Transactions on Dependable and Secure Computing, 2017.

[35] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," IEEE Sensors Journal, vol. 16, no. 1, pp. 254-264, 2015.

[36] K. Han, J. Kim, T. Shon, and D. Ko, "A novel secure key paring protocol for RF4CE ubiquitous smart home systems," Personal and ubiquitous computing, vol. 17, no. 5, pp. 945-949, 2013.

[37] S. S. Alotaibi, "Registration center based user authentication scheme for smart E-governance applications in smart cities," IEEE Access, vol. 7, pp. 5819-5833, 2018.

[38] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," IEEE Access, vol. 8, pp. 43711-43724, 2020.

[39] Ruckus Networks. What's all the Ruckus? [Online]. Available: `https://www.ruckuswireless.com/company/overview`

[40] Duo. Security and Reliability. [Online]. Available: `https://duo.com/about/security-and-reliability`

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Author Biographical Sketch

Abubakar Sadiq Sani received the MSc Degree from Middlesex University, London, United Kingdom, in 2012, and the Professional Education from Massachusetts Institute of Technology, Cambridge, United States of America, in 2014. He is a Certified Ethical Hacker and EC-Council Certified Security Analyst and worked in the industry for a few years. He received the PhD degree in Engineering and Information Technologies from The University of Sydney, Sydney, Australia, in 2020. He is currently a Senior Lecturer with the School of Computing and Mathematical Sciences, University of Greenwich, London, UK. His primary research interests include cybersecurity and privacy for the Internet of Things, network and communication security, enterprise resource planning, secure software engineering, and blockchain for cybersecurity and its application in the Internet of Things.

Professor Elisa Bertino (F'02) is a Professor of computer science at Purdue University. She is currently the Director of the Purdue Cyberspace Security Laboratory (Cyber2Slab), where she leads multi-disciplinary research in data security and privacy. She was a Professor and the Department Head with the Department of Computer Science and Communication, University of Milan. She has been a Visiting Researcher with the IBM Research-Almaden, San Jose, Microelectronics and Computer Technology Corporation, Rutgers University, and Telcordia Technologies. Her research focuses on digital identity management, biometrics, data trustworthiness, privacy techniques, security for sensors and drones, security for content distribution networks, systems for the management of security and privacy policies, assured information sharing. She is a Fellow member of ACM, IEEE and AAAS. She received the 2002 IEEE Computer Society Technical Achievement Award for outstanding contributions to database systems and database security and advanced data management systems and the 2005 IEEE Computer Society Tsutomu Kanai Award for pioneering and innovative research contributions to secure distributed systems.

Dong Yuan received the BEng and MEng degrees from Shandong University, Jinan, China, in 2005 and 2008, respectively. He received the PhD degree from Swinburne University of Technology, Melbourne, Australia, in 2012, all in computer science. He is a senior lecturer in School of Electrical and Information Engineering, The University of Sydney. Prior to that, he was a research fellow in the School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia, 2012-2015. His research interests include cloud computing, data management in parallel and distributed systems, scheduling and resource management, cybersecurity, Internet of Things, and workflow systems.

Ke Meng (M'10–SM'19) received the Ph.D. degree in electrical engineering from the University of Queensland, Brisbane, QLD, Australia, in 2009. He is currently a Senior Lecturer with the School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW, Australia. He was previously a Lecturer with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW, Australia. His research interests include pattern recognition, power system stability analysis, wind power, energy storage, and smart systems.

Zhao Yang Dong (M'99–SM'06–F'17) obtained Ph.D. degree from the University of Sydney, Australia in 1999. He is currently the SHARP professor and Director of the University of New South Wales Digital Grid Futures Institute, The University of New South Wales, Australia. He is also Director for ARC Research Hub for Integrated Energy Storage Solutions. He was previously Professor and Head of School of Electrical and Information Engineering, University of Sydney, and Ausgrid Chair and Director of the Ausgrid Centre for Intelligent Electricity Networks, the University of Newcastle, Australia. He also held industrial positions with Transend Networks (now TAS Networks), Australia. His research interests include smart grid, power system planning, power system security, renewable energy systems, electricity market, load modelling, and computational intelligence and its application in power engineering. He is an editor of IEEE Transactions on Smart Grid, IEEE Power Engineering Letters, and IET Renewable Power Generation.

Credit Author Statement - SPrivAD

Abubakar Sadiq Sani: Abstract, Introduction, Network Architecture and Attack Model, Proposed Solution, Implementation and Experimentation, Formal Security Validation, Case Studies, Related Work, Conclusion and Future Work.

Elisa Bertino: Abstract, Introduction, Network Architecture and Attack Model, Proposed Solution, Security and Privacy Analyses, Implementation and Experimentation, SCABA, Related Work, Case Studies, Conclusion and Future Work.

Dong Yuan: Abstract, Introduction, Network Architecture and Attack Model, Performance Analyses, Formal Security Validation, SCABA, Related Work, Case Studies, Conclusion and Future Work.

Ke Meng: Abstract, Introduction, Network Architecture and Attack Model, Performance Analyses, SCABA, Related Work, Conclusion and Future Work.

Zhao Yang Dong: Abstract, Introduction, Network Architecture and Attack Model, Security and Privacy Analyses, SCABA, Related Work, Conclusion and Future Work.

Signed

All Authors

December 20, 2021