Distributed Key Management in Microgrids

Vaios Bolgouras, Christoforos Ntantogian, Emmanouil Panaousis, *Member, IEEE,* and Christos Xenakis

***Abstract*—Security for smart industrial systems is prominent due to the proliferation of cyber threats threatening national critical infrastructures. Smart grid comes with intelligent applications that can utilize the bidirectional communication network among its entities. Microgrids are small-scale smart grids that enable Machine-to-Machine (M2M) communications as they can operate with some degree of independence from the main grid. In addition to protecting critical microgrid applications, an underlying key management scheme is needed to enable *secure M2M message transmission* and *authentication*. Existing key management schemes are not adequate due to microgrid special features and requirements. We propose the Micro sElf-orgaNiSed mAnagement (MENSA), which is the first *hybrid* key management and authentication scheme that combines Public Key Infrastructure (PKI) and Web-of-Trust concepts in microgrids. Our experimental results demonstrate the efficiency of MENSA in terms of scalability and swiftness.**

Keywords*—*Microgrid, Security, Machine-to-Machine, Key management

I. Introduction

During the last decades, nation states turn to renewable sources to diversify their energy mix and cope with the increasing demand for energy power. The traditional power grid can neither cope with the efficient management of diverse energy sources nor can respond effectively to events leading to blackouts. The introduction of Information and Communication Technology (ICT) to traditional power networks provides certain advantages like efficiency, reliability, resilience, and distributed intelligence. This led to the proliferation of smart grid as the next generation of the power grid.

Along with smart grid, the *microgrid* concept gains popularity. A microgrid is formed by a group of electricity producers and consumers in a limited geographic location. It is typically connected to a smart grid but it can also operate autonomously, in an "islanded" mode, depending on physical conditions and policies agreed among its members. A microgrid is a network of interconnected smart devices that have the ability to communicate bidirectionally either by using the aforementioned machine-to-machine (M2M) communication paradigm in islanded mode, or through the internet. Microgrids utilize power consumption-oriented applications which are of highly sensitive nature. This poses the requirement for trusted communication within the network.

The similarities and dependence between microgrids and smart grids [1] expose them to common cyber threats [2], [3], [4]. To mitigate the risk associated with a number of threats, *key management* can support cryptographic operations required for securing microgrid M2M communications and establishing trust relationships.

The major challenges, which specifically concern key management in microgrid networks, are the following:

C1. a microgrid is a network with *high churn* meaning that nodes frequently join and leave, affecting the efficiency of centralized solutions due to the overhead created by multiple and constant node connection requests to a single entity;

C2. when the Certification Authority (CA) is compromised, the traditional approach is to revoke all certificates issued and this is an administratively intensive task that would temporarily obstruct the smooth operation and impair information exchange;

C3. a microgrid can operate either in parallel with an existing power grid or in an "islanded" mode using the M2M communication paradigm; if smart meters lose connectivity to the CA, e.g. due to network outages, it is not currently feasible to validate their certificates affecting the security level of the entire microgrid and the seamless execution of the processes performed inside the network; and

C4. the storage of certificates to a central server creates a single point of failure which may result in the discontinuation of all network operations.

In this paper, we resolve these issues by proposing a novel key management scheme. Nodes can join and leave frequently without having a negative impact on the network's efficiency, and if the endorser of the certificates gets discredited, no certificate revocation will be required. Similarly, if the endorser becomes unavailable, the network operations will not cease. Due to the MENSA's decentralized nature, there is no single point of failure. We have particularly focused on proposing a fast, flexible and scalable solution with low overhead that requires the minimum number of modifications, in terms of software and hardware, to the microgrid nodes. MENSA has the ability to operate without a CA, as a decentralized and distributed network; consequently the CA-related operational problems and security threats mentioned above are mitigated.

In this paper, we present a distributed and scalable key management and authentication scheme for microgrids, namely the Micro sElf-orgaNiSed mAnagement (MENSA). We build on the basic concepts of the scheme presented in [5] and provide a detailed description of all key management operations presenting a complete solution tailored to microgrids. We have implemented and experimentally evaluated MENSA in microgrid environments. To the best of our knowledge, this is the first paper that combines (a) PKI and (b) the WoT concept found in Pretty Good Privacy (PGP) [6], in a hybrid solution providing efficient look-ups of trust relationships, key manage-

V. Bolgouras is with Neurosoft S.A., Athens 14122, Greece (e-mail: v.bolgouras@neurosoft.gr)

C. Dadoyan and C. Xenakis are with the Department of Digital Systems, University of Piraeus, Piraeus 18534, Greece (e-mail: dadoyan, xenakis@unipi.gr)

E. Panaousis is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK (e-mail: e.panaousis@surrey.ac.uk)

ment, entity authentication. It also ensures device integrity by employing remote attestation in microgrids. MENSA presents advantages over existing schemes, like ef ciency, scalability and decentralization. Additionally, for credential protection and critical operation execution, especially on the smart meter side that the customer has physical access to, we propose the utilization of a Trusted Execution Environment (TEE) [7].

In the next section we discuss related work and background on key management approaches for smart grids. Section III describes MENSA. Section IV presents simulation-based performance results for MENSA while Section V discusses security related issues and gives a critical overview of this paper. Finally, Section VI summarizes important points and concludes this paper.

II. Related Work

There is a large number of key management schemes for smart grid architectures, which have been proved to be insecure and susceptible to different attacks. The solution in [8] combines elliptic curve public key cryptography based on the Needham-Schroeder protocol, along with symmetric keys for the agents to communicate with each other; however, it is susceptible to man-in-the-middle attacks [9]. The authors of [9] propose a symmetric key distribution scheme by utilizing a trusted third party, mainly to address problems in the literature regarding the certi cate revocation process; this solution was also found to be vulnerable to an impersonation attack [10]. One of the most straightforward, yet insecure, key management methods is sharing a single symmetric key among many or even all smart grid nodes, used by systems like [11]. In such a system, if one node is compromised, then the whole network is at risk.

There are more works that propose key management schemes for smart grids, like [12], [13], [14], [15] and [16]. However, most of them target large network infrastructures controlled by a CA. This architecture is not suitable for microgrids that by de nition can operate autonomously using the machine-to-machine communication paradigm (C3). [12] and [13] are based on binary trees to manage secret keys shared among network entities. [14] uses symmetric keys with frequent key updates among the nodes of the smart grid, hindering scalability. [15] employs a trusted third party to wirelessly manage and distribute encryption keys that will be used on meter data, based on the location of the smart meters. Authors in [16] proposed a mechanism for mutual authentication between a smart meter and an authentication server, utilizing Enhanced Identity-Based Cryptography (EIBC) for securing smart grid communications using a public key infrastructure.

There are some schemes that are closer to MENSA, since they exclusively focus on key management in microgrids. First, the solution in [17] requires the microgrid nodes to be organized in a binary tree. When a node joins or leaves the network, partial key update is performed from the parent of the joining/leaving node up to the root of the tree. This solution cannot cope with (C1), presented in the previous section. In [18] the functionality of the proposed distributed key management approach for microgrids is based on an one-way function tree. In this case, partial key update is needed when nodes join or leave the microgrid (C1). The solution in [19] utilizes a key management server to store microgrid keys creating a single point of failure (C4). In [20] another approach to key management is presented, where the scheme operates as a hierarchical network, with a pair-wide key pre-distribution on the microgrid's devices.

A reminiscent of our proposed solution is Chord-PKI [21], a generic scheme which combines PKI with Chord for secure communications over P2P networks, utilizing threshold cryptography. On the contrary, in order to avoid the hurdles of threshold cryptography that requires the use of CAs to distribute the key shares (C2), our work makes use of WoT instead. To the best of our knowledge, the only solution that applies Chord in the smart grid domain is [22]. However, their approach is focused only on improving the management of Certi cate Revocation Lists (CRLs), which is only one aspect of a key management scheme.

III. MENSA

This section presents in detail theidMb sElf-orgaNiSed mAnagement (MENSA) scheme. First, an overview of the architecture and operation of a microgrid is given, followed by the basic building blocks and operations of MENSA.

A. Functional Components

The components that comprise a basic microgrid infrastructure are presented in Figure 1.

Fig. 1: Functional components of a microgrid.

In Figure 1, the TEE is depicted as a standalone element for demonstration purposes. This is embedded in the smart meter. The microgrid can either be connected to the main utility grid or operate in a standalone “islanded” mode. In the former case, the microgrid is able to sell or buy energy from the grid according to its members' needs.

B. Architecture

We assume that in each home or building there are one or more smart meters connected to a segmented mesh network, which further includes aggregators assisting the electricity provider with the collection of information from the smart meters and also introducing new nodes to the network. Despite the existence of an hierarchical structure and many sub-networks forming a main grid, separated with rewalls from each other, integrating MENSA into such networks does not introduce any impediments. Each part of the network can have its own MENSA structure and communicate with the utility

Fig. 2: MENSA as an overlay network.

independently. A single smart meter can be connected to several appliances or renewable energy resources in order to send commands and measure energy consumption or production. Each smart meter contains a special element that implements a Trusted Execution Environment (TEE) to protect the device's private keys [23], [24], [25]. In this way, the device's private keys can only be used from within the TEE and they are never exposed to anyone that has physical access to the device.

Figure 2 shows MENSA as an overlay network within the microgrid architecture. MENSA resides on top of a logical layer, an overlay, which allows bi-directional communications among the microgrid components. This is a Distributed Hash Table (DHT) layer that provides a generic DHT functionality, where we can store overlay (key, value) pairs, as in every other locally stored hash table, and retrieve the value back based on the key. MENSA is based on Stoica's Chord [26], where nodes are placed as IDs occupying a circular identifier space; in our case, the smart grid and each node of the microgrid are represented as a single node of the Chord ring. Chord is scalable and efficient, requiring $O(\log N)$ communication hops, where N is the total number of nodes in the system. Chord and Kademlia are considered the main DHT candidates for P2P communication [27]. Despite that when using Kademlia to relay messages shorter paths are formed, Chord provides better scalability when considering messaging cost and a smaller average packet size [27], causing less overhead at the network. The overlay can support diverse applications that require efficient look-up services, like authentication with MENSA; other services that could benefit from the overlay include billing, and secure aggregation.

Above the DHT layer there is MENSA, according to which the overlay key of a node n is $K_n = h(Pk_n + ID_{device})$. This concatenation makes use of a public key Pk_n (a key stored to each node, details will be provided in the next section) and it ensures that K_n will be unique even when the device identifiers are not. On the other hand, the overlay value is the digital certificate of the node, which follows the OpenPGP message format [6] creating the MENSA ring. This serves as a certificate storage and query structure. The certificates are stored in the DHT in a distributed fashion. Upon a direct trust path creation between node A and node B, node A retrieves B's certificate and checks its validity. If successful, A inserts node B in its routing table; this is called finger table, as in Chord. These steps (presented in more detail at Section III-D) are repeated, creating a Web-of-Trust (WoT). Finally, a MENSA ring is a Chord ring which uses the tuple $(K_n, Cert_n)$ as (key, value).

C. Prerequisites and assumptions

Initially, node n has to generate a K_n, as presented in Section III-B, to connect to the overlay. As a first step, n receives or generates a key pair Pk_n=Sk_n, upon which the creation of a self-signed certificate is based. Then, n receives a certificate $Cert_n$ signed by other nodes that act as introducers to the MENSA ring. Certificate $Cert_n$ follows the OpenPGP message format [6] and, apart from the first signature from the introducers, it can carry signatures from other endorsers if needed. Introducers can be the administrative owners or other "empowered" nodes of the microgrid acting on a local level.

D. Node join

In order to join the MENSA ring, a node should first have its certificate signed by at least one introducer that owns a valid certificate otherwise the node cannot join the ring. The more signatures the certificates get from introducers, the easier the creation of trusted paths becomes and a safeguard is in place in case of an unavailable or compromised introducer. The selection of the certificates that shall be checked upon a node's join is linked to the creation of its finger table according to Chord. Each node possesses this table containing a list of next hops to be used when executing a search, ensuing improved look up operation. MENSA nodes that do not have a certificate signed by a trusted introducer, are considered untrusted and cannot be included in finger tables, hence they do not become a part of the network's WoT. The join operation is concluded when the new node verifies the validity of all certificates assigned to the nodes in its finger table.

Node join can be explained as follows. We assume that node n wants to join the MENSA ring shown in Figure 2. First, n gets its certificate signed by one, at least, introducer, which here is b. When n joins the ring, Chord assigns to its finger table one or more node IPs. Node n will check the validity of the respective certificates (i.e., overlay values) assigned to the nodes that are part of its finger table. This validation assures that a certificate is well-formed, not expired and signed by an introducer node.

```
Function nodeJoin(k)
  if Cert_i is valid then
    while next(IP_k) to be stored in fingerTable_n
     do
      if Cert_k is signed by introducer i then
        //  Cert_k is trusted
        n stores IP_k in fingerTable_n
      end
    end
  end
```

Algorithm 1: Certificate check during node join.

```
Function n.find(n')
  if n' resides in n.fingerTable then
    // n' is trusted
    return success
  else
    send request to the next trusted node p closest to
      n' from node n
    if n' resides in p.fingerTable then
      // n' is trusted
      return success
    else
      send request to the next trusted node k
        closest to n' from node p
      .
      .
      .
    end
  end
  // No trust chain was found
  return failure
```

Algorithm 2: Searching for another network node.

The formal process is presented in Algorithm 1. When the IP values have been assigned to a node's finger table on the Chord level, n goes through all of them to check the corresponding certificates. At first, n checks the validity of the introducer's certificate $Cert_i$; if it is valid and $Cert_n$ is signed by it, then the corresponding IP is saved in the finger table.

E. Normal operation

After the node join procedure takes place, the microgrid operates with certificates signed by the introducers. If an introducer's certificate is compromised, invalid, expired or its validity cannot be verified (e.g., due to intermittent communication or no connectivity), all operations (apart from node join) can be executed based on signatures from the rest of the microgrid's introducers. When a node n wants to securely communicate with a node n' using MENSA, a standard chord search will commence. This action can be executed efficiently utilizing the Chord ring that has already been formed and contains routing tables with nodes that are already part of the WoT. Using its finger table, n will eventually locate n'.

An example operation of MENSA is demonstrated with the help of Figure 2. We assume that node n wants to communicate securely with node p; this is feasible by performing a chord search utilizing the finger tables that contain only trusted nodes. Algorithm 2 presents this formal process. An iterative search commences from node n to find a node with n' stored in its finger table. As long as the search does not locate the desired node, chain length grows. If there is no node that trusts n', the search is deemed unsuccessful.

F. Certificate revocation

Regarding certificate revocation, in a distributed system like a microgrid, a typical Certificate Revocation List (CRL) may be difficult to manage [22]. In MENSA there are three ways to revoke a certificate: implicitly, explicitly by the same node, or explicitly by another node.

The implicit revocation of $Cert_n$ is the simplest one: each certificate is revoked after its expiration time has passed. Expired certificates are recognized during normal certificate verification. To facilitate this, in the finger tables, IPs are accompanied by an expiration timestamp. The node with the expired certificate will have to get through the verification process again.

In the second case, a node can revoke its own certificate by using a revocation certificate, as described in OpenPGP [6]. The revocation certificate of a node n ($RevC_n$) is a certificate that revokes n's public key and it is signed by n's private key. The revocation certificate does not require the knowledge of the private key. In order to cover cases where n's private key is lost or not accessible, $RevC_n$ is created at the same time with the normal certificate $Cert_n$ and stored locally. A node n that wants to revoke its certificate does not need to send the revocation certificate to all nodes in MENSA, but only to the nodes that have n in their finger table. As a result, n gets completely cut off from the network's WoT.

It is also possible that a different node n' explicitly revokes $Cert_n$. In this case n' will be an empowered node, like the introducer, which is authorized to revoke a certificate of another entity (e.g., a smart meter). To enable n' to revoke $Cert_n$, node n sends its revocation certificate $RevC_n$ to n' when it is created and n' stores it locally. Normally, n' will revoke $Cert_n$ if n is misbehaving. Abnormal node behaviour can be detected utilizing behaviour-based or specification-based methods like [28].

Trusted computing is intended to provide reliable evidence about the state of software executing on a system; malicious behaviour can be identified through remote attestation [29]. Using remote attestation, n' can determine if n is compromised and revoke $Cert_n$ accordingly. Similarly to the previous the revocation certificate is sent only to the nodes that have n in their finger table. Here, however, the additional risk is that n' can revoke $Cert_n$ even when it is not necessary, leading to DoS. In order to minimize this risk, revocation certificates are stored inside the TEE of the designated node, in secure persistent storage memory which is cryptographically protected, like in [30].

G. Trusted Execution Environment

In order to protect elements (e.g., smart meters) that reside on the customers' premises from intervention, MENSA adopts the concept of TEE [7] to provide a protected environment with limited access. A TEE provides a trusted computing environment utilizing two virtual processing cores with different privileges to create two "worlds": the *Normal*, for executing common application processes, and the *Secure*, for executing security-sensitive code only. We used a slightly modified version of the trusted computing environment proposed in [25], utilizing PGP certificates. The employed TEE supports the following operations.

1) *Secure storage*: In MENSA, the TEE is used as a secure storage for IPs that reside in finger tables, secret and private keys, as well as revocation certificates. Secure storage provides encryption and integrity checking for all saved objects. In this way, unauthorized access to keys and certificates is prevented and any modifications are disclosed.

2) *Finger Table updates*: All the nodes that reside inside a smart meter's finger table are considered trusted. It is essential that the operations related to the processing of the finger tables are secure and not exposed to tampering. Adversaries could introduce, to the network, malicious nodes if the finger tables were updated outside the TEE. In addition to that, normal nodes could be excluded from the network by being deleted from finger tables, by unauthorized entities, possibly leading to DOS attacks.

3) *Key revocation*: As presented in Section III-F, the certificate of a node n can be revoked by another node n'. Remote attestation is a procedure by which n' can control whether n runs the designated software/firmware or not; if not, n' uses $RevC_n$ to revoke $Cert_n$. The attester (i.e., n'), is engaged into a challenge-response protocol with the target node (i.e., n), where n responds back with a signed hash of its software and firmware. All operations related to remote attestation are executed inside TEE without leaving the protected environment. Key revocation using remote attestation enables the microgrid to eliminate modified smart meters that may be malicious, providing an extra layer of protection against attackers who target individual smart meters.

4) *Security operations*: The TEE offers a protected environment for operations that are directly relevant to key management such as signing of other certificates. Procedures that require the use of one or more of the secret or private keys, like digital signatures, are executed inside the TEE so that keys or other sensitive material are never exposed to the potentially hostile environment of the smart meter.

H. Node leave

When a node leaves the network it can explicitly revoke its certificate, by sending a revocation certificate, or implicitly, by not taking any action. In the latter case, its certificate will expire. On the Chord level, a re-organization of the finger tables will take place. When this happens, the affected nodes will need to check the certificates of the newly assigned nodes. For example, in Figure 2 it is inferred that node k is part of node j's finger table. If node k leaves the network, the gap left in j's finger table will be covered by the successor of node k, which in this case is node p. Finally, the process described in Algorithm 1 is followed by node j to check the validity of $Cert_p$.

IV. Evaluation

First, we evaluate the scalability of MENSA by analyzing the maximum size of the finger tables in relation to the microgrid size. Then, we provide experimental results from implementing MENSA on the OverSim P2P simulator [31][1].

[1]For reproducibility purposes we make the simulation code available to the public (https://github.com/VaiosBolgouras/MENSA).

TABLE I: Effect of ring growth in MENSA.

N	fingerTable size
500	8
5,000	12
15,000	13
30,000	14
.	
.	
5,000,000	22

TABLE II: Simulation parameters.

Parameter	Scenario 1	Scenario 2
Network size	up to 3,000	{500 to 30,000}
routingType	iterative	iterative
joinRetry	2	-
stabilizeDelay	20 sec	20 sec

A. Scalability

The DHT, which is comprised by the $K_n/Cert_n$ pairs, along with the nodes' finger tables are considered the foundation of MENSA and contribute to the scalability of this implementation. The entries in the finger table of each node do not increase substantially with number of nodes, while at the same time no additional certificates are saved locally. Each node in the MENSA ring has in its finger table at most a low percentage of IPs, compared to the number of total network nodes, as shown in Table I. The values were derived by Theorem 2 from [32], which states that the searches performed in such a network have an upper bound of $O(\log N)$.

MENSA stores a small number of node IPs in each node (e.g., a maximum of 14 in a ring of 30,000 nodes). This demonstrates that MENSA is scalable in microgrid environments and can also support much larger networks (e.g., 5,000,000 nodes). Further proof of MENSA's scalability is shown by the experimental results presented next.

B. Implementation

We have implemented MENSA in C++ and integrated it into the OverSim P2P network simulation framework. We define two simulation scenarios: in Scenario 1 nodes are added to the ring one at a time measuring the Node Join delay, while in Scenario 2 we have microgrids of stable sizes and we measure the delay of different operations (e.g. Node Join). An overview of the simulation parameters can be found in Table II.

In our experiments, the *iterative* *routingType* is used. According to this type, the initiating node receives information from intermediate nodes on how to reach the target of a lookup operation, as described in the example of Section III-E.

When a node tries to join the ring, it is allowed to perform 2 attempts: this is indicated by the joinRetry parameter. Finally, *stabilizeDelay* which is set to 20 sec, shows the time for which the simulation was run before commencing the collection of results; this is necessary in order to complete the processing of any unfinished queues from the bootstrapping phase.

In Scenario 1, we consider that at least one introducer node, with a valid certificate, is available and we evaluate the mean Node Join delay (more details in Section IV-C). To measure Node Join delay, the simulator is configured to introduce new nodes to the microgrid, until a maximum number of 30,000 nodes is reached. In Scenario 2, we run a series of experiments, using different microgrid sizes, ranging from 500 to 30,000 nodes, and we evaluate the following:

- average trust chain length (Section IV-D)
- probability of finding trust between two random nodes (Section IV-E)
- average search time when a trusted path between two random nodes is to be established (Section IV-F)

For host-related delays we employed a trusted computing enabled smart meter as described in [25], where the TEE environment is provided by Open-TEE [33]. We introduced the measured delay of signing a single digital certificate (38.47 ms) by the TEE to the simulation framework. The main purpose of Open-TEE is to support the development of trusted applications without relying on any specific hardware platform. We have implemented MENSA by utilizing Open-TEE as a proof of concept but also taking into account an alternative use of Open-TEE mentioned by the authors of [33].

C. Node join

For measuring the delay of a single node joining the MENSA ring, we implemented Scenario 1. The simulator is configured to start from a zero-size network and introduce new nodes, up to a maximum of 30,000 nodes. The join process of the node includes the operations described in Section III-D. The new node gets its certificate signed by at least an introducer, checks the validity of the certificates that are to be inserted in the finger table and it creates the finger table.

The simulation results show that for up to 5,000 nodes the mean join time is 1.55sec, while for Node Joins taking place between 20,000 to 30,000 nodes, the respective value is 2.2sec. This slight decline in performance, considering the network's magnitude, is mainly a byproduct of the overall increased requests the microgrid nodes have to process. The certificates' signing and validation delays also have an impact, though it is close to negligible because of the increase by just two nodes in the finger tables' maximum capacity (shown in Table I). This means that the introduction of new nodes to the microgrid, which is executed during the installation of a smart meter, or when a certificate renewal takes place, has an insignificant and predictable delay.

D. Trust chain length

In order to measure how the average trust chain length changes with different sizes of the microgrid, we performed a series of experiments following Scenario 2. The trust chain is an ordered list of certificates starting from the node initiating a look-up operation up to the target node; the procedure of creating a trust chain is described in Section III-E.

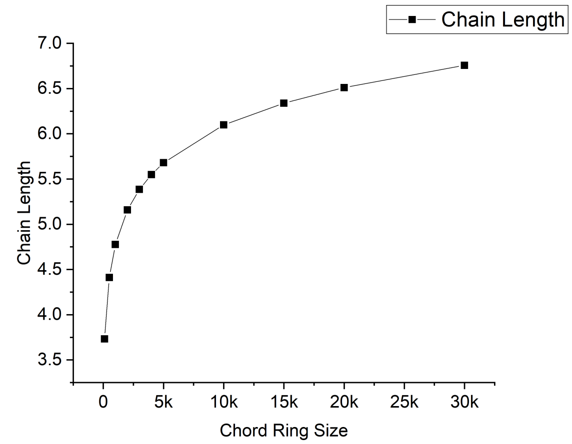


Fig. 3: Average certificate chain length.

The length of the chain includes the initiator and the target node; for example, a trust chain of length 5 means that three intermediate nodes are placed between the two aforementioned nodes. For each request, we randomly select two microgrid nodes and check if the first can trust the second by using Algorithm 2.

The graph in Figure 3 shows that a chain length (from 3.7 to 6.7) is expected for microgrid sizes up to 30,000 nodes. This means that the two nodes trust each other implicitly with intermediate nodes ranging from 1 to 5 nodes. In a limited environment like a microgrid, we regard implicit trust with this number of intermediate nodes as acceptable, thus MENSA performs adequately in terms of trust. For greater microgrid sizes, as we can infer from Figure 3, the chain is not expected to increase significantly; indicatively, the chain length from 10,000 to 30,000 has increased only by one, from approximately 6 to 7, maintaining the trustworthiness of MENSA operations.

An initial observation here, which also applies to the rest of the experiments in the following sections, is that no significant changes are perceived in the microgrids behavior as the size increases, which is a testament to the scalability of MENSA. This behavior is the result of the progressive and minimal increase in the finger table nodes, as presented above in Table I.

E. Probability of finding trust

With this series of experiments we want to determine the probability that two random nodes will be able to establish trust relationships between them, even when one of the network's introducers has invalid certificate. The network is created according to Scenario 2. For each request, we first randomly select two microgrid nodes and check if the first can trust the second, following Algorithm 2. If trust is found between the two nodes we mark it as success otherwise we mark it as

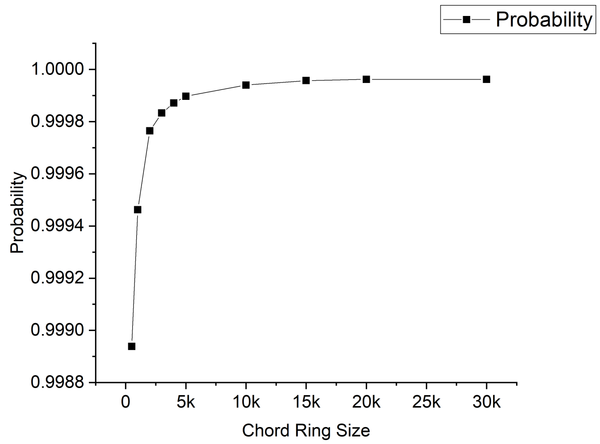


Fig. 4: Average Trust percentage.

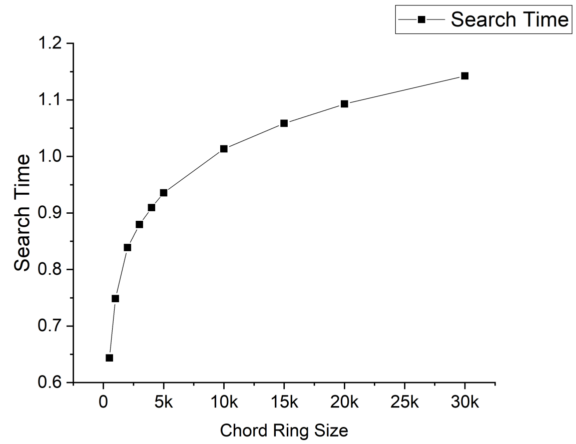


Fig. 5: Search Time.

failure. In the end, we compute the percentage of successes against the total requests.

Figure 4 shows that regardless of the microgrid size, it is almost certain that a trusted path is going to be established. As the number of nodes in the network increases, the probability of finding trust also trends upwards, since more paths and alternative choices that could be utilized to reach a node become available. After 10,000 nodes we observe very slight increase, as the probability has almost reached its maximum value of 1, which means absolute certainty that a trust chain exists for every potential request. Comparing this result with a traditional centralized PKI approach with an invalid CA certificate, the latter would present a 0% of success since all node certificates would be considered invalid as well.

F. Search time

This series of experiments present the average time needed for a random node to establish trust relationships with another random node, according to Scenario 2. The delay was measured from the time the first node creates the request until the trust relationship is verified.

Figure 5 shows that for microgrid sizes up to almost 10,000 nodes the average search time is under 1 sec (0.64-1.01), while it maintains low values up to 1.14 sec as the micro grids size increases to 30,000 nodes. These low values are mainly a byproduct of building the trust relationships between the nodes at the joining stage. A Node Join takes place much more rarely than searches do, so it makes sense to embody a time and resource consuming activity like certificate verification at the earlier stage. Responsible for the increase in search time as the network size grows is the corresponding increase of the chain length, as messages need time to be transmitted, received and processed at each node.

V. Discussion

In this section, we provide a security analysis of our scheme by employing a few representative attack scenarios. For each case, we study how MENSA will respond. Next, we review common issues encountered in ICT, that may appear in a microgrid as well, and analyze how MENSA alleviates them.

A. Security Analysis

In this section we analyze the possible attack scenarios derived from the characteristics of our architecture and the desirable system properties.

1) Node Join: An attempt to insert a fake node controlled by a malicious party in the MENSA ring could take place. However, bootstrapping of new nodes is controlled by the administrative owner of the microgrid, so it would not be possible for such an action to take effect.

2) Introducer Certificate Revocation: An issue that may arise when an introducer's certificate is revoked, is the impact such an action is going to have on the already deployed nodes. In the approach followed by a traditional PKI, all certificates issued by the CA whose certificate is revoked, would have to be revoked as well and the need for the issuance of new certificates would arise. In MENSA, a node that has already joined the microgrid will have its certificate signed by more than one introducer, and it will be considered valid until expired or revoked. The main advantage is that when an introducer's certificate is revoked, it is not necessary to revoke any node certificate of the microgrid; existing certificates can be verified based on other endorsers' signatures. In practice, in the traditional case the operation of the microgrid will be stalled until new certificates are issued, affecting all of its nodes; in MENSA, operation will continue as usual.

Multiple or all introducer's certificates getting invalidated at the same time is considered unlikely, due to the decentralized and distributed nature of the network. When all the introducers are compromised at the same time, it would heavily impact on the network and negatively affect its operations.

3) Byzantine Attacks and Misbehaving nodes: To further protect the microgrid from attacks where a single or multiple authenticated devices are compromised and under the control of adversaries, we can use a reputation framework to include ratings from all the transactions between the nodes, in addition to the above certificate path-building method. On every communication between an initiating and a target node, an outcome is recorded and its reputation score is calculated. For the execution of these transactions and the storage of scores, the TEE will be used so that not even the nodes themselves have access to such critical operations and data.

We do not wish to claim specific parameter values as accurate ratings other than the positive and negative outcomes between events. For instance, supposing a node j is malicious and is misbehaving in the look-up operation, node n can downgrade it in its reputation table and therefore avoid the problematic node during look-up. After a while, intentional look-up misbehaviour by specific nodes, whether they are part of a bigger Byzantine attack scheme, or acting alone, will be effectively represented in the rest of the ring and they will result in skipping the misbehaving nodes during the look-up operation. This approach results in a reputation based path ranking that is similar to the discrete ranking of PGP Web-of-Trust [6].

B. Critical Appraisal

In Section I we presented 4 major issues specifically for key management in microgrids and afterwards showed how MENSA overcomes them. Here we analyze how MENSA alleviates issues that are found in common ICT systems as well.

First of all, our scheme is based on the well-known PGP Web-of-Trust and asymmetric cryptography operations, so that it can be considered *resilient against well-known attacks*, at least to the extent these two building blocks can be considered as resilient. It is also *robust against key compromise*, since with the utilization of trusted computing, the secret/private keys never leave the device.

MENSA can support *distributed operation* of its nodes given that, after the bootstrapping phase, its operation can be based on a Web-of-Trust and no unique central TTP is needed. Hence, it shows high availability even over intermittent connections or no connectivity at all with part of the microgrid.

Microgrid devices, just like in the smart grid, are expected to have a long lifetime, in the order of 20 years. MENSA supports *upgradeability* since its distributed nature allows digital certificates to be easily and inexpensively updated with longer key sizes.

Regarding *scalability*, MENSA can support large numbers of devices as after bootstrapping, where the administrative owner of the microgrid must be involved, MENSA is decentralized and there is low administrative cost. Moreover, as demonstrated in Section IV-A, even large microgrid sizes result in small finger tables, adding low overhead to each node.

Efficiency is highly related to the hardware that will be used. Although MENSA utilizes digital certificates and asymmetric cryptography, there are ways to mitigate the performance penalty by using session keys based on these certificates. Our experimental results in Section IV proved that for microgrid sizes of up to 30,000 nodes, MENSA is efficient even when using smart meters whose hardware specifications are comparable to embedded devices (e.g., Raspberry Pi).

The *computational overhead* for the smart meters is minimal. Handling and forwarding messages when searches are performed is a resource consuming task that essentially refers to carrying out a linear search with complexity $O(n)$ for a specific node, as shown in Table I. Moreover, the process of the RSA based [6] certificate verification introduces $O(M^2)$ overhead [34], where M is the modulus[2] length. Introducers have to handle the task of signing certificates, in which case the overhead is $O(M^3)$.

VI. Conclusions

Due to the microgrid's special characteristics and requirements, existing key management solutions are not directly applicable creating opportunities for innovation in the field. MENSA is the first distributed hybrid key management and authentication system for microgrids, which eliminates the need for a Trusted Third Party (TTP) with high availability.

Its operation is based on a DHT for efficient discovery of trust relationships among the microgrid nodes. Having the administrative owner of the microgrid taking part during the bootstrapping phase, we ensure that it will be hard for malicious nodes to join the microgrid. After this phase, the enforcement of a trust policy provides a decentralized and flexible solution that promotes scalability and resilience.

The proposed key management solution is intended for microgrids and can efficiently support network sizes of up to 30,000 nodes, as indicated by our simulation results. Moreover, the diagram curves demonstrate that supporting larger network sizes would be a viable option for MENSA.

We believe that MENSA will pave the way towards developing microgrids further and it will help realizing their full potential in terms of scalability and performance efficiency. On top of this lightweight solution, a wide range of intelligent programs may find application, utilizing MENSA's effectiveness and swiftness.

Acknowledgement

This research has been funded by the European Commission as part of the SealedGRID project (H2020-MSCA-RISE-2017, Project ID: 777996).

References

[1] R. H. Lasseter, "Smart distribution: Coupled microgrids," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 1074–1082, 2011.

[2] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344 – 1371, 2013.

[3] F. Aloul, A. Al-Ali, R. Al-Dalky, M. Al-Mardini, and W. El-Hajj, "Smart grid security: Threats, vulnerabilities and solutions," *International Journal of Smart Grid and Clean Energy*, vol. 1, pp. 1–6, 09 2012.

[4] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *Communications Surveys and Tutorials, IEEE*, vol. 14, pp. 998–1010, 01 2012.

[2] A private RSA key is comprised by two integers: the modulus and the *private exponent.*

[5] F. F. Demertzis, G. Karopoulos, C. Xenakis, and A. Colarieti, "Self-organised key management for the smart grid," in *Proceedings of the 14th International Conference on Ad-hoc, Mobile, and Wireless Networks - Volume 9143*. New York, NY, USA: Springer-Verlag New York, Inc., 2015, pp. 303–316.

[6] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "OpenPGP Message Format," RFC 4880 (Proposed Standard), Internet Engineering Task Force, Nov. 2007, updated by RFC 5581.

[7] GlobalPlatform, "TEE System Architecture, version 1.0," December 2011, last accessed 4-10-2016.

[8] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 375–381, 2011.

[9] J. Xia and Y. Wang, "Secure key distribution for the smart grid," *Smart Grid, IEEE Transactions on*, vol. 3, no. 3, pp. 1437–1443, Sept 2012.

[10] J. H. Park, M. Kim, and D. Kwon, "Security weakness in the smart grid key distribution scheme proposed by Xia and Wang," *Smart Grid, IEEE Transactions on*, vol. 4, no. 3, pp. 1613–1614, Sept 2013.

[11] NIST, "Guidelines for smart grid cybersecurity: Vol. 1 - smart grid cybersecurity strategy, architecture, and high-level requirements vol. 2 - privacy and the smart grid vol. 3 - supportive analyses and references," NIST, Tech. Rep., 2014.

[12] J.-Y. Kim and H.-K. Choi, "An efficient and versatile key management protocol for secure smart grid communications," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. IEEE, 2012, pp. 1823–1828.

[13] X. Long, D. Tipper, and Y. Qian, "An advanced key management scheme for secure smart grid communications," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 504–509.

[14] S. Xiao, W. Gong, and D. Towsley, "Dynamic key management in a smart grid," in *Dynamic Secrets in Communication Security*. Springer New York, 2014, pp. 55–68.

[15] I. Parvez, A. Sarwat, L. Wei, and A. Sundararajan, "Securing metering infrastructure of smart grid: A machine learning and localization based key management approach," *Energies*, vol. 9, 2016.

[16] H. Nicanfar, P. Jokar, K. Beznosov, and V. C. M. Leung, "Efficient authentication and key management mechanisms for smart grid communications," *IEEE Systems Journal*, vol. 8, no. 2, pp. 629–640, 2014.

[17] Y. H. Ma, D. D. Cui, and X. Y. Wang, "An efficient key management protocol for multi-microgrid distribution system," in *Applied Mechanics and Materials*, vol. 385. Trans Tech Publ, 2013, pp. 1007–1010.

[18] M. I. Baza, M. M. Fouda, A. S. T. Eldien, and H. A. Mansour, "An efficient distributed approach for key management in microgrids," in *Computer Engineering Conference (ICENCO), 2015 11th International*. IEEE, 2015, pp. 19–24.

[19] V. Kounev, D. Tipper, A. A. Yavuz, B. M. Grainger, and G. F. Reed, "A secure communication architecture for distributed microgrid control," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2484–2492, 2015.

[20] A. V. D. M. Kayem, H. Strauss, S. D. Wolthusen, and C. Meinel, "Key management for secure demand data communication in constrained micro-grids," in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, March 2016, pp. 585–590.

[21] A. Avramidis, P. Kotzanikolaou, and C. Douligeris, "Chord-pki: Embedding a public key infrastructure into the chord overlay network," vol. 4582, pp. 354–361, 06 2007.

[22] M. Cebe and K. Akkaya, "Efficient management of certificate revocation lists in smart grid advanced metering infrastructure," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2017, pp. 313–317.

[23] A. J. Paverd and A. P. Martin, "Hardware security for device authentication in the smart grid," in *Smart Grid Security*. Springer, 2012, pp. 72–84.

[24] J.-M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *Communications Workshops (ICC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–5.

[25] G. Karopoulos, C. Xenakis, S. Tennina, and S. Evangelopoulos, "Towards trusted metering in the smart grid," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2017 IEEE 22nd International Workshop on*. IEEE, 2017, pp. 1–5.

[26] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.

[27] E. Harjula, T. Koskela, and M. Ylianttila, "Comparing the performance and efficiency of two popular dhts in interpersonal communication," *2011 IEEE Wireless Communications and Networking Conference*, pp. 2173–2178, 2011.

[28] C. Panos, C. Xenakis, P. Kotzias, and I. Stavrakakis, "A specification-based intrusion detection engine for infrastructure-less networks," *Comput. Commun.*, vol. 54, no. C, pp. 67–83, Dec. 2014.

[29] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, "Principles of remote attestation," *International Journal of Information Security*, vol. 10, no. 2, pp. 63–81, Apr 2011.

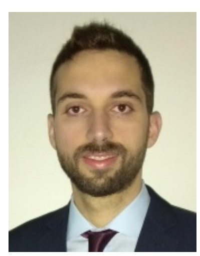
[30] D. Hein, J. Winter, and A. Fitzek, "Secure Block Device–Secure, Flexible, and Efficient Data Storage for ARM TrustZone Systems," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1. IEEE, 2015, pp. 222–229.

[31] I. Baumgart, B. Heep, and S. Krause, "OverSim: A flexible overlay network simulation framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, May 2007, pp. 79–84.

[32] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[33] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan, "Open-TEE – an open virtual trusted execution environment," in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, ser. TRUSTCOM '15. IEEE Computer Society, 2015, pp. 400–407.

[34] W. Freeman and E. Miller, "An experimental analysis of cryptographic overhead in performance-critical systems," in *MASCOTS '99. Proceedings of the Seventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE Computer Society, 1999, pp. 348–357.



Vaios Bolgouras is a Security Analyst at Neurosoft S.A., a Research Assistant at the Systems Security Laboratory, and a PhD student at the Digital Systems Department, University of Piraeus. His area of expertise focuses on network security and privacy in smart grid ecosystems. He received the B.Sc. degree in Digital Systems from the University of Piraeus, Greece, in 2015, and the M.Sc. degree in Digital Systems Security from the University of Piraeus, Greece, in 2018. Prior to Neurosoft S.A., Vaios was an Information Security Consultant at PricewaterhouseCoopers, with a main focus on data privacy and business compliance with the corresponding regulations.