Efficient Generation and Distribution of CRLs for IEEE 802.11s-based Smart Grid AMI Networks

Kemal Akkaya*, Khaled Rabieh†, Mohamed Mahmoud†, Samet Tonyali∗

Abstract—In this paper, we propose a novel algorithm for reducing the size of certificate revocation lists (CRLs) created and distributed for IEEE 802.11s-based Smart Grid Advanced Metering Infrastructure (AMI) networks. Rather than maintaining a huge-size single CRL that introduces unnecessary search time and storage, the idea is to generate groups of smart meters (SMs) within the AMI network and create CRLs based on these groups. Creating groups is appropriate in AMI networks since the SMs are stationary in contrast to traditional mobile wireless networks. Our proposed grouping algorithm is based on the created paths from leaf SMs to the gateway as well as the immediate neighborhood of each SM. Via grouping, the length of the CRL for each SM and the corresponding distribution overhead can be reduced significantly. Simulation results have shown that our approach can maintain a balance between the size of the CRL and the number of signatures generated by CAs while guaranteeing security of the communications.

I. INTRODUCTION

There has been a lot of proposals for building the underlying communication infrastructure for SG AMI applications [1]. One of the options is to use wireless communications based on several technologies such as Zigbee, RF Mesh, Wireless LAN and the newly approved IEEE 802.11s mesh standard [2]. In any case, the communication needs to be secured since wireless networks are easy targets for the attackers. Therefore, a lot of mechanisms were proposed in the recent years to address various security and privacy preservation challenges for the wireless communication infrastructure of SG [3]. All these works assume the availability of keys at the SMs. While using symmetric-key based solutions have advantages in terms of performance, these solutions require the creation, distribution and renewal of large number of keys which brings a lot of overhead to the utilities since the underlying network size for AMI would be very large. Hence, a lot of utilities opt to adapt asymmetric-key or public-key cryptography (PKC)-based solutions which has less overhead in terms of key management.

While PKC is a viable option to be used in SG communications, it still poses challenges in terms of certification of the public keys which is done by Certification Authorities (CAs). CAs also perform certificate revocation due to various reasons. Once a certificate is revoked by the CA, it needs to be published in a list called CRL and distributed to the related parties which will be using this certificate. Consequently, each SM needs to check the validity of the certificate it uses before it can verify a signature. CRLs are typically created for every node in the network and can be accessible at a public server. In the case of SG AMI applications, the ability of a SM to access the server of the CA is not possible due to communication overhead and Internet accessibility of the SMs. Therefore, CRL should be distributed to the SMs in advance so that it will be available locally at each SM. While it is possible to distribute a single CRL for all the SMs in the AMI network (or even in a city), this may not be needed since SMs will mostly be communicating with the gateway. As time passes, the CRL size may grow significantly and thus becomes a major overhead in terms of search and storage at the SMs. Given that SMs will not have a lot of resources in terms of CPU and storage, it does not make sense to use a single CRL for the whole network. It will be beneficial to reduce the size of the CRLs for SG AMI networks. This reduced size will also help in keeping the CRL distribution overhead low.

This paper proposes a novel mechanism to redefine the CRLs for SG AMI networks based on the groups of SMs that have the highest possibility for communication. The basic idea is to form CRLs for each group and use a single CRL within a group to reduce the delay and storage overhead. We identify the groups in AMI networks based on the communications patterns among the SMs. Typically, each SM needs to follow a path to the gateway and thus will have the need to talk to each of these nodes within its path. This pattern is motivated from the AMI applications where SMs power data are collected regularly. There is other communication patterns which necessitates the communication of an SM with is immediate neighboring SMs (i.e., the SMs within 1-hop distance). These kind of communications would be needed in case of collective demand response or Plugin Electric Vehicles (PHEV) communications via SMs [1].

Our algorithm assumes the availability of shortest paths from each SM to the gateway due to use of IEEE 802.11s that already creates such paths. We first identify the leaf SMs which do not act as relays for the other SMs. Basically, the paths from all leaf SMs will include all the SMs’ paths to the gateway. When these paths are identified, we determine the shortest paths which share same SMs along their paths. We strive to group these paths within the same group since
the SMs on these paths have also high chance of interaction among themselves due to geographical closeness. To maintain some balance among the size of the groups, we put a threshold on the number of leaf SMs that can be added to a group when creating groups. The CA generates a separate CRL for each of these groups and the CRLs are distributed by the gateway to every SM. At the end of this process, we also identify SMs which have 1-hop neighbors belonging to groups different than themselves. These nodes exchange their CRLs and will hold multiple CRLs for different communication needs. We implemented the proposed approach under ns-3 [4] and assessed the performance with respect to traditional way of creating and distributing CRLs. The results indicate that the proposed approach can significantly reduce the CRL size with limited message overhead for distribution.

This paper is organized as follows. In the next section, we summarize the related work. Section III provides the preliminaries about the topic and defines the problem. Section IV presents the proposed approach along with the algorithm pseudo-code. In Section V, we assess the performance of the proposed approach. Finally, Section VI concludes the paper.

II. RELATED WORK

Certificate revocation is one of the most important components of the PKC. It has been studied in the context of the wireline Internet [5]. A survey and discussions on revocation methods’ tradeoffs is provided in [6]. Regarding the handling of CRLs in wireless networks, there is a number of works as detailed below.

For instance, Wasef et al. [7] have proposed a decentralized revocation scheme for vehicular ad hoc networks (VANETs). The scheme is based on a pairing-based threshold scheme and a probabilistic key distribution technique. It enables a group of well-behaving vehicles to revoke a misbehaving vehicle. Raya et al. [8] proposed a technique to revoke the tamper-proof device. The technique is useful when all the certificates of a vehicle are to be revoked. The CA sends a message to the tamper-proof device to command it to stop all the security functions. In [9], the authors argue that the PKI is essential to secure VANETs as it can fulfill most of the security requirements. To complement the security services provided by PKI, the authors introduce complementary security mechanisms that can meet location privacy, efficient authentication, and distributed and fair revocation. They also propose a mechanism for mitigating the effect of the denial of service attacks in VANETs.

In PKC-based authentication schemes, the revocation status of the message sender’s certificate should be verified before verifying the sender’s signature. In [10], the authors propose an expedite message authentication protocol. It replaces the time-consuming CRL checking process by an efficient revocation checking process using a keyed hash message authentication code (HMAC), where the key used in calculating the HMAC is shared only between non-revoked nodes. Crpeau et al. [11] propose a certificate revocation scheme for wireless ad hoc networks. The scheme not only provides a measure of protection against false accusation attacks, but it also aims to eliminate the window of opportunity whereby revoked certificates can be used to access network services. In [12], H. Guo et al. propose a batch authentication protocol for vehicle to smart grid communication. Instead of verifying individual packets, the aggregator waits for some time to receive multiple responses from a batch of vehicles. The aggregator verifies the received responses by only one signature verification.

Existing works on AMI network security such as [13] propose the use of a PKI and digital signatures but do not provide any mechanisms for certificate revocation, even though it is a required component of any PKI-based solution. Different aspects of certificate revocation problem in SG applications were discussed in [14] without providing a solution to SG neighborhood area networks (NANs). To the best of our knowledge, studying certificate revocation in SG NANs has not been well studied yet.

III. PRELIMINARIES

A. Underlying Infrastructure

SG has a communication infrastructure that consists of 3 major subnetworks: home area networks (HANs), NANs and WANs. Meter data collection and communication with the homes is done through the NANs by considering several wired and wireless underlying network technologies [15]. In this paper, we consider a NAN implemented using a wireless infrastructure based on IEEE 802.11s mesh networking standard. IEEE 802.11s is the standard for bringing multi-hopping capability to wireless LANs. The nodes in 802.11s mesh are given names based on their roles. All nodes are Mesh Points (MP) and are able to provide connectivity at the data link layer between other MPs. If an MP also provides connectivity to another network such as the Internet, it is called a Mesh Portal Point (MPP). An MP becomes a Mesh Access Point (MAP) if it provides access to wireless clients which are referred to as Mesh Station (Mesh STA).

In our NAN, all the smart meters (SMs) will act as MPs/MAPs. There will also be some additional nodes acting as relay MPs when there is no SM available. Note that we may have Mesh STAs from HANs that can connect with SMs and have them act as MAPs. The gateway node which will be connected to the utility will be MPP. The connection can be via WiMax or 4G. A CA is assumed to be communicating with the utility for certificate generation and management. A sample NAN is depicted in Fig. 1.

In this NAN, the MPs will send their readings to MPP using mesh path selection and forwarding mechanisms of 802.11s called Hybrid Wireless Routing Protocol (HWMP). The paths from each SM to gateway will be available and thus will be used in our grouping algorithm. Note that we use SG NAN and SG AMI Networks interchangeably throughout the paper.

B. Certificates and CRLs

Certificates are issued by a CA and sent to the gateway for distribution to SMs. When a certificate is issued, its validity is limited by an expiration date. Note that verification of the expiration date of a certificate is necessary but not sufficient
IV. GROUP-BASED APPROACH

A. Motivation

To understand the communication patterns among the SMs in a SG NAN, we consider the needs of three major applications: AMI, Demand Response (DR), and PHEV. In AMI, each SM sends its power reading to a gateway node periodically through multi-hop routes using the 802.11s-based NAN. In such a case, one major issue is that each SM and gateway need to ensure that the packet has been sent by legitimate sources. For instance, let us assume that $SM_A$ wants to send data to the gateway and the packet should be relayed by the SMs $SM_B$ and $SM_C$. After receiving the packet, the gateway needs to ensure that the packet has been sent by $SM_A$. Without this verification, any node can impersonate other SMs and send data under its name. In addition, relays $SM_B$ and $SM_C$ should verify that they are relaying packets for a legitimate node. Without this verification, outsider attackers can send bogus packets to overwhelm the network. For these purposes, $SM_B$ and $SM_C$ must verify $SM_A$’s signature. In this scenario, $SM_B$, $SM_C$, and the gateway need the revocation information of $SM_A$ so that if $SM_A$’s certificate is revoked, they should not relay or accept its packets. This leads us to the conclusion that these SMs on the same route to the gateway should have a CRL which will include all the SMs’ revoked certificate information. Therefore, it will be wise to put all these nodes in the same group when creating CRLs.

The communication pattern due to other mentioned applications can be different though. For instance, in case of DR or PHEV, the SMs will act as sources or relays when communicate with each other. For DR, an SM will need to communicate to its neighbors that are accessible possibly within 1-hop distance. For PHEV, an SM will route data to another SM in the neighborhood which will require communication with its 1-hop neighbors (and obviously with PHEVs). This

C. Problem Definition

The problem of certificate revocation introduces a lot of overheads with the increased network size. In SG, the SMs form a large-scale NAN and it is not feasible to have CRLs stored on a remote server for accessibility. Therefore, it is wise to distribute these CRLs to the SMs to accelerate the checking process. This indicates that a CRL should be available locally at every SM. Nonetheless, the communication pattern in SG AMI applications is in such a way that typically SMs do not communicate with each other frequently. SMs send power data to a data collector (i.e., gateway) or SMs exchange messages among themselves in 1-hop distance. Therefore, there will be a lot of SMs which do not communicate at all and thus do not need to check their revocation status. Hence, shorter CRLs could be used to save time.

Our problem can be defined as follows: “Given a NAN with $n$ SMs and their paths to a gateway, group the SMs in such a way that SMs within each group communicate with each other with the highest possibility”.

Note that there are two extreme cases which are not desirable in the context of SG.

- A single CRL is generated for the whole network (or city). The CA will generate only one signature but the CRL will be too long due to storing the IDs of revoked certificates. This can be broadcast to all SMs. Since the CRL size is long, this introduces additional processing and distribution delay, consumes a lot bandwidth for distribution and requires additional storage.

- One CRL for each of the SMs based on its neighbors. In this way, the CRL length will be equal to the number of neighbors of an SM and thus the processing delay will be very short. However, the CA needs to generate $n$ signatures if there are $n$ SMs in the network. The update and dissemination of this CRL introduces a lot of message overhead. In addition, this solution does not guarantee end-to-end verification among an SM and gateway since the only secure interaction will be among the neighbors.

We propose a solution that stays in the middle of the above approaches in terms of the CRL size. Our goal is to identify the group of nodes which will be likely to communicate and create a CRL for each group like this. In this way, we guarantee that SMs can communicate with others securely based on the needs of the applications. This creates a balance among the size of the CRL and the number of signatures needed from the CA as well.
necessitates creating groups within the neighborhood of an SM which will have CRLs for the members of that group only.

Combining these facts, we propose a grouping approach for CRL creation (i.e., one CRL for each group). Basically, the groups will include SMs sharing same routes. For handling the neighborhood communications, we will determine adjacent groups and distribute CRLs to the nodes which have links to multiple groups.

B. Overview of the Approach

The idea of our grouping approach is based on the routes from each leaf SM to the gateway. A leaf SM is the one that does not act as a relay for any of the other SMs in the network as part of the routes. We know that 802.11s standard already provides the routes from each SM to the gateway. Since leaf SMs are at the edge of the network, collecting the routes of these leaf SMs will cover the routes of every single SM in the network as shown in Fig. 2.

![Fig. 2: Creating groups based on the routes to the gateway.](image)

Our centralized algorithm which can be run at the utility company or gateway first starts by determining the routes of the leaf nodes in the NAN and stores all the routes with their lengths in descending order. Each route has a leaf SM as the starting node and lists the node ID of the next hop until the gateway. For each route in the list, our approach determines the other routes which share a node (i.e., branches) with this route. Our goal is to put the nodes which follow shared routes into the same group. Consider the example in Fig. 2. The leaf SMs in this network are numbered as SM2, SM3, SM7, SM10, SM12, SM13, SM14, SM16 respectively. The routes to the gateway from these leaf SMs are listed as follows:

- Route 2: 2-1-Gateway and length= 2 hops
- Route 3: 3-1-Gateway and length= 2 hops
- Route 7: 7-6-4-Gateway and length= 3 hops
- Route 10: 10-9-8-Gateway and length = 3 hops
- Route 12: 12-11-8-Gateway and length= 3 hops
- Route 13: 13-11-8-Gateway and length= 3 hops
- Route 14: 14-11-8-Gateway and length= 3 hops
- Route 16: 16-15-5-4-Gateway and length= 4 hops

We can see that Route 2 and Route 3 share node SM1. So we can group them together (Group 1). Similarly, Routes 7 and Routes 16 share node SM4 so we can also group them together too (Group 2). Finally, Routes 10, 12, 13 and 14 share the node SM8 which can be grouped under Group3.

**Algorithm 1 grouping(Routes[])**

1: Sort the Routes[]
2: For all Routes[i].GID = -1
3: cnt = length.Routes[]
4: while i < cnt do
5: if Route[i].GID == -1 then
6: Route[i].GID = i
7: Group[i]++ //Increase the group node count
8: j = i + 1
9: for all j < cnt do
10: if Route[j].GID < Threshold then
11: Route[j].GID = i
12: if length.Group[i] < Threshold then
13: Group[i]++
14: end if
15: end if
16: end for
17: i++
18: else
19: i++
20: end if
21: end while

To provide load balancing among the groups, we also define a threshold for the maximum number of leaf SMs that can be in a single group. In this way, we can prevent the cases where all the routes are shared and only one group is created. Obviously, this is not desirable since we are trying to reduce the size of the CRLs for efficiency. At the end of this process, each SM will have a group ID and for each group ID, a single CRL would be created at the CA.

The pseudo-code for grouping of the routes is provided in Algorithm 1. In this algorithm, Routes[] is an array of lists. Route[i] represents the i\(^{th}\) route with its elements being a list of SM IDs. Group[] is a hash map which has SM IDs and SM count in it. Group[i] represents the SMs which have the group ID as i.

C. Distribution of CRLs

While the distribution of CRLs is not a frequent process, it can still pose an overhead in particular if there is traffic on the NAN due to various activities in addition to AMI. Therefore, the distribution of CRLs should be done in an efficient manner. One obvious solution to distribute the CRLs is to send each SM via the paths available from the gateway. This means, the gateway needs to initiate a separate unicasting for each SM in the network. While this approach can introduce additional traffic, our group-based approach eliminates some of the unicasts since the SMs on the routes to the leaf SMs already share the same CRL with the leaf SM. Therefore, it is sufficient to unicast to each leaf SM rather than unicasting to every single SM in the network. The SMs on the routes will retain a copy of the CRL when forwarding it towards the leaf SMs. Referring to Fig. 2, unicasting the CRL to SM\(_{16}\) and
would be enough to reach other members of Group 2. There is no need to unicast separately to the other members of Group 2.

D. Multiple Groups

As a result of the grouping, there will be some groups which are totally disjoint from the rest of the network. This means none of the SMs in these groups have any links with the rest of the network. In fact, such a group is like a tree rooted at the gateway and would be a disjoint component when the gateway is removed from the network. An example group is shown in Fig. 2 where Group 1 is disjoint from the rest of the groups. For such groups, it totally makes sense to have a single CRL that includes the revocation information of only the SMs in that group.

However, this will not always be the case, especially if the network is in a densely populated area. There will be a lot of different groups which will have links among their members. In other words, there will be SMs which have neighbors that belong to a different group than themselves. For instance, in Fig. 2 Group 2 and 3 have a link that connects them (SM6 and SM9). In that case, when SM6 would like to communicate with SM9, checking the CRL of Group 2 will not be sufficient since the SM9’s removed certificates would be stored in Group 3. These cases would appear in DR and PHEV applications as mentioned before. Therefore, SM6 will need to have the CRL of Group 3 available locally.

To address this problem, we propose to use multiple CRLs at the SMs which are likely to communicate with SMs in other groups. Specifically, after the groups are created, each SM will communicate with its 1-hop neighbors and collect their group IDs. If there are SMs with different group IDs, then their CRLs will be exchanged among these SMs. If the groups IDs all match, then there is no need to exchange CRLs. This step will be performed once the grouping is completed. It will be done completely in a distributed manner and will only add 1 more message transmission to the CRL distribution overhead.

V. EXPERIMENTAL EVALUATION

A. Experiment Setup

We consider random NAN topologies consisting of varying number of SMs. All these topologies are connected topologies generated by a utility program developed in house. We assumed a transmission range of 120 m and an area of 800 m x 800 m. A gateway is selected randomly among the nodes which can be at any location. We assumed that IEEE 802.11s is the underlying protocol to provide communication among the SMs and the gateway. The proposed approach is developed under ns-3 simulator [4] which has a built-in implementation of 802.11s. The computations for grouping is done at the gateway and distributed via unicasting to each of the SMs. We opted to follow this distribution since current 802.11s standard does not support multicasting capability. Two sets of experiments were conducted to assess the grouping performance as well as the distribution performance. The experiments are repeated for 30 different topologies for significance.

B. Performance Metrics and Baselines

The following performance metrics are considered for assessing the proposed approach.

- **Size of the CRL:** This indicates the data size for the CRL. The size of the CRL is important for storage and search performance.
- **Signature Overhead:** This metric indicates the number of signatures performed by the CA.
- **CRL Message Overhead:** This indicates the number of messages required to distribute the CRLs.
- **CRL Delay:** This metric indicates the time it takes the CRL to be distributed to all SMs.

We compare our approach with two baselines. The first approach considers a single CRL for all the SMs in the network (S-CRL). The second approach considers a CRL based on the neighborhood list of each SM (N-CRL). Thus each SM will get a different CRL. Our approach is shown as Group-based CRL (G-CRL) in the experiment results.

C. CRL Size Comparison

We conducted experiments to measure the CRL size for different number of nodes under different topologies in the worst case for S-CRL the size is always 1. For N-CRL, the size is the number of neighbors of a node. We get the average of all the nodes in the network. The size of the CRL is based on the assumption that 20% of the nodes are compromised and thus we assumed 20% of the total SM count as the CRL size. The results are shown in Table I. Based on the results, we observe that G-CRL certificate size in average is significantly less than the current approach and N-CRL which is the most important metric in terms of storage overhead and search delay. Furthermore, the size of the CRL in G-CRL is almost fixed even if the network scales.

<table>
<thead>
<tr>
<th>Network size</th>
<th>S-CRL</th>
<th>N-CRL</th>
<th>G-CRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>20</td>
<td>4</td>
<td>1.4</td>
</tr>
<tr>
<td>200</td>
<td>40</td>
<td>6.8</td>
<td>1.4</td>
</tr>
<tr>
<td>300</td>
<td>60</td>
<td>9.6</td>
<td>1.6</td>
</tr>
<tr>
<td>400</td>
<td>80</td>
<td>11.4</td>
<td>1.4</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
<td>15.8</td>
<td>1.6</td>
</tr>
</tbody>
</table>

D. Signature Count Comparison

We also counted the number of signatures required at the CA for all approaches. The results shown in Table II indicate that our approach requires much less signatures than N-CRL. In addition, the signature count does not increase linearly which indicates its scalability.

E. Distribution Overhead

Finally, we conducted experiments to assess the distribution overhead of the approaches. We counted the number of messages required to distribute all CRLs to every SM as well as the three other performance metrics. These metrics are: 1) average
packet delay from gateway to all SMs; 2) completion time for CRL distribution; 3) packet delivery ratio (PDR) from gateway to all SMs; and 4) total throughput at all SMs. For S-CRL, we did not use broadcasting while it is an option to be implemented. This is because it cannot be used with TCP as we are using TCP for our approaches in order to guarantee packet delivery.

The message overhead for the three approaches is shown in Table III. Note that for G-CRL, we also considered the exchange of CRLs among the neighbors which added an additional message for each SM. The results indicate that G-CRL can reduce the message overhead compared to N-CRL and S-CRL due to using grouping.

To assess the performance in terms of the other metrics, we conducted experiments by creating a grid-based topology with 100 nodes due to space constraints. This is also to reduce interference among the nodes so that we can eliminate the effects of it on the performance metrics for fair comparison of approaches. The results are depicted in Table IV. According to these results, we observe that G-CRL can provide better delay and completion time than other approaches. Again this is due to the fact that our approach only transmits to leaf SMs. The throughput is also much less compared to others indicating that our approach consumes much less bandwidth for CRL distribution.

VI. CONCLUSION

In this paper, we proposed a group-based algorithm for creating and distributing the CRLs in an 802.11s-based NAN. The SMs that are part of the NAN are grouped based on the possibility of interaction which is mostly through the routes to the gateway node. Our approach strives to put all the SMs on the same path to the same group so that they use the same CRL when communicating. In addition, we allow an SM to maintain multiple CRL in order to communicate with its immediate neighborhood which is also highly possible due to characteristics of other applications on NAN.

We implemented the proposed G-CRL approach in ns-3 simulator that runs a draft version of 802.11s. The experiment results indicate that G-CRL can reduce the CRL size significantly which helps reducing the search delay and the storage overhead in resource limited SMs. In addition, group-based approach keeps the CRL distribution message overhead comparable to the traditional approach while reducing the delay for transmission of CRLs. The proposed approach does not bring any additional burden to the nodes since the computations can be done at the gateway or utility.

ACKNOWLEDGEMENT

This work is supported in part by US National Science Foundation under the grant number 1318872.

REFERENCES


<table>
<thead>
<tr>
<th>Approach</th>
<th>Average Delay (sec)</th>
<th>Completion Time (sec)</th>
<th>PDR (%)</th>
<th>Throughput (Kilobit/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-CRL</td>
<td>0.61</td>
<td>1.08</td>
<td>94.87</td>
<td>5.74</td>
</tr>
<tr>
<td>N-CRL</td>
<td>0.53</td>
<td>1.01</td>
<td>94.99</td>
<td>4.33</td>
</tr>
<tr>
<td>G-CRL</td>
<td>0.30</td>
<td>0.32</td>
<td>94.36</td>
<td>2.12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network size</th>
<th>S-CRL</th>
<th>N-CRL</th>
<th>G-CRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>229</td>
<td>229</td>
<td>241</td>
</tr>
<tr>
<td>200</td>
<td>511</td>
<td>511</td>
<td>503</td>
</tr>
<tr>
<td>300</td>
<td>765</td>
<td>765</td>
<td>757</td>
</tr>
<tr>
<td>400</td>
<td>1065</td>
<td>1065</td>
<td>1032</td>
</tr>
<tr>
<td>500</td>
<td>1366</td>
<td>1366</td>
<td>1307</td>
</tr>
</tbody>
</table>