Scalable Certificate Revocation Schemes for Smart Grid AMI Networks Using Bloom Filters

Khaled Rabieh, Mohamed Mahmoud, Kemal Akkaya, and Samet Tonyali

Abstract—Given the scalability of the Advanced Metering Infrastructure (AMI) networks, maintenance and access of certificate revocation lists (CRLs) pose new challenges. It is inefficient to create one large CRL for all the smart meters (SMs) or create a customized CRL for each SM since too many CRLs will be required. In order to tackle the scalability of the AMI network, we divide the network into clusters of SMs, but there is a tradeoff between the overhead at the certificate authority (CA) and the overhead at the clusters. We use Bloom filters to reduce the size of the CRLs in order to alleviate this tradeoff by increasing the clusters’ size with acceptable overhead. However, since Bloom filters suffer from false positives, there is a need to handle this problem so that SMs will not discard important messages due to falsely identifying the certificate of a sender as invalid. To this end, we propose two certificate revocation schemes that can identify and nullify the false positives. While the first scheme requires contacting the gateway to resolve them, the second scheme requires the CA additionally distribute the list of certificates that trigger false positives. Using mathematical models, we have demonstrated that the probability of contacting the gateway in the first scheme and the overhead of the second scheme can be very low by properly designing the Bloom filters. In order to assess the scalability and validate the mathematical formulas, we have implemented the proposed schemes using Visual C. The results indicate that our schemes are much more scalable than the conventional CRL and the mathematical and simulation results are almost identical. Moreover, we simulated the distribution of the CRLs in a wireless mesh-based AMI network using ns-3 network simulator and assessed its distribution overhead.

Index Terms—Keywords: Certificate revocation; public key cryptography; smart grid security; Public key infrastructure; AMI.

1 INTRODUCTION

The Smart Grid (SG) is a revolutionary upgrade to the existing power grid. It will have complex networks of intelligent electronic devices, distributed generators, and dispersed loads that require communication networks for management and coordination. It aims to enhance the current grid’s reliability and efficiency [1], [2], [3]. Among the many new applications SG provide, one major application is the Advanced Metering Infrastructure (AMI) which collect smart meter data using a two-way communication network. SMs can be connected via a wireless mesh network with one gateway serving as a relay between the meters and the utility company. The AMI network is used to monitor the power demands over short periods, provide more accurate billing as well as utilize dynamic pricing to facilitate the reduction of peak demand [4].

The primary security requirements for the AMI communications are identity and message authentication, message integrity, non-repudiation, accountability, and access control. The public-key cryptography (PKC) is the most practical and common cryptosystem that can achieve these security requirements [5], [6]. In PKC, public and private keys are issued for each SM. The public key should be widely known and the private key should be kept secret. The announcement of the public key is usually performed by using a signed document called a public key certificate that binds the certificate holder’s identity to its public key. The entity that signs the certificate is called the certificate authority (CA). The CA supports the creation, renewal, and revocation of the public key certificates. To authenticate a message, the SM signs the message with its private key and a digital signature algorithm. The verifier of the message’s signature first verifies the certificate, and then verifies the signature with the signer’s public key (extracted from the certificate) and signature verification algorithm.

When a certificate is issued, its lifetime is limited by an expiration date. After this date, the certificate is considered expired and should not be accepted by the SMs. However, there are several strong motivations that necessitate revoking certificates before the expiry date to secure the network such as private key compromise, excluding malicious meters, removal of meters from the network, etc. If the compromised key is not revoked, the attackers can use it to launch attacks under the name of the key’s holder. If a meter shows a malicious behavior, the meter’s certificate should be promptly revoked to protect the proper operation of the network. Once this is done, messages from this meter will be discarded. The certificates of malfunctioned meters that are removed from service should be revoked to prevent misusing their keys to launch attacks. Thus, SMs in a wireless mesh-based AMI network should discard the messages that are signed with revoked certificates. This requires a check to determine the certificate revocation status before accepting the signatures.

Online certificate status protocols (OCSPs) [7] and Certificate Revocation Lists (CRLs) are widely used approaches...
for certificate revocation [8], [9], [10]. In OCSP, an online and interactive certificate status server stores updated revocation information of the certificates of interest. When a node needs to check the status of a certificate, it sends a query packet to the server that replies with a response packet having the status of the certificate. For the CRL approach, the CA has to compose a list of revoked certificates’ serial numbers and distribute it to the nodes that need to verify the certificate holders’ signatures. To verify the status of a certificate, a node should check whether the certificate’s serial number is in the CRL that is stored locally.

In order to use the OCSP in AMI networks, it is costly to deploy new devices to act as OCSP server, so either the gateways or the CA should be used. It is insecure to allow the gateways to manage the certificate revocation because they are deployed in streets and unattended, and thus lack physical security. It is also inefficient and unscalable to use the CA as a certificate status server. The scalability of the AMI network will create a bottleneck at the CA and direct connections between the meters and the CA are required. The main advantages of the CRLs can be summarized as follows: 1) local verification: certificates can be verified locally without contacting the CA; and 2) scalability: if the overhead of distributing the CRLs is low, the CRL scheme becomes scalable. Therefore, we believe that CRLs are more appropriate to the AMI networks if the overhead is reduced. Although public key cryptography is used extensively in the literature to secure the AMI communications [4], the certificate revocation in AMI networks has received little attention in spite of its importance.

The AMI networks have the following special characteristics that necessitate novel techniques for certificate revocation. Limited physical security: the likelihood of compromising the SMs is high because they are unattended and accessible by attackers; scalability: the number of meters in AMI network is expected to be very large, and thus the number of certificates in the system is large; limited computational power/storage: cost-efficient SMs have limited storage and computational power, and thus it may be inefficient to store and search long CRLs; and limited bandwidth: the AMI network’s bandwidth is limited and it will be overwhelmed in transmitting too much data for different applications, and thus it is not efficient to distribute long CRLs.

It is inefficient to create a customized CRL for each meter having the certificates it needs because the number of different CRLs will be large and too much overhead will be required to create, sign and distribute them. On the other hand, given the scalability of the AMI networks, creating one CRL is inefficient because its size will grow significantly as time passes and thus becomes a major overhead in terms of distribution, search, and storage. In this paper, in order to tackle the scalability of the AMI network, it is divided into clusters of SMs. However, clustering alone is not enough, but the size of the certificate revocation information needs to be reduced. This is because there is a tradeoff between the computation/communication overhead at the CA and the communication/storage overhead at the clusters. This tradeoff can be controlled by the cluster sizes. Specifically, increasing the clusters’ size reduces the number of different CRLs that should be composed and distributed by the CA but increases the size of the CRLs which necessitates more overhead to distribute and store them. Similarly, reducing the clusters’ size increases the number of different CRLs that should be composed and distributed by the CA but decreases the size of the CRLs and thus reduces the overhead of distributing and storing them.

The paper aims to alleviate this tradeoff by increasing the clusters’ size with acceptable overhead. Specifically, instead of distributing deterministic CRLs, we use Bloom filter to distribute shorter probabilistic revocation lists. However, Bloom filters suffer from false positives, where a search may mistakenly indicate that a valid certificate is in the list of revoked certificates. This is not acceptable in AMI networks because meters may discard important messages such as a command to disconnect power in case of payment default. False positives can also cause financial losses if a meter misses power pricing information. Different from [11] and [9] that use Bloom filters for certificate revocation in vehicular ad hoc networks, we propose two Bloom filter based revocation schemes for AMI networks that can enable the meters to identify and nullify the false positives.

The first scheme, called two Bloom filter vectors based (TBFV) scheme, uses the fact that Bloom filters are free of false negatives, i.e., if a search indicates that a certificate is not in the list, this is certainly true. Instead of distributing only one Bloom filter vector for revoked certificates like [9], [11], the scheme distributes two fixed-size and short vectors: one for revoked certificates and another one for valid certificates. A certificate is valid if it is not found in the revoked certificate vector. Similarly, a certificate is revoked if it is not found in the valid certificate vector. If a certificate is found in the two vectors, this indicates that one of the filters triggered false positive. In this case, the meters should contact the gateway to return the correct certificate status. We use a Merkle tree to enable the gateway to provide efficient evidence for certificate revocation without contacting the CA or trusting the gateway. The tree is constructed by the CA and stored in the gateway. Resolving the false positives at the gateway level instead of the CA level can improve the scalability of the scheme. We derive a mathematical formula to the probability of contacting the gateway as a function of the two filters’ parameters. We demonstrate that this probability can be low by properly designing the Bloom filters.

In TBFV scheme, the SMs can identify the false positives locally and need to contact the gateway to resolve them. In some cases, the AMI network has too much traffic and the required communications to resolve the false positive may constitute a burden. Therefore, we propose a second scheme, called false positive certificate list based (FPCL), that can identify and resolve the false positives locally. The scheme distributes one Bloom filter vector for the revoked certificates and the list of valid certificates that can trigger false positives. To verify a certificate, a SM first checks if the certificate’s serial number is in the list. A certificate is valid if it is found in the list. If the certificate is not found, a SM checks whether the certificate is in the Bloom filter vector. The certificate is revoked if it is found in the vector, otherwise it is valid. We derive a mathematical formula for the expected communication overhead of the scheme.

In order to assess the scalability of the proposed schemes and validate the mathematical formulas, we have imple-
mented them and tested under different scenarios using Visual C. The results indicate that our schemes are much more scalable than the conventional CRL and the mathematical and simulation results are almost identical. In addition to this implementation, we have also incorporated the proposed schemes in a real AMI network that is based on IEEE 802.11s based wireless mesh networks [12]. We have used the ns-3 network simulator [13] and simulated the revocation information distribution in order to assess the distribution overhead on the network performance such as end-to-end delay.

To the best of our knowledge, this is the first work that employs Bloom filters for certificate revocation without false positives in smart grid AMI communications.

The remainder of the paper is organized as follows. In the next section, we discuss the network and threat models. The concepts of Bloom filters and Merkle tree are discussed in section 3. In sections 4 and 5, we present our proposed schemes. Section 6 is dedicated to security analysis and performance evaluation. In Section 7, we provide a summary to the related work. Finally, we conclude the paper in Section 8.

2 NETWORK AND THREAT MODELS

The assumed AMI network provides two-way communication between the utility company and the consumers’ SMs. The AMI network is divided into clusters and each cluster has a large number of neighborhood area networks (NANs). Each NAN has SMs and a gateway node (or a lead meter). The SMs communicate with the gateway using an IEEE 802.11s-based wireless mesh network [12] as shown in Fig. 1. The gateway communicates with the utility company and the CA via a long distance communication (e.g., 4G or WiMax). The gateway can initiate communication sessions to any SM, and similarly, every SM can initiate communication sessions to the gateway possibly via multi-hop routes. The gateway and each SM have a unique identity, public/private key pair and certificate. All the SMs know the gateway’s identity and public key. They also know the CA’s identity and public key.

The utility company will use the AMI network to collect fine-grained power consumption readings to perform state estimation to the power grid. The readings’ transmission rate can be less than 30 seconds for business premises and less than 15 minutes for residential ones [14]. The gateway relays some data to the meters, e.g., firmware updates and cryptographic and security data such as keys, renewed certificates, and revocation information from the CA. The gateway can also relay commands to the SMs, e.g., if a consumer defaults on the electricity bill, the utility can send a command to turn off the power [6].

AMI networks can be implemented with multi-hop wireless mesh topology. In this topology, each meter should act as a router to relay other meters’ packets. After receiving a packet, each meter should verify the sender’s signature and certificate to ensure that the packet is sent from a legitimate meter before relaying it. Without this check, external attackers can simply flood the network with bogus packets to disrupt the communications. There are applications that necessitate the communication of a SM with other SMs, e.g., in case of collective demand response [15]. Meters may also need to exchange information to optimally decide how to reduce the load.

For the threat model, some attackers attempt to manipulate the revocation information to mistakenly add valid certificate or remove revoked ones. This attack is serious in the sense that the attackers can exclude victim SMs from the network. The attackers may attempt to impersonate the gateway or the CA to transmit false revocation information. Attackers can replay valid packets or manipulate them if this can help achieve their goals. The gateways and SMs are not trusted but the CA is trusted.
3 Preliminaries

In this section, we explain the concepts of the Bloom filter and Merkle tree which are used in the proposed schemes.

3.1 Bloom Filter

The Bloom filter has been used in computing applications for different purposes [16], [17]. It can reduce the overhead of distributing and searching a set of items. The essential idea of the Bloom filter is illustrated in Fig. 2. The filter is a data structure (or bit array) that stores a set of elements compactly in a bit-vector \( B = \{ B_1, B_2, ..., B_m \} \) with length of \( m \) bits [18]. Initially, all the vector’s \( m \) bits are zeroes. Each certificate’s serial number \( SN_i \) to be stored should be hashed with \( K \) independent hash functions, \{ \( H_1() \), \( H_2() \), ..., \( H_K() \) \}. Each hash function maps a given certificate serial number to a number in the range \{0, 1, ..., \( m - 1 \)\} which represents the address of a bit location in \( B \). The result of hashing \( SN_i \) \( K \) times is a sequence of \( K \) hash values \{\( n_1, n_2, ..., n_k \)\} \( \in \{0, 1, ..., m - 1\} \). To store an element \( SN_i \), the bits that are pointed by the addresses resulted from the \( K \) hashes should be set.

To check whether a given certificate’s serial number is in the list of certificates of a Bloom filter vector, the certificate’s serial number is hashed by the \( K \) hash functions resulting in \( K \) bit locations in the vector. As illustrated in Fig. 3, the corresponding filter bits are checked: if all the bit locations are set, the certificate is in the list with some probability, otherwise it is definitely not in the list. This comes from the fact that a bit can be set to one multiple times when the hashes of different serial numbers point to the same location. This situation is called false positive and the probability that this occurs is called false-positive probability.

The false-positive probability of a Bloom filter can be driven as follows. The probability that any particular bit \( B_i \) has zero is \( (1 - 1/m)^{KN} \), since the location \( B_i \) must be avoided by all \( KN \) hash values, where \( N \) is the number of elements in the vector. Therefore, the probability that a particular bit is set \( Pr(B_i = 1) = 1 - (1 - 1/m)^{KN} \). In case of false positive, each of the \( K \) hash values must point at a bit location that is set to 1. The probability that this occurs is \( Pr(B_1 = 1 \text{ and } B_2 = 1 \text{ and } \ldots \text{ and } B_K = 1) \) which is \((1 - (1-1/m)^{KN})^K\). For more information on the computations of the Bloom filter, we refer to [19].

3.2 Merkle Tree

Fig. 4 shows a Merkle tree for eight revoked certificates with serial numbers \{\( SN_1, SN_2, ..., SN_8 \)\}. The eight leaf nodes are the hash values of the certificates’ serial numbers, i.e., \( h_i = H(SN_i) \forall i = 1,2, ..., 8 \), where \( H() \) is a one-way hash function [20]. The internal nodes are derived from their child nodes. For instance, the value of the node \( h_{3,4} \) is \( H(h_3|h_4) \), and the value of the root node is \( h_{1,8} = H(h_{1,4}|h_{5,8}) \), where \( | \) means concatenation. Each leaf node has a unique path to \( h_{1,8} \) that is called verification path (VP). Using this path, it can be verified that a leaf node belongs to the tree.

4 Two Bloom Filter Vectors Based Scheme (TBFV)

4.1 Overview of the Approach

The CA prepares the revocation information for each cluster. It first creates two Bloom filter vectors; one vector is for the serial numbers of the revoked certificates in the cluster and the second one is for the valid certificates. It also creates
4.2 Certificate Verification

In order to verify the status of a certificate, the SMs use the technique illustrated in Fig. 5. The technique makes use of the fact that Bloom filters are free from false negatives, i.e., if a certificate is not found in the filter, this is certainly true. The SM first checks whether the certificate’s serial number is in the revoked certificates’ Bloom filter vector. If the serial number is not in the vector, the certificate is certainly valid. Otherwise, the SM checks whether the certificate’s serial number is in the valid certificates’ Bloom filter vector. If it is not found, the certificate is certainly revoked. Finally, if the certificate’s serial number is found in the two vectors, this indicates that one of the Bloom filters triggered false positive and the SM should contact the gateway to query about the certificate status. It is impossible to not find a certificate’s number in the two vectors because the CA must have added it to one of them. For the same reason, it is impossible to find a certificate in the two vectors without false positive.

4.3 False Positive Resolution

When a meter inquires about the status of a certificate (i.e., false positive case), the gateway responds with “Valid” or “Revoked”, and the verification path of the Merkle tree. For instance, using the Merkle tree shown in Fig. 4, when a meter inquires the gateway about the status of the certificate $SN_i$, the gateway replies with the following verification path \{SN_2, h_{3,4}, h_{5,8}\}. The SM can verify the path by first computing $h_1 = H(SN_1), h_2 = H(SN_2), h_{1,2} = H(h_1|h_2),$ and $h_{1,4} = H(h_{1,2}|h_{3,4}).$ Then, it makes sure that $h_{1,8}$ is similar to $H(h_{1,4}|h_{5,8}).$ This is an undeniable proof that the CA has revoked the certificate because the root node is signed by the CA and it is infeasible to modify the tree after it is created. It is also infeasible to fabricate the CA’s signature. Since only the efficient hashing operations are required to verify the certificates, the computational overhead is low.

The advantage of using the Merkle tree is that the gateway can prove the status of a certificate without contacting the CA or trusting the gateway. Trusting the gateway is not required because the tree is constructed by the CA. Another advantage is that it can reduce the computation overhead on the SMs because they need to use the efficient hashing operations to verify the verification path and one signature verification during the lifetime of the tree.

4.4 Approach Analysis

Equations 1 and 2 give the false positive probabilities of the revoked and valid certificates Bloom filters, respectively. $N_r$, $K_r$, and $m_r$ are the number of revoked certificates, the number of hash functions used to compose the revoked certificates’ Bloom filter, and the bit vector size of the revoked certificates’ Bloom filter. Similarly, $N_v$, $K_v$, and $m_v$ are the number of valid certificates, the number of hash functions used to compose the valid certificates’ Bloom filter, and the bit vector size of the valid certificates’ Bloom filter.

$$FPR_r = (1 - (1 -\frac{1}{m_r})^{K_r N_r})^{K_r}$$  
(1)

$$FPR_v = (1 - (1 -\frac{1}{m_v})^{K_v N_v})^{K_v}$$  
(2)

The SMs fail to verify the certificates in case of false positive at either the revoked certificates’ filter or the valid certificates’ filter. The certificate verification failure probability ($P_f$) is given in Equation 3. $P_f$ equals to the probability that a certificate is revoked and the valid certificate filter triggers false positive or the certificate is valid and the revoked certificate filter triggers false positive. It can be seen that the failure probability can be determined by setting six parameters: the number of revoked and valid certificates ($N_r$ and $N_v$), the size of the Bloom vectors ($m_r$ and $m_v$), and the number of hash functions ($K_r$ and $K_v$).

$$P_f = Pr(\text{the certificate is revoked}) \times FPR_r + Pr(\text{the certificate is valid}) \times FPR_v$$  
(3)
In order to check a certificate’s revocation status, the SM stores the tree. To address these issues, in this section, we propose another scheme called False Positive Certificates List Based scheme (FPCL).

5.2 Overview
FPCL can identify and resolve the false positives locally without contacting the gateway. The CA generates one Bloom filter for the revoked certificates and the list of certificates’ serial numbers that trigger false positives ($C_l$). The Bloom vector is computed using the technique explained earlier. To compose the false positive certificates list, the CA has to check all the valid certificates. For each certificate, it hashes its serial number using the $K$ hash functions. The CA adds the certificate to the list if it is found in the Bloom vector, i.e., the corresponding $K$ bits in the filter are set, because the certificate triggers false positive. The CA signs the Bloom vector and the false positive certificate list and sends them to the gateways to distribute to the SMs. The SMs accept the revocation information if the signature verification passes.

In order to check a certificate’s revocation status, the SMs should use the procedure illustrated in Fig. 8. The SM first checks whether the certificate’s serial number is in the false positive certificate list. If it is found, the certificate is certainly valid. Otherwise, the SM hash the serial number with the $K$ hash functions to determine whether it is in the revoked Bloom filter. If all the corresponding bits in the vector are set, the certificate is certainly revoked, else, the certificate is valid. In this scheme, the Bloom filter is free of false positive because all the certificates other than the ones in the $C_l$ cannot trigger false positives.
Result of check?

Start

Check if SNi is in the Bloom filter

Result of check?

Check if SNi is in the false positive certificates list

Yes

No

SNi is valid

No

Yes

SNi is revoked

5.3 Approach Analysis

Equation 4 gives the average revocation information size \(R_t\) in Bytes that equals to the Bloom filter vector bit size \(m_r\) and the average size of the false positive certificate list \(C_l\). The average number of certificates in \(C_l\) equals to the probability of false positive of the Bloom filter \((FPR_r) \times \) the number of valid certificates \((N_v)\), where \(FPR_r = (1 - \frac{1}{m_r})^{K_r \times N_r}\). The size of each certificate’s serial number is 6 Bytes like the IP addresses in IPv6. In the evaluation section, we will compare this derived mathematical formula with simulation results.

\[
R_t = \frac{m_r}{8} + (1 - \left(1 - \frac{1}{m_r}\right)^{K_r \times N_r})^{K_r} \times N_v \times 6
\]  

(4)

Equation 4 indicates that the overhead has two components the Bloom filter size and the size of the \(C_l\). Fig. 9 shows how the two components contribute to the overhead as \(m_r\) increases. In the figure, \(N_v\) and \(N_r\) are 10,000 and 1,000, respectively. It can be seen that the increase of \(m_r\) decreases the size of \(C_l\) until it becomes negligible comparing to \(m_r\). This is because the size of \(C_l\) depends on the false positive probability and the increase of \(m_r\) reduces the false positive probability, i.e., fewer number of valid certificates can trigger false positives as \(m_r\) increases. From Fig. 9, we can conclude that there should be an optimal \(m_r\) that can minimize the overhead of this scheme.

Fig. 10 gives the relation between the revocation information in FPCL and \(m_r\) size. We select \(N_v\) and \(N_r\) to be 10,000 and 1,000 certificates, respectively. It can be seen that at small \(m_r\), the revocation information size is large because the false positive rate is high and thus the size of \(C_l\) is very large. The increase of \(m_r\) decreases the revocation information size. This is because the reduction rate in \(C_l\) (due to reducing the false positive rate) is much more than the increasing rate in \(m_r\). After the revocation information size reaches a minimum value, it starts to increase because the reduction rate in \(C_l\) becomes less than the increase rate of \(m_r\), and thus the increase of \(m_r\) increases the revocation information size. Choosing a good value for the Bloom filter vector is important to reduce the revocation information overhead comparing to the conventional CRL. From the figure, the minimum revocation information in FPCL is 1.8 KB when the size of \(m_r\) equals to 1.4 KB. Comparing to the CRL size that is 5.86 KB, it is clear that the overhead in FPCL can be only 32\% of the overhead in CRL, but if the value of \(m_r\) is not optimal, the overhead of FPCL can be more than the CRL.

Fig. 11 depicts how the optimum \(m_r\) changes when \(N_v\)
and $N_v$ increase, i.e., when the AMI network size increases. We select the values 2,000, 4,000, 6,000, 8,000 and 10,000 to $N_v$ and $N_r$ equals to 20% of $N_v$. The figure shows that $m_r$ should increase as $N_v$ and $N_r$ increase because they increase the false positive probability which increases the $C_l$ and the false positive probability and thus the $C_l$ size can be reduced by increasing $m_r$. When the network size changes, the CA has to update the Bloom filter size in the first certificate revocation update to stay operating at the optimal point. This can be done easily in our scheme because the revocation information is prepared by the CA and the meters are not involved at all. For example, from Fig. 11, when the number of valid certificates changes from 2,000 to 4,000, the CA has to increase the Bloom filter size from 0.45 to 0.95 Kbytes to operate at the optimal point.

Similar to the traditional CRL scheme, a cluster’s revocation information should be updated when certificates in the cluster are revoked. This should not affect the revocation information of other clusters. First, the CA should compute new Bloom filters. This computation is efficient because it needs only hashing operations. Then, in order to reduce the communication overhead, the CA does not need to send the new filters to the meters, but it can only send the locations of the bits that should change in the current Bloom filter to update it.

6 Evaluations

In this section, we evaluate the proposed schemes, TBFV and FPCL, and compare them with the conventional CRL scheme. We conducted simulations using Visual C to verify the mathematical formulas given earlier. NS-3 network simulator is used to study the impact of the revocation information distribution on the network performance. Finally, we analyze the security of the proposed schemes.

6.1 Overhead Evaluations

1. Communication Overhead

Fig. 12 gives the revocation information size of TBFV, FPCL and CRL at different values of $N_r$ and $N_v$, where $N_r = 0.1 N_v$. For TBFV scheme, we report the minimum revocation information that can keep the certificate verification failure probability below a given value. For FPCL scheme, we report the minimum revocation information. In order to calculate the minimum revocation information size, we initialize our experiment by a relatively small Bloom filter size. Then, the Bloom vector size is increased gradually and the revocation information size is recalculated until the minimum revocation information size is found. The results demonstrate that the revocation information sizes of our schemes are less than the CRL size.

When comparing TBFV with FPCL, it can be seen that FPCL has less revocation information size compared to TBFV with a verification failure probability ($P_f$) of 0.1. However, we observe that the revocation information size of TBFV is less than FPCL when $P_f$ increases to 0.15, but the meters need to contact the gateway more in TBFV. This is because the reduction of the revocation information size in TBFV increases the false positives that increase $P_f$. For example, TBFV scheme has revocation information sizes of 2.7 KB and 760 bytes to store 1,000 revoked certificates when $P_f$ is 0.1 and 0.15, respectively. On the other hand, FPCL’s revocation information size is 1.8 KB to store the same number of revoked certificates, but the CRL size is about 5.9 KB. The increase of the $N_r$ and $N_v$ increases the revocation information in TBFV to obtain the same $P_f$ limit. This is because the increase of $N_r$ and $N_v$ increases the false positive which increases $P_f$. To maintain the same $P_f$, $m_r$ and/or $m_v$ should increase to reduce the false positive probability. Similarly, the increase of the $N_r$ and $N_v$ necessitates increasing $m_r$ and/or $Cl$ to maintain the minimum revocation information. It can also be seen that TBFV can decrease the amount of revocation information with increasing the verification failure probability but the gateway will be contacted more.

Similarly, we repeat the simulation when $N_r = 0.2 N_v$. Fig. 13 demonstrates that the revocation information sizes of TBFV and FPCL are still less than the CRL. It can also be seen that FPCL has less revocation information size than TBFV with $P_f = 0.1$ and 0.15 which is different from the results given in Fig. 12. FPCL has minimum revocation information size when the number of revoked certificates is large. This indicates that FPCL is more scalable than the other schemes. It is noteworthy to mention that there is no additional communication overhead needed by FPCL since false positive cases can be resolved locally. Consequently, FPCL is suitable when the communication between the
meters and SMs is not fast or there is high traffic in the AMI network.

Fig. 12 and 13 depict how the schemes scale up in large networks. It can be seen that as \(N_r\) and \(N_v\) increase (in scalable networks), the difference between CRL and our scheme increases. This indicates that our schemes significantly outperform CRL in large networks. For example, when the number of certificates is 10,000 and \(N_r = 1000\), the revocation information sizes of TBFV \((P_f \leq 0.1)\), TBFV \((P_f \leq 0.15)\), and FPCL require 46%, 13% and 31% of the CRL size, respectively. Similarly, when the number of certificates is 10,000 and \(N_r = 2000\), the revocation information sizes of TBFV \((P_f \leq 0.1)\), TBFV \((P_f \leq 0.15)\), and FPCL require 45.8%, 35% and 26.6% of the CRL size, respectively.

When certificate verification fails in TBFV, the meter has to send a Certificate Status Inquiry (CSI) packet to the gateway. The gateway replies with Certificate Status Response (CSR) packet. CSI packet has the serial number of the certificate, the meter’s identifier, timestamp and signature. The total size of the packet is 72 bytes. The CSR packet should have the gateway signature, certificate serial number, time stamp, identity of the meter, and the verification path. The size of the CSR packet without the verification path is 73 bytes. The size of the verification path is the path length \(\times\) the size of the hash tags. For the signature scheme, we used the Elliptic Curve Digital Signature Algorithm (ECDSA). This was chosen since its overhead is minimal comparing to other public key cryptosystems. It also uses a short key size that is comparable to the current symmetric cryptographic schemes. It is an approved signature algorithm for the US government use [21].

2. Computational Overhead

When a signed message is received, the SM has to check if the certificate is revoked. The latency of this check should be minimized to expedite message authentication. In CRL scheme, the certificate verification latency equals to the time to find a certificate’s serial number in the CRL. This time depends on the location of the serial number in the list and also the list size. For our schemes, the certificate verification latency mainly depends on the computation time of the \(K\) hashing operations. This time is fixed and does not depend on the number of revoked certificates. Hash functions are very efficient in computation.

To evaluate the computational overhead of our schemes, we implemented a Bloom filter, using five hash functions based on SHA-1 and a vector of \(m = 9.77\) KB that stores 10,000 certificate’s serial numbers. The certificates’ serial numbers are generated randomly. Each serial number is 6 bytes. We used Crypto++ library [22] to implement the SHA-1. We also used a laptop with an Intel processor at 1.6 GHZ and 1 GB RAM to implement the filter. Although the computational power of a meter is less than a laptop, we scale the results with the factor of five to obtain a realistic estimation to the expected overhead.

Our measurement results demonstrate that a certificate can be added to the Bloom filter in 214 \(\mu s\). To measure the time to check whether a certificate is in the filter, we take the average to 30,000 certificates checks. Half of the certificates’ serial numbers were in the filter and the other half of them were not in the filter. We compute one hash value and check its corresponding bit. If it is one, we proceed and compute the next hash value, otherwise the certificate is not in the vector and we quit. The average certificate check time is around 175.48 \(\mu s\). This is less than the certificate addition time because \(K\) hashing operations should be performed to add a certificate, but once a hash value points at a location that stores zero, the search ends, so less than \(K\) hashes may be required to search for a certificate in the vector. The verification time of TBFV is more than FPCL because it needs two Bloom filter search operations but FPCL needs only one search operation. For composing the revocation information, TBFV and FPCL nearly take the same time because in both cases all the valid/revoked certificates should be hashed \(K\) times.

3. Merkle Tree

A tree of height \(h\) has \(N_v = 2^h\) leaves and \(N_r - 1\) interior nodes. If the tree is big and requires large storage area, the gateway can store only the certificates’ serial numbers and some nodes and computes the rest of the nodes when required. This is proper for our scheme because the tree may not be used very often during its lifetime. Computing the root node requires \(N_v\) hashing operations to calculate the leaves and \(2^h - 1\) hashing operations to compute the interior nodes. Several schemes have been proposed to balance between the storage area and the computational overhead of the Merkle tree. For instance, M. Szydlo [23] proposed an algorithm for Merkle tree traversal which requires only logarithmic space and time. For a tree with \(N_r\) leaves, the algorithm computes sequential tree leaves and verification path in time \(2\log_2(N_r)\) and space less than \(3\log_2(N_r)\), where the units of computation are hash function evaluations or leaf value computations, and the units of space are the number of node values that should be stored. Creating the Merkle tree needs efficient hashing operations and one signature by the CA. The path length equals to \(\log_2(N_r)\), where \(N_r\) is the number of revoked certificates. The number of hashing operations required to verify the path equals to the path length plus one.

6.2 Mathematical Models Verification

We have used Visual C to implement the proposed schemes. To implement the Bloom filters, we used SHA-1 hash func-
tion [24] to compute the bit location resulted from hashing the certificates’ serial numbers. However, SHA-1 produces a digest of 160 bits which is much more than what we need. The Bloom filter’s hash values should have \(\lceil \log_2(m_r) \rceil\) output bits, where \(m_r\) is the number of bits in the Bloom filter vector. To reduce the SHA-1’s output to the desired number of bits, we used the approach illustrated in Fig. 14. We strive to keep the random and unbiased nature of SHA-1. We divide the SHA-1 output to blocks of \(\lceil \log_2(m_r) \rceil\) bits. Padding bits that can be zeros or ones are added if the last block is less than \(\lceil \log_2(m_r) \rceil\) bits. Then, we apply exclusive OR (XOR) bitwise operations to the blocks to calculate the \(\lceil \log_2(m_r) \rceil\) bit hash value. If the SHA-1 output is random (as expected), the resultant \(\lceil \log_2(m_r) \rceil\) bit hash value should be random as well because it is computed from it. To derive the \(K\) hash functions that are used in the Bloom filter using SHA-1, we used the technique used in [19]. \(K\) hash functions can be driven from SHA-1 by appending a specific random value to the input, e.g., \(H(r_1|SN_1), H(r_2|SN_1), \ldots\), and \(H(r_K|SN_1)\). Each function has a different random value that is fixed and known to the meters.

We setup an experiment to compare the certificate verification probability in TBFV with the mathematical formula given in Equation 3. We consider two cases where \(m_r = 4\) Kbytes and 8 Kbytes. The number of valid certificates is 9,000 and the number of revoked certificates is 1,000. We first generate the certificates’ serial numbers randomly. Then, we add the revoked certificates to one Bloom filter and add the valid certificates to another filter. We count the number of times a meter fails to verify the certificates’s status locally and divide it by the total number of verifications. We run our experiment 20 times and report the average value. Fig. 15 gives the certificate verification probability at different values of \(m_r\) for both the mathematical and simulation results. The figure demonstrates that the experiment and mathematical results are almost identical.

We also setup an experiment to compare the minimum revocation information size of FPCL with the mathematical formula given in Equation 4. We consider two cases where \(N_r = 0.1\) \(N_v\) and \(N_r = 0.2\) \(N_v\). We run our experiment 20 times and give the average value. For the experiment and mathematical equation, we calculate the minimum revocation information by starting with a small \(m_r\) and gradually increase it until the minimum revocation information is found. Fig. 16 gives the minimum revocation information size at different number of valid certificates for both the mathematical and simulation results. The figure demonstrates that the experiment and mathematical results are almost identical.

### 6.3 Network Simulation Results

The proposed schemes have been implemented in a more realistic setting where the underlying communication infrastructure is assumed to be based on 802.11s-based wireless mesh networks (WMNs) [25]. The simulations are performed under ns-3 which has a built-in implementation of 802.11s. We created a scenario where the CA divides the AMI network into clusters of NANs and each NAN is led by a separate gateway as shown in Fig. 1. Each gateway is assumed to distribute the revocation information to each

![Fig. 15: Comparing the math results with the simulation results in TBFV scheme.](image)

![Fig. 16: Comparing the math results with the simulation results in FPCL scheme.](image)

of the SMs in its NAN. Each NAN topology is generated randomly by using NPART topology generation tool [26]. All topologies are connected (i.e., each SM can reach the gateway). We generated 30 different topologies for each NAN size and got the average of the results for statistical significance.

In order to assess the impact of the data traffic on the revocation information distribution, we assumed that all the SMs are sending their power readings to the gateways at the time of distributing the revocation information. The SMs are sending their power readings at every 10 seconds which is used by some utilities in the real-life applications of AMI [27]. We assume a transmission range of 120m for each of the SMs. The underlying MAC protocol used was 802.11g, and we implemented TCP as transport layer protocol for reliability. The simulations were run for 100 seconds. The results are the snapshot at the end of 100th sec.

In our simulations, we tested two different NAN sizes: 40 and 80 meters. The size of the cluster is also a parameter which is assumed to be 6,000 and 10,000 meters for testing scalability. For any case, we considered two scenarios for the number of revoked certificates: 10% and 20% of the total number of valid certificates in the network. This means we distributed a list of 600 & 1,200 revoked certificates for the cluster size of 6,000 meters and 1,000 & 2,000 revoked certificates for the cluster size of 10,000 meters. These revocations lists are distributed by each gateway to all of the members of their NANs.

We considered the following three performance metrics
to assess the network performance when the revocation information is to be distributed:

- **End-to-end (ETE) Delay**: This metric indicates the average time it takes for a revocation information packet to reach a SM.
- **Packet Delivery Ratio**: This metric indicates the ratio of the number of all packets sent from the gateway to the number of all packets received at the SMs.
- **Throughput**: This metric indicates the amount of data received at the SMs per second. This metric will show how much data is transmitted in the network and thus hint about the bandwidth usage.

The results for all the different cases and approaches (i.e., TPFV with different probability of failures depicted as TPFV $P_f$, FPCL, and CRL) are shown in Table 1.

As can be seen from these results, the PDR values do not change much under different approaches when NAN size is taken as 40 SMs. However, when the NAN size is 80 SMs and the cluster size increases, this makes a significant impact on PDR under different approaches. This is mainly due to increasing size of revocation information. As the size of the revocation information increases beyond a certain size, the TCP protocol needs to apply fragmentation as there is a maximum size a connection can accommodate (e.g., MTU of 1,500 bytes). This in turn creates more fragments for each packet that need to be transmitted separately. This results in a congestion in the NAN, and an increase in the number of dropped packets. For instance, networks with 20% revoked certificates is affected significantly by this congestion. The PDR drops almost the half of that of 10% revoked certificates case. However, we can see that the proposed FPCL approach can keep up with the increased revocation information size and maintains the same level of PDR performance. It also provides the highest PDR compared to others especially when the cluster size is 10,000 SMs.

The most noticeable conclusion out of these results are on the ETE delay performance of the proposed approaches. As the size of the revocation list increases the number of fragments created by the TCP protocol increases as well. This results in an increased contention by the SMs to access the channel. In addition, as the number of dropped packets increases, the number of retransmissions by the gateway increases as well. All of these would contribute to the increased ETE delay for the packets. In all cases, TBFV with 0.15 verification failure and FPCL show the best ETE delay performance for 10% and 20% certificate revocation respectively because the size of revocation lists they use is the least amongst the others.

Overall we can claim that TBFV $P_f = 0.15$ and FPCL can be used to distribute revocation information with the minimal impact on the network performance. They significantly improve the network performance in terms of ETE delay an PDR compared to the existing CRL method.

### 6.4 Security Analysis

It is not secure to allow the nodes to communicate when the updated revocation information is not available. This means that the nodes cannot communicate when the certificate revocation scheme fails. In addition, if the information is manipulated, attackers can sabotage the communications,
e.g., by adding valid certificates to the list or removing revoked ones. In TBFV, the two filter vectors and the root of the Merkle tree are timestamped and digitally signed by the CA. Modifying or fabricating them is impossible without knowing the CA’s private key. The attackers may try to change the vectors and/or the Merkle tree root before relaying the packet sent from the CA to the victim meters. However, any change in them will fail the CA’s signature verification. Similarly, the Bloom filter vector and the false positive certificate list in FPCL are signed to ensure the security of the distribution of the revocation information. If an attacker uses the certificate of a meter in a different cluster, the certificate’s serial number will not be found in the revoked Bloom filter vector in TBFV and the meters may accept it. Therefore, the CA should include the cluster’s identifier in the certificates and the meters should verify the identifier before verifying its revocation. The CA should ensure that the Bloom filters of a cluster store all the cluster’s certificates.

To thwart replay attacks, the timestamps can be verified to ensure the freshness of the packets. To break our schemes, attackers may try to impersonates the gateway or the CA and send fabricated packets to the smart meters. Digital signatures are used to secure the communications between the meters and the gateway. The attackers cannot impersonate the gateway or the CA because they can not compute correct signatures. With using secure hash function, Merkle tree cannot be manipulated because it is impossible to find two different certificate serial numbers that can generate the same hash value. Since the tree is authenticated by digitally signing its root, it cannot be modified without invalidating the digital signature.

7 RELATED WORK

In [28], Khurana et al. have discussed the main security issues in SG. The authors have identified public key management as a challenge due to the system scalability and complexity. In [6], a protocol is proposed to ensure that customers who default on their payments can be switched off remotely. In order to prevent attackers from launching attacks to interrupt the citizens’ electricity supply, the scheme uses public key cryptography to secure the protocol without addressing certificate revocation. Metke et al. [29] survey the existing key security technologies for extremely large and wide-area communication networks and study their applicability for the smart grid. Based on studying the security requirements as well as the scale of the smart grid, the authors strongly believe that the PKC is the most effective key management solution for securing the SG.

Existing works on AMI network security such as [4] propose the use of PKC but do not provide any mechanisms for certificate revocation, although it is a required component. Different aspects of certificate revocation problem in SG applications were discussed in [30] without providing a solution to AMI networks.

In [31], Wohlmacher et al survey the different revocation methods. Zhao et. al. [32] propose an efficient key management scheme with key revocation in smart grid communications. Instead of the large computation cost of signature verification needed in PKI communications, the scheme only requires to perform Media Key Block (MKB) process to extract the shared key. In MKB process, broadcast encryption [33] is used to recover the shared key between different devices. The authors study the revocation of malicious devices’ keys where new keys are computed at the Key Distribution Center (KDC) and distributed to non-malicious users. The revoked users will not be able to decipher the shared key without updating the initial parameters of each device. The authors also compare the proposed scheme with the internet key exchange methodology in PKI to show the superiority of their scheme in terms of communication and computation cost. In vehicle-2 grid (V2G) communications, Vaidya et al. [34] propose an efficient PKI infrastructure to overcome the traditional X.509 based PKI shortcomings. These shortcomings are the large size of traditional PKI certificates and long verification time. The proposed infrastructure is based on elliptic curve cryptography and self-certified public key technique to reduce the size of the certificate and verification time.

Chan et. al. [35] propose an enhanced PKI mechanism to resist the Denial of Service (DoS) attack in wireless smart grid communications. In the domain of smart grid, a DoS attack can prevent legitimate smart meters from connecting to a node if an attacker keeps sending fake signatures and certificates because the signature verification computation cost is high. For this reason, a lightweight signature verification scheme is proposed for authenticating different smart grid devices.

Akkaya et al. [36] propose an efficient grouping algorithm to distribute CRLs in an 802.11s based AMI networks. The grouping algorithm is based on the fact that SMs are communicating with the gateway to send/receive utility information. Each SM uses the same route to communicate with the gateway i.e. communicate with the same group of SMs for forwarding its packets. Instead of storing a large CRL in each SM, SMs only keep the CRL of its group to minimize the communication and storage overhead of CRL.

In VANETS, revocation of malicious vehicles is extensively studied. In that sense, it is noteworthy to discuss the revocation schemes employed in VANETS. In [37], Raya et al. propose a scheme to isolate misbehaving nodes in VANETs until a centralized revocation is issued by the CA. A group of neighboring nodes perform a voting on the misbehavior of a specific node. If the accumulation of votes exceeds a predefined threshold, then a warning message is broadcasted to the neighboring vehicles to inform them to ignore all the messages transmitted by the misbehaving nodes. Raya et al. [11] have investigated certificate revocation problem in VANETs. They argue that low false positive is acceptable for most VANETs’ applications. Unlike VANETs, false positives will not be acceptable in AMI networks because meters may miss important messages such as commands to disconnect power and power pricing information. In [9], Haas et al. proposed a scheme to distribute CRLs quickly even during the incremental deployment of VANETs. They also proposed a revocation scheme for anonymous communications, where each vehicle uses several certified pseudonymous instead of a unique identity to preserve its privacy. The authors argue that a vehicle can avoid triggering a false positive by discarding any of its own certificates that will trigger a false positive before it is ever sent, thus preventing a receiver
from generating a false positive. Obviously, this cannot be applied to AMI because each meter will use one certificate. Different from [9], [11], our paper studies using Bloom filter for AMI networks and proposes a novel scheme to mitigate the false positives.

Papadimitratos et al [38] propose a scheme to distribute large CRLs efficiently across wide regions in VANETs. In [39], a scheme is proposed to distribute the load of a server to a set of participating clients. The authors introduce a new class of graphs that support efficient and fault-tolerant revocation. In [40], Wasef et al propose a distributed certificate-service scheme for VANETs. The scheme aims to offer flexible interoperability for certificate service in heterogeneous administrative authorities and reduce the complexity of certificate management. In [41], Wasef et al propose an expedite message authentication protocol for VANETs. To authenticate a message, vehicles should check if the certificate of the sender is included in the CRL. The proposed protocol replaces the time-consuming CRL checking process by an efficient revocation checking process.

8 CONCLUSION
We have proposed certificate revocation schemes for large scale AMI networks. The network is divided into clusters of SMs, and in order to alleviate the tradeoff between the overhead at the certificate authority (CA) and the overhead at the clusters, we used Bloom filters to reduce the size of the CRLs in order to alleviate this tradeoff by increasing the clusters’ size with acceptable overhead. To address the false positive issue of the Bloom filter, we proposed two revocation schemes called TBFV and FPCL to identify and resolve the false positives. In order to evaluate the performance of our schemes, mathematical model, Visual C, and ns-3 simulation results have been presented. The results have demonstrated that our mathematical results are almost identical to the simulation results. Our evaluations demonstrated that our schemes are much more efficient than the conventional CRL approach. In most cases, our schemes require less than 33% of the overhead required in the CRL approach. Overall we can claim that TBFV $P_f = 0.15$ and FPCL can be used to distribute revocation information with the minimal impact on the network performance. They significantly improve the network performance in terms of ETE delay an PDR compared to the conventional CRL method.

9 ACKNOWLEDGEMENT
This work is supported in part by US National Science Foundation under the grant number 1319087.

REFERENCES
Mohamed M. E. A. Mahmoud received PhD degree from the University of Waterloo in April 2011. From May 2011 to May 2012, he worked as a postdoctoral fellow in the Broadband Communications Research Group - University of Waterloo. From August 2012 to July 2013, he worked as a visiting scholar in University of Waterloo, and a postdoctoral fellow in Ryerson University. Currently, Dr Mahmoud is an assistant professor in Department Electrical and Computer Engineering, Tennessee Tech University, USA. The research interests of Dr. Mahmoud include security and privacy preserving schemes for smart grid communication network, mobile ad hoc network, sensor network, and delay-tolerant network. Dr. Mahmoud has received NSERC-PDF award. He won the Best Paper Award from IEEE International Conference on Communications (ICC'09), Dresden, Germany, 2009. Dr. Mahmoud is the author for more than twenty three papers published in major IEEE conferences and journals, such as INFOCOM conference and IEEE Transactions on Vehicular Technology, Mobile Computing, and Parallel and Distributed Systems. He serves as an Associate Editor in Springer journal of peer-to-peer networking and applications. He served as a technical program committee member for several IEEE conferences and as a reviewer for several journals and conferences such as IEEE Transactions on Vehicular Technology, IEEE Transactions on Parallel and Distributed Systems, and the journal of Peer-to-Peer Networking.

Dr. Kemal Akkaya is an associate professor in the Department of Electrical and Computer Engineering at Florida International University. He received his PhD in Computer Science from University of Maryland Baltimore County in 2005 and joined the department of Computer Science at Southern Illinois University (SIU) as an assistant professor. Dr. Akkaya was an associate professor at SIU from 2011 to 2014. He was also a visiting professor at The George Washington University in Fall 2013. His current research interests include security and privacy, energy aware routing, topology control, and quality of service issues in a variety of wireless networks such as sensor networks, multimedia sensor networks, smart-grid communication networks and vehicular networks. Dr. Akkaya is a member of IEEE. He is the area editor of Elsevier Ad Hoc Network Journal and serves on the editorial board of IEEE Communication Surveys and Tutorials. He has served as the guest editor for Journal of High Speed Networks, Computer Communications Journal, Elsevier Ad Hoc Networks Journal and in the TPC of many leading wireless networking conferences including IEEE ICC, Globecom, LCN and WCNC. He has published over 90 papers in peer reviewed journal and conferences. He has received “Top Cited” article award from Elsevier in 2010.

Samet Tonyali is currently a Graduate Research Assistant in Computer Science Department at Southern Illinois University Carbondale, USA and pursuing his Ph.D. degree in the same department. He received the B.S. degree and the M.S. degree in Computer Engineering from Marmara University, Istanbul, TURKEY in 2011 and 2013, respectively.

Khaled Rabieh received the B.Sc. degree in computer science from Ain-shams University, Cairo, Egypt, in 2004 and the M.Sc. degree in communications and information technology from Nile University in 2012. He is currently a research assistant and working toward the Ph.D. degree at the Center for Manufacturing Research, Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, Tennessee, United States.