2-2021

Communication-efficient Certificate Revocation Management for Advanced Metering Infrastructure and IoT Integration

Mumin Cebe

Kemal Akkaya

Future Generation Computer Systems, Vol. 115 (February 2021): 267-278. DOI.

Communication-Efficient Certificate Revocation Management for Advanced Metering Infrastructure And IoT Integration

Mumin Cebe
Computer Science Department, Marquette University, Milwaukee, WI
Kemal Akkaya
Department of Electrical and Computer Engineering, Florida International University, Miami, FL

Abstract

Advanced Metering Infrastructure forms a communication network for the collection of power data from smart meters in Smart Grid. As the communication between smart meters could be secured utilizing public-key cryptography, however, public-key cryptography still has certain challenges in terms of certificate revocation and management particularly related distribution and storage overhead of revoked certificates. To address this challenge, in this paper, we propose a novel revocation management approach by utilizing cryptographic accumulators which reduces the space requirements for revocation information significantly and thus enables efficient distribution of such information to all

smart meters. We implemented the proposed approach on both ns-3 network simulator and a testbed. We demonstrated its superior performance with respect to traditional methods for revocation management.

Keywords

Smart grid security, Critical infrastructures security, Cyber–physical systems, Certificate revocation, Internet-of-Things (IoT), Cryptographic accumulators, Authentication-authorization-accounting and key management

List of Abbreviations

AMI Advanced Metering Infrastructure
CA Certificate Authority
CRL Certificate Revocation List
DHT Distributed Hash Table
DMZ Demilitarized Zone
HES Head-End System
HMS Head-End Management Server
MAC Medium Access Control
OCSP Online Certificate Status Protocol
NIST National Institute of Standards and Technology
PKI Public Key Infrastructure
RSA Rivest–Shamir–Adleman

1. Introduction

The existing power grid is currently going through a major transformation to enhance its reliability, resiliency, and efficiency by enabling networks of intelligent electronic devices, distributed generators, and dispersed loads [1], which is referred to as *Smart(er) Grid*. Advanced Metering Infrastructure (AMI) network is one of the renewed components of Smart Grid that helps to collect smart meter data using a two-way communication [2]. Smart meters and integrated Internet-of-Things (IoT) devices are typically connected via a wireless mesh network with a gateway (or access point) serving as a relay between the meters and the utility company.

The security requirements for the AMI network are not different from the conventional networks as confidentiality, authentication, message integrity, access control, and non-repudiation are all needed to secure the AMI. Confidentiality is required to prevent exposure of customer's private data to unauthorized parties while integrity is necessary to ensure that power readings are not changed for billing fraud. Furthermore, authentication is crucial to prevent any compromised smart meters communicating with other smart meters. On the other hand, the National Institute of Standards and Technology (NIST) urges to use Public Key Infrastructure (PKI) for providing the security requirements of AMI [3]. As an example, companies such as Landis&Gyr and Silver Spring Networks already use PKI to provide security for millions of smart meters in the US [4]. In such a PKI, the public-keys for smart meters and utilities are stored in *certificates* which are issued by Certificate Authorities (CAs). The

employment of PKI in AMI requires management of certificates which include the creation, renewal, distribution and revocation. In particular, the certificate revocation and its association with smart meters are critical.

1.1. Problem description and existing solutions

Several reasons necessitate revoking certificates, such as key compromise, certificate compromise, excluding malicious meters, renewing devices, etc. Besides, if there is a vulnerability in the algorithms or libraries that are used in certificate generation, a massive number of revocations may additionally occur. For instance, a recent discovery of a chip deficiency on RSA key generation caused revocation of more than 700K certificates of devices that deployed this specific chip [5] and renowned heartbleed vulnerability caused the revocation of millions of certificates, immediately [6]. Thus, to establish secure communication, a smart meter should check the status of the other smart meter's certificate against a certificate revocation list (CRL) that keeps all revoked certificates. Considering the large number of smart meters in an AMI and the fact that the expiration period can be even lifelong in particular applications [4], the CRL size will be huge. Consequently, revocation management becomes a burden for the AMI infrastructure which is typically restricted in terms of bandwidth. This overhead is particularly critical since the reliability and efficiency of AMI data communication are crucial for the functionality of the Smart Grid. Considering the potential impact on the performance of AMI applications [7], handling the overhead of revocation management is essential.

Certificate revocation management is commonly handled by utilizing CRL that is stored in the smart meter. The status of a smart meter is determined by checking whether its certificate is listed in the CRL or not. An alternative method would be to store the CRL in a remote server as in the case of Online certificate status protocols (OCSPs) [8], [9]. In OCSP, an online and interactive certificate status server stores revocation information. Thus, each time a query is sent to the server to check the status of the certificate. While OCSP-like approaches can be advantageous on Internet communications, employing them for AMI is not attractive since it will require access to a remote server for each time. In this regard, another alternative would be to use OCSP *stapling* [9] where the smart meters query the OCSP server at certain intervals and obtain a signed timestamped OCSP response which is included ("stapled") in the certificate. Again, this approach also needs frequent access to a remote server. Moreover, the 'stapled' certificates should be downloaded frequently by smart meters to ensure security, and this will create additional traffic overhead on the AMI which affects applications such as demand response or outage management.

1.2. Our approach and contributions

In this paper, we propose a communication-efficient revocation or CRL management scheme for AMI networks by using RSA accumulators [10]. RSA accumulator is a cryptographic tool which is able to represent a set of values with a single accumulator value (i.e., digest a set into a single value). Also, it provides a mechanism to check whether an element is in the set or not which implicitly means that cryptographic accumulators can be used for efficient membership testing. Due to the attractiveness of size, in this paper, we adapt RSA accumulators for our needs by introducing several novel elements as following:

- An accumulator manager is introduced within the utility company (UC) that is tasked with collection of CRLs from CAs and accumulating these CRLs (i.e., revoked certificates' serial numbers) to a single accumulator value which will then be distributed to the smart meters.
- We also introduce a non-revoked proof tuple for allowing a smart meter to check whether another meter's certificate is revoked without referring to the CRL file.
- We defined additional entities within AMI and assign functions to them to govern an accumulator based revocation management.
- We introduced several security countermeasures against possible attacks to a accumulator-based scheme.

The computation and communication related aspects of the proposed approach is assessed via simulations in ns3 network. In addition, we built an actual testbed using in-house smart meters to assess the performance realistically. We compared our approach with the other methods that use conventional CRL schemes and Bloom-filters [11]. The results show that the proposed approach significantly outperforms the other existing methods in terms of reducing the communication overhead that is measured with the completion time. The overhead in terms of computation is not major and can be handled in advance within the utility that will not impact the smart meters.

This paper is organized as follows: In the next two sections, we summarize the related work and the background. Section 4 introduces the threat model. Section 5 presents the proposed approach with its features. Sections 6 Evaluation of the approach and its objectives, 7 Performance evaluation are dedicated to evaluation criteria and experimental validation. Section 8 analyzes the security of the approach. Section 9 discusses the benefits and limitations. The paper is concluded in Section 10.

2. Related work

Due to increasing threats towards Smart Grid, there has been a number of efforts to adapt PKI for Smart Grid communication infrastructure. For instance, Metke et al. [12] surveyed the existing key security technologies in Smart Grid domain by mainly focusing on PKI. On the other hand, the study [13] stressed the importance of revocation overhead of PKI in Smart Grid. Beyond directly related studies on the PKI and Smart Grid relation, we also focus on studies about cryptographic accumulators and membership management. In this section, we examine the relation between this study and previous studies and highlight major differences.

2.1. Revocation management in AMIs

The studies [7] investigated different revocation management aspects such as short-lived-certificate scheme, tamper-proof device scheme, Online Certificate Status Protocol (OCSP), conventional CRL, and compressed CRL. However, this study just hypothetically analyzed the applicability of existent revocation solutions for AMI. The first offered approach that focused on reducing the revocation management overhead for AMI was based on Bloom Filters [14]. They provided a Bloom Filters based scheme particularly to reduce the size CRL.

However, since Bloom Filters suffer from false positives, the approach requires accessing the CA to check the validity of a certificate. Our proposed scheme, on the other hand, never requires accessing a remote server and provides a better reduction on CRL size. The study in [15] use distributed hash

tables (DHT) to reduce the CRL size again. Although this study provides a reduction in CRL size, it suffers from additional inter-meter communication overhead for accessing the CRL information.

We would like to note that a very preliminary version of this work was published in [16]. In this work, we improved the various aspects of the previous one. First, we improved computation performance utilizing Euler's Theorem. Second, we extended our threat model to new attack types that were not considered in the conference version. In this regard, we changed our approach in several ways: We proposed to use an initial secret during accumulation. We then introduced a non-revoked proof concept that was not used before in any of the revocation works. This required major changes to the accumulation process which was not in [16]. We finally proposed an extensive certificate verification protocol as countermeasures to the new threats. This also required proposing a new secure multi-level AMI architecture as opposed to the monolithic architecture used in [16]. In addition, we added several new experiments with accumulator computation overhead under various assumptions.

2.2. Cryptographic accumulators

Benalog and DeMare [17] first introduced cryptographic accumulators. After their first appearance, there have been studies [10], [18], [19] offering to use them for membership testing. However, these studies solely focused on building the cryptographic fundamentals of accumulators, and thus, omit application-specific issues and security features when deploying them. Besides, these studies are offering to use accumulators for membership testing by accumulating a valid list. Considering AMI, accumulation of valid smart meter's certificates to provide a revocation mechanism would constitute a significant overhead due to the fact that revocation frequency is less than that of creating new certificates (i.e., no need to update the accumulator each time when a new smart meter is added to AMI). Furthermore, since the number of revoked certificates is also less than the number of valid certificates which affects the required computation time significantly [6]. Our approach mitigates these drawbacks by addressing security and application-specific issues and offering to use CRLs instead of valid certificates.

3. Preliminaries

Before explaining our approach we provide some cryptographic background of accumulators and its particular form as RSA accumulators. In addition, to help the reader grasp a general idea of revocation management through CRLs, we explain the CRL and delta-CRL notions.

3.1. Background on cryptographic accumulators

Benaloh and De Mare [17] introduced the cryptographic accumulator concept which is a one-way hash function with a special property of being *quasi-commutative*. A quasi-commutative function is a special function $\mathcal{F}$ such that $y_0, y_1, y_2 \in \mathbb{Y}$:

$$\mathcal{F}(\mathcal{F}(y_0, y_1), y_2) = \mathcal{F}(\mathcal{F}(y_0, y_2), y_1)$$

(1)

The properties of this function can be summarized as follows: *(1)* it is a one-way function, i.e., hard to invert; *(2)* it is a hash function for obtaining a secure digest $\mathcal{A}$ (i.e., accumulator value) where $\mathcal{A} = \mathcal{F}(\mathcal{F}(\mathcal{F}(y_0, y_1), y_2), \dots, y_n)$ for a set of values $\{y_0, y_1, y_2, \dots, y_n\} \in \mathbb{Y}$; *(3)* it is a *quasi-commutative* hash

function which is different from other well-known hash functions such that the accumulator value $\mathcal{A}$ does not depend on the order of y_i accumulations.

These properties allow cryptographic accumulators to be used for a condensed representation of a set of elements. In addition, since the resulting accumulated hashes of y_i ($\mathbb{Y} = \{y_i; 0 < i < n\}$) stays the same even if the order of hashing is changed, it can be used for efficient membership testing by using a special value called witness value w_i. For instance, the witness w_i of corresponding y_i is calculated by accumulating all y_i except the case where $i \neq j$ (e.g., $w_i = \mathcal{F}(\mathcal{F}(\mathcal{F}(y_0, y_1), \dots, y_{j-1}, y_{j+1} \dots, y_n))$). Then, when necessary any of the members can check whether is also a member of the group by just verifying whether $\mathcal{F}(w_i, y_i) = \mathcal{A}$. Note that, because $\mathcal{F}$ is a one-way function, it would be computationally infeasible to obtain w_i from y_i and $\mathcal{A}$. However, there is a risk for collusion in this scheme when an adversary can come up with $w_i{}'$ and $y_i{}'$ pairs where $y_i{}' \notin \mathbb{Y}$ to obtain the same accumulator value: $\mathcal{F}(w_i{}', y_i{}') = \mathcal{A}$. In the literature, there is already a cyrptographic accumulator, namely the RSA construction [20] which guarantees that finding such pairs is computationally hard by restricting the inputs to the accumulator function to be prime numbers only. This scheme is known as collision-free accumulator that enables secure membership testing (i.e., without any collision). Therefore, in this paper, we chose to employ RSA construction which is elaborated next.

3.2. RSA accumulator

RSA accumulator [20] has a RSA modulus $\mathcal{N} = pq$, where p and q are strong primes. The RSA accumulation value $\mathcal{A}$ is calculated on consecutive modular exponentiation of prime numbers set $\mathbb{Y} = \{y_1, \dots, y_n\}$ and g is quadratic residue of $\mathcal{N}$ as follows:

$$\mathcal{A} = g^{y_1, \dots, y_n} (mod \mathcal{N})$$

(2)

The witness w_i of corresponding y_i is calculated by accumulating all values except y_i:

$$w_i = g^{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n} (mod \mathcal{N})$$

(3)

Then, the membership testing can be done via a simple exponential operation by comparing the result with the accumulator value $\mathcal{A}$:

$$w_i^{y_i} \leftrightarrow \mathcal{A}$$

(4)

The described accumulator scheme so far basically allows generation of a “witnesses” to prove that an item is in the set. A more advanced accumulator would offer proofs of non-membership which proves that an item is **NOT** in the set [21]. For this scheme, let us assume any $x \notin \mathbb{Y} = \{y_1, \dots, y_n\}$. In a nutshell, the non-witness values can be computed by the following steps: Let u denote $\prod_{i=1}^{n} y_i$, the scheme finds non-witness nw_1, b value pairs of x by solving the equation of $nw_1 \times u + b \times x = 1$using the Extended Euclidean algorithm. Then, the scheme computes an additional value nw_2 such that:

$$nw_2 = g^{-b} (mod \mathcal{N})$$

(5)

After these steps, the item x will have cryptographic proof values $nw_1 \& nw_2$ which can be used to ensure that the item x is **NOT** in the set $\mathbb{Y}$. Then, any third party that possess the $\mathcal{A}$ value can do the non-membership test c of via a simple exponential operation by checking whether the following equation holds:

$$\mathcal{A}^{nw_1} \leftrightarrow {nw_2}^x \times g(mod\mathcal{N})$$

(6)

Besides, if a new value y' is added to list, the accumulator value is updated by using the previous accumulator value $\mathcal{A}$:

$$\mathcal{A}' = \mathcal{A}^{y'}(mod\mathcal{N})$$

(7)

3.3. Certificate, CRL and delta CRLs

As we deal with certificates, we would like to also provide some basic background on certificates and their management. Certificates are issued by a CA with a planned lifetime to an expiration date and have unique serial numbers. Once issued, these certificates are valid until their expiration date. However, there are various reasons that cause a certificate to be revoked before the expiration date. These reasons include but not limited to compromise of the corresponding private key, changing the underlying device infrastructure, etc.

Revocation causes each CA regularly issued a signed list called a CRL which is a time-stamped list consisting of serial numbers of revoked certificates and revocation dates. When a PKI-enabled system uses a certificate (for example, for verifying the integrity of a message), that system should not only check the time validity of the certificate, but an additional check is required to determine a certificate's revocation status during the integrity check. To do so, CRL can be checked to determine the status of the certificate.

There are two main types of CRL: *full CRLs* and *delta CRLs*. A full CRL contains the status of all revoked certificates which are not expired yet. *Delta CRLs*, which is a concept defined in RFC 5280 [22], contain only the status of newly revoked certificates that have been revoked after the issuance of the last *full CRL* and before the new release of it. Therefore, a *full CRL* is issued for a limited time frame and should be updated regularly. Until next update time, *delta CRLs* help keeping track of the newly revoked certificates. When *delta CRLs* are enabled, the CA can distribute *full CRLs* at longer intervals (for reducing distribution overhead) and delta CRLs at shorter intervals. An important point about delta CRL concept is that it does not eliminate the requirement of full CRL distribution. The full CRL must still be re-distributed when the previous full CRL expires since CRL has also a lifetime period as certificates and the lifetime period of delta CRLs are dependent on the lifetime of the previous full CRL. This means both the *full CRL* and *delta CRL* should be updated regularly by all the potential nodes that will be using them. In the case of AMI, these CRLs may contain thousand of revoked certificate IDs due to longer expiration dates of issued certificates (even lifelong [4]) and need to be distributed to the each smart meters which will cause a huge overhead due to their size as will be shown in the experiments.

4. System and threat model

In this paper, we build a revocation management scheme for a typical AMI infrastructure. Basically, revocation information is collected by the utility company in the forms of CRL files. Each CRL file contains revoked certificates IDs issued by different CAs. Then, all these revocation information are disseminated to AMI through a 4G/LTE and AMI mesh communication infrastructure. A sample system model is shown in Fig. 1.

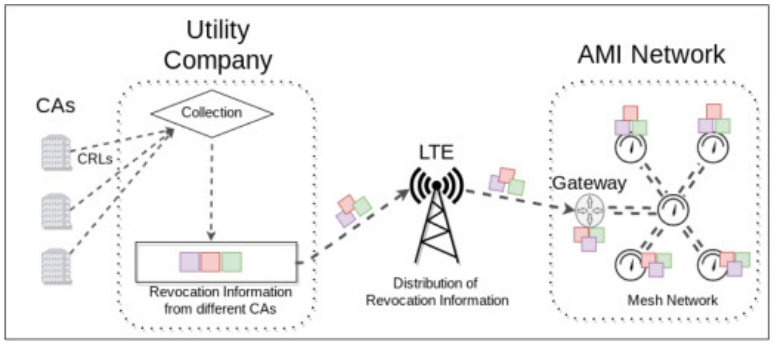


Fig. 1. System model.

The security of the proposed revocation management scheme depends on the secure implementation of the proposed accumulator-based system. Therefore, we consider the following threats to the security of the proposed approach and identified the relevant security goals. Note that in our attack model, we assume that both the accumulation process within the perimeter and smart meters (outside the perimeter) can be compromised. Besides, the communication between UC and smart meters is happening on a non-secure medium which means an adversary can eavesdrop the communication both actively and passively. This threat model is very strong and adequate to represent the increasing threats to Smart Grid. However, a single counter-measure against this threat model would not be sufficient when considering the broad and diverse attack surface of it. To ensure the security of AMI against adversaries, the utility company needs to deploy intrusion prevention systems and proper attack prevention tools as well. Thus, we assume that a PKI inspection system along with an intrusion detection system (IDS) is already deployed and provides device-level controls to protect PKI keys and informs UC in case of any infiltration.

1. **Compromising Smart Meters:** In an attacker's perspective, the meter/gateway is the entry point to the AMI. The attacker can use a compromised smart meter or impersonate the gateway to apply various attacks.
2. **Compromising the UC Servers:** Apparently, compromising the servers within perimeters of UC provides lots of attack opportunities to adversary. The adversary can target AMI by directly attacking revocation management through compromising servers that governs revocation operations.
3. **Compromising the Communication:** When UCs are deploying AMI systems, they generally opt-out enabling encryption since IEEE standards does not enforce the UCs to deploy encryption due to various reasons [23]. It makes AMI open to adversaries who can easily eavesdrop whole AMI traffic or a portion of it. This can also pose a threat to revocation management.

5. Proposed approach

5.1. Overview

The proposed approach basically eliminates the need to store and distribute CRLs when the devices communicate in a secure manner. Instead of keeping a CRL file for verification of revocation status of certificates, our approach dictates to store at each device (e.g., smart meter, gateway, HES, etc.) only an accumulator value and a proof which proves the validity of the device's certificate. The accumulator value and proof can be computed at the utility company and distributed to devices in advance. Any updates regarding revoked certificates trigger re-computation of these values. Keeping just two integer values for revocation management brings a lot of efficiency in terms of storage and distribution overhead as will be shown in the Experiments section. In the next subsections, we will explain the details of our approach.

5.2. Adaptation of RSA accumulator for our case

To apply the cryptographic accumulators for revocation management, the revocation management needs to be viewed holistically from the lens of systems thinking to ensure security. We took a bottom-up approach while adapting the accumulator scheme to our approach. First, we modified the CRL inputs to meet the requirements for constructing a secure accumulator setup. Second, we improved the performance of accumulator calculation. Third, the accumulation process was divided into different functions and their tasks were defined. Then, we introduced new entities to AMI and assigned tasks to them. Lastly, we constructed a revocation check protocol that utilizes the produced accumulator solution. This section covers how we accomplished all these steps in details.

a. **Integration of CRL and non-witness Concept:**
   In the traditional CRL approach, when a smart meter presents its certificate to the recipient meter, that meter needs to verify that the presented certificate is **NOT** in the CRL. To be able to employ the accumulator approach, we generate *non-witness values* for the presenter to prove that it is not in the list. We accumulate the revocation information (stored in CRLs) into a single accumulator value and produce non-membership witnesses for the non-revoked smart meters.
b. **Reducing the Complexity of Accumulator Computation:**
   While computing the accumulator value using Eq. (2), the exponent needs to be computed as $\prod_{i=1}^{n} y_i$ before doing the modular exponentiation. This becomes infeasible when the size of $\mathbb{Y}$ increases since $u = \prod_{i=1}^{n} y_i$ will be $n \times k$ bits assuming each y_i is a k-bit integer. In our approach, we decided to use Euler's Theorem [24] to cope with this complexity. With access to the totient of $\mathcal{N}$ (i.e., $\phi(\mathcal{N})$), the exponent of g in accumulation computation will be $u' = \prod_{i=1}^{n} y_i mod \phi(\mathcal{N})$. Thus, with the knowledge of the totient, it becomes more efficient to compute the required values via reducing the u by $\phi(\mathcal{N})$.
c. **Generating Prime Inputs for the Accumulator:**
   For accumulation, we can use the certificate IDs (c_{id}) which are generated by the CAs. However, to ensure a collision-free accumulator, we need to use only prime numbers as dictated by the RSA accumulator. Since CRLs contain arbitrary serial numbers for certificate IDs, it is necessary to compute a prime representative for each certificate ID as an input to the RSA accumulator. Thus, we used the random oracle based prime number generator described in [25] to obtain prime representatives of certificates from their serial numbers (c_{id}). The scheme basically has a

random oracle $\Omega()$ function which produces a random number t for an input c_{id}. We use $\Omega()$ to find a *256-bit* number, d, which causes the result of the following equation to be a prime number:

$$y = 2^{256} \times \Omega(c_{id}) + d$$

(8)

By solving this equation, we generate a prime representative y for a revoked certificate. The reader is referred to [25] for security proof details of the method.

d. Defining Functions of Revocation Management:

After preparing the inputs, we compiled and modified the offered accumulator structure and proposed the following functions to construct revocation management for AMI. Our RSA accumulator uses the following input sets: $\mathbb{Y}$ *is the set of prime representatives of revoked certificates' serial numbers* and $\mathbb{X}$ *is set of prime representative of valid certificates' serial numbers* where $x \in \mathbb{X}$:

- $aux_{info}, \mathcal{N} \leftarrow Setup(k)$: This function is to setup the parameters of the accumulator. It takes k as an input which represents the length of the RSA modulus in bits (e.g., 2048, 4096, etc.) and generates $\mathcal{N}$ modulus along with aux_{info} which is basically Euler's totient $\phi(\mathcal{N})$.
- $\mathcal{A} \leftarrow ComputeAcc(\mathbb{Y}, r_k, aux_{info})$: This is the actual function which accumulates revocation information by taking prime representatives of serial numbers set $\mathbb{Y}$. While computing the accumulator value, we propose to use an initial random secret prime number r_k as a first exponent (g^{r_k}) in Eq. (2).
- $nr_{proof} \leftarrow ComputeNonRevokedProof(aux_{info}, \mathbb{Y}, x)$: This function first computes a pair of non-witness values represented as (nw_1, nw_2) for a valid certificate whose prime representative is x. Then, the UC concatenates the non-witness value pair with x and the serial number of the certificate creating a 4-tuple called nr_{proof}.
- $0,1 \leftarrow RevocationCheck(\mathcal{A}, nr_{proof})$: When a smart meter which has a prime representative x wants to authenticate itself to another party, the other one uses nr_{proof} and $\mathcal{A}$to verify that x is *not* in the accumulated revocation list by checking Eq. (6).
- $\mathcal{A}^t \leftarrow UpdateAcc(\mathcal{A}^{t-1}, \mathbb{Y}^t)$: This function is for updating the accumulator value $\mathcal{A}$ when the revocation information is updated via deltaCRLs. It takes a set of prime representatives of corresponding newly revoked certificates $\mathbb{Y}^t$ and latest accumulator value $\mathcal{A}^{t-1}$, and returns the new accumulator value $\mathcal{A}^t$ by utilizing Eq. (7).
- ${nr_{proof}}^t \leftarrow UpdateNonRevokedProof(\mathcal{A}^t, \mathbb{Y}^t, x)$: This function is for updating the non-revoked proof of corresponding valid smart meters when the revocation information is updated via deltaCRLs. It takes a set of prime representatives of corresponding newly revoked certificates $\mathbb{Y}^t$, the updated accumulator value $\mathcal{A}^t$, and the prime representative x and returns non-revoked proof ${nr_{proof}}^t$ of smart meter after some additional certificates are revoked by utilizing the process for Eq. (5).

Next, we define the components of the proposed framework.

5.3. Components of revocation management system

We propose the system architecture shown in Fig. 2 to enable the proposed revocation management and to define its interaction with the deployed AMI components. In addition, the newly introduced components of this architecture and their roles in executing the above defined functions are described below:

- *Smart Meters and Gateway:* The smart meters and gateway can directly communicate with each other and with Head-end System (HES) over LTE. Thus, to ensure the security of applications, these devices need to run the $RevocationCheck()$ function and carry the latest $\mathcal{A}$and the corresponding nr_{proof}.
- *Head-End System:* HES is an interface between the utility operations center and smart meters, and it is located in a demilitarized zone (DMZ). The primary function of the HES is collecting the power data from smart meters and transfer them to head-end management servers (HMS). Since it has two-way communication with smart meters, it needs to run the $RevocationCheck()$ function and carry the latest $\mathcal{A}$ and its nr_{proof}.
- *CRL Collector:* The CRL collector plays one of the key roles in our revocation management system. It basically collects CRLs from various CAs and feeds them to the Accumulator Manager. Since it has an open interface to the outside network (communicating with other CAs), it is placed in DMZ area.
- *Accumulator Manager:* Accumulator Manager is the core of our revocation management scheme. It gets CRL information from the CRL Collector and accumulates them to obtain latest accumulator value. It implements the *Setup()*, *ComputeAcc()*, *ComputeNonRevokedProof()*, *UpdateAcc()*, and *UpdateNonRevokedProof()* functions. Whenever a new accumulator value is calculated at a time t, it sends the accumulator value $\mathcal{A}^t$ and updated ${nr_{proof}}^t$ to the HMS which then forwards them to HES for distributing to the smart meters.
- *Head End Management Server:* The collected data is managed within HMS. It basically monitors activity logs, identifies new devices and manages incident response processes. As mentioned, the HMS collects the newly generated $\mathcal{A}$ and nr_{proof} values and sends them to HES for distribution.

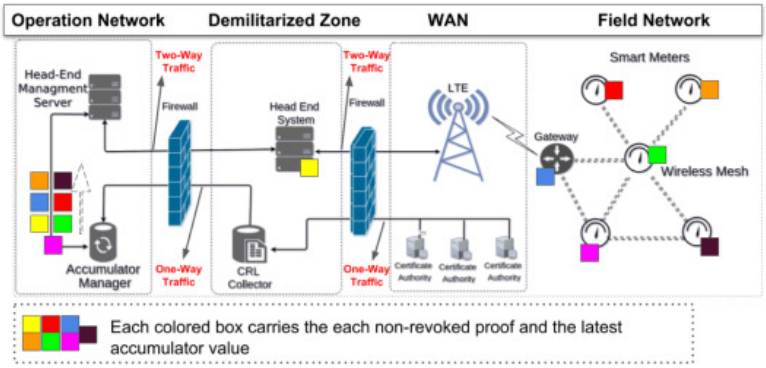


Fig. 2. The structure of proposed revocation management.

5.4. Revocation and certificate verification processes

In this section, we describe the proposed revocation scheme and the protocol for certificate verification.

5.4.1. Accumulating the CRL

This process includes two phases namely the setup phase and the update phase which are described below.

- **The setup phase:** In this phase of our approach, the Accumulator Manager in the UC basically accumulates the revoked certificate IDs in *full CRLs*. This process works as follows: The *full CRL* files are read, and each certificate ID and its issuer's public key are concatenated to obtain a unique string that will be input to the accumulator. Note that the issuer's public key is concatenated on purpose to eliminate any duplicates in serial numbers that may come from different CAs. Then, the Accumulator Manager calculates prime representatives for each concatenated string and accumulates these prime representatives to obtain the accumulator value. Finally, the Accumulator Manager generates non-revoked proofs (i.e., the 4-tuple nr_{proof}) for each end-device (smart meter, gateway, HES, etc.) by using $ComputeNonRevokedProof()$ function.
- **The update phase:** This phase is for revocation information updates that can be done through *delta CRLs*. Due to such updates, the accumulator value $\mathcal{A}$ and nr_{proof} values should be updated. To update these values, the Accumulator Manager first prepares the prime representatives for the newly revoked certificates (i.e., the ones that are included in the *delta CRLs*) by following the same approach in the setup phase. It then updates the previously computed accumulator value, $\mathcal{A}^{t-1}$, by using the $UpdateAcc()$ function to obtain $\mathcal{A}^{t}$ which is then used to generate new nr_{proof} tuples for the end devices by using the $UpdateNonRevokedProof()$ function.

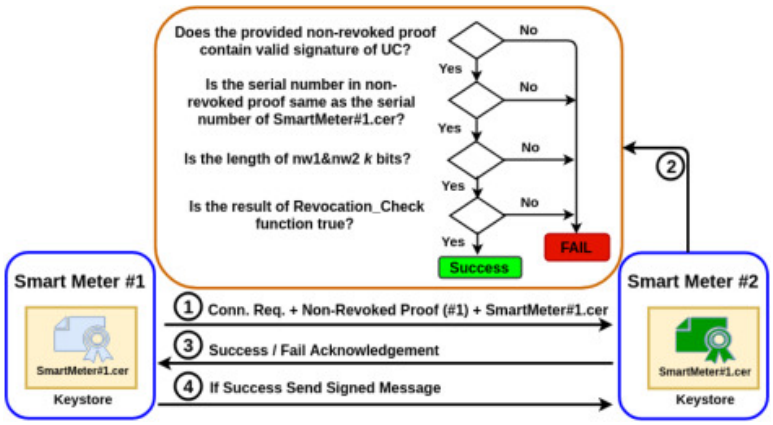


Fig. 3. Certificate verification protocol scheme.

5.4.2. Certificate verification protocol

When two meters communicate by sending/receiving signed messages, the signatures in these messages need to be verified. To be able to start the verification process, a receiving device needs to use the public key (for signature verification) presented in the certificate sent to itself. To ensure that

this certificate is not revoked, then it needs to initiate a process which we call as certificate verification protocol. Fig. 3 shows an overview of this process. Basically, the receiving device checks the corresponding nr_{proof} tuple's signature to ensure that it is produced by the UC. Once the signature is verified, it then checks whether the serial number within the tuple is same as the serial number of the provided certificate (i.e., either EndDevice#1.cer). For additional security, it also checks the length of the $nw_1 \& nw_2$ to see whether it is equal to the first accumulation setup parameter k. Next, it perfoms $RevocationCheck()$ function and checks whether the provided nr_{proof} is correct. Finally, the signature of connection request message is checked to ensure the integrity and authenticity of the request. If all these steps are successful, the end-device has successfully complete the certificate verification protocol. Note that, without carrying the nr_{proof}, a smart meter cannot be authenticated even if it has a valid certificate.

6. Evaluation of the approach and its objectives

The main objective of our work is to decrease the dissemination overhead of revocation information on AMI. However, although any reduction in this overhead is important for the general health of Smart Grid, achieving this goal without sacrificing the security is vital. Thus, we have determined the following general measures to evaluate the proposed approach in terms of security, communication, computation and storage.

- First, we will evaluate distributing process of non-revoked proofs to smart meters to assess the communication-related overhead of our approach.
- Second, since our approach requires computationalresources to calculate the non-revoked proofs and accumulator value, we will evaluate computational costs on UC servers.
- Considering the limited computation resource of a typical smart meter, it is essential to evaluate computational aspects on smart meters as well. Moreover, we will evaluate storage space requirement of our approach for smart meters.
- Finally, we will assess the security of the proposed approach against threats that are defined in Section 4.

We evaluate the communication, computation and storage overhead of our approach by using the following metrics:

- *Completion Time*: This metric is defined for communication overhead assessment, which indicates the total elapsed time to complete the distribution of accumulator value and non-revoked proofs to the smart meters from the HES. This metric hints on the communication overhead of revocation management in terms of assessing how it keeps the communication channels busy which are critical for carrying other information.
- *Computation Time*: This is the metric to measure the total time for completing the required computations such as computation of accumulator value, prime representatives, and revocation check time, etc.
- *Storage*: This metrics indicates the amount of space for storing the CRL information in the meters.

For comparison to our approach, we used two other baselines from the literature:

- *Traditional CRL Method*: Each smart meter keeps the whole CRL [13] locally which is distributed by the UC.
- *Bloom Filter Method*: A Bloom filter [11] is used to store revoked certificates information. Note that, we employed *murmur* hash function, which is a non-cryptographic hash function suitable for *fast* hash-based lookup, to build this Bloom filter. In this case, the Bloom Filter is distributed to each meter by the UC.

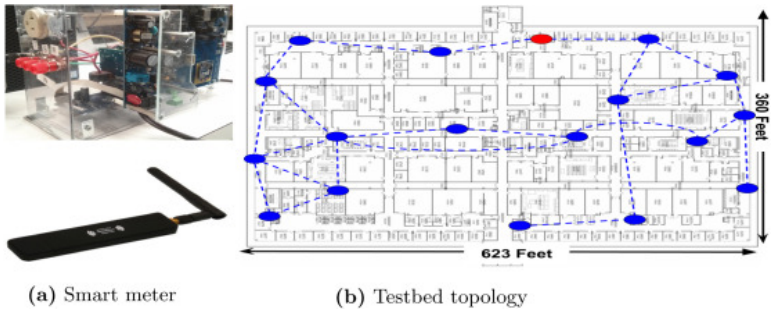
7. Performance evaluation

7.1. Experimental setup

To assess the performance of the proposed approach, we implemented it in C++ by using FLINT [26], which is the fastest library for number theory and modular arithmetic operations over large integers. For the RSA modulus generation and prime representatives computation, we used Crypto++ library since it allows thread-safe operations. We prepared a binary-encoded *full CRL* and *delta CRL* that have been digitally signed according to RFC 5280 standard and contained 30,000 and 1000 revoked certificates for *full CRL* and *delta CRL* respectively. The *full CRL* was used to compute $\mathcal{A}$ and nr_{proof} tuples during the setup phase while the *delta CRL* ws used for updating both $\mathcal{A}$ and nr_{proof} tuples.

For communication overhead assessment, we used the well-known ns-3 simulator [27] which has a built-in implementation of IEEE 802.11s mesh network standard. The underlying MAC protocol used was 802.11g. We created two different AMI grid topologies that consist of 81 and 196 smart meters. Even though the number of smart meters in our simulation setup is less than a real AMI setup, it still represents a practical setup in terms of the number of hops due to limited transmission range of 802.11g which leads to multiple hops to reach a smart meter from the gateway (e.g., for 81 nodes the average hop count is 6 and for 196 setup average hop count is 9). In a typical AMI setup in the wild, utilities are able to use 900MHz frequency bands [28] which helps to reach thousand of smart meters through a few hops due to the extended transmission range. Unfortunately, ns-3 does not support those frequencies to build a mesh network, and thus we created a simulation environment which reflects similar number of hops as in the wild.

Although ns-3 provides very good simulation environment in terms of signal propagation, it still lacks to reflect the effects of real conditions on the signal such as path attenuation, refraction and diffraction while it propagates in wild. To see the effects of such conditions, we also built an IEEE 802.11s-based mesh network comprised of 18 Protronix Wi-Fi dongles attached to Raspberry-PIs which are integrated with the in-house meters as shown in Fig. 4a. We carefully dispersed the meters on the floor as shown in Fig. 4b and build the shown multi-hop routing structure among meters by limiting transmission range by decreasing Tx-Power up to by a factor of 16 [29]. By such positioning and decreased Tx-Power, we strive to mimic realistic conditions on signal propagation and its effects on multi-hop routing in a real AMI setup.



(a) Smart meter (b) Testbed topology

Fig. 4. AMI testbed.

7.2. Communication overhead

7.2.1. Distribution overhead

In this subsection, we report on the completion time for the non-revoked proofs distribution of our approach with respect to other baselines both in simulation and testbed environments. The results which are shown in Fig. 5 indicate the accumulator approach significantly reduces the completion time compared to local CRL and bloom filter approaches due to condense accumulating. Even with respect to Bloom filter, which is touted as one of the most efficient methods in the literature, our approach reduced the completion time in approximately more than 10 orders of magnitude.

Another critical observation from the simulation results is the scalability capabilities of our approach. While especially for the local CRL approach, the completion time increases significantly, this is not the case for our approach. This can be attributed to the fact that the accumulator value is independent of the revoked CRL size while the overhead of other methods is proportional to the CRL size. The main overhead of our approach is directly related to the accumulator setting which was 2048 bits in our case. Therefore, even for very large-scale deployments that can have millions of meters, the overhead will not be impacted. In analyzing the experiments results for the testbed, we observe that the completion time takes more time even though the network size is much smaller. This is mainly because of the signal propagation issues such as path attenuation, refraction, interference from other devices, etc. within the building which does not exist in ns-3 simulations. Such issues cause more errors and packet loss and thus increase the re-transmissions to complete all packet distributions. In fact, the AMI infrastructure might have a similar challenge depending on the geographical location (e.g., urban vs rural environments) and thus the distribution of CRL will become even more critical. Therefore, our approach will be more suitable for such environments to reduce the impact from the wild.

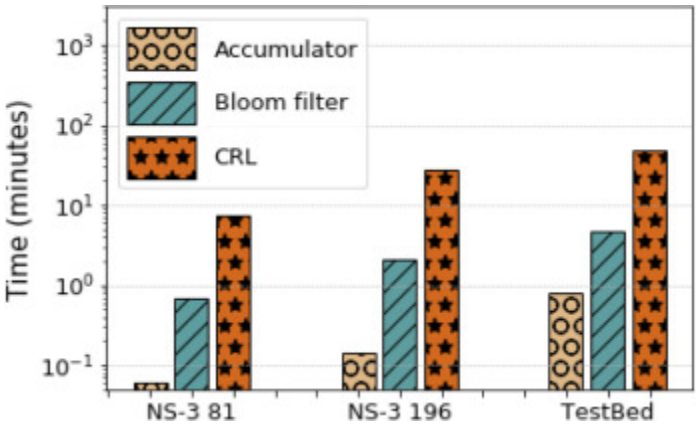


Fig. 5. CRL distribution overhead .

7.2.2. Update overhead

In this subsection, we conducted experiments to assess the overhead of CRL updates assuming that such updates are done regularly using the *delta CRL* concept. Fig. 6 shows revocation update overhead in terms of the completion time.

As in the case of full CRL, our approach significantly outperforms others due to of the size of the delta CRL. However, the results for the Bloom filter approach shows a different trend this time. It performs worse than the local CRL approach. This can be explained as follows: For each updated revocation information, the Bloom filters must be created from scratch to carry both previous and newly revoked certificates. As a result, updating the CRL will take slightly more time than the whole CRL distribution for Bloom filter and thus will take more time than the local CRL approach. Note that the overhead of CRL distribution is proportional to the size of the delta CRL and thus the completion time follows a similar trend with the results in Fig. 5.

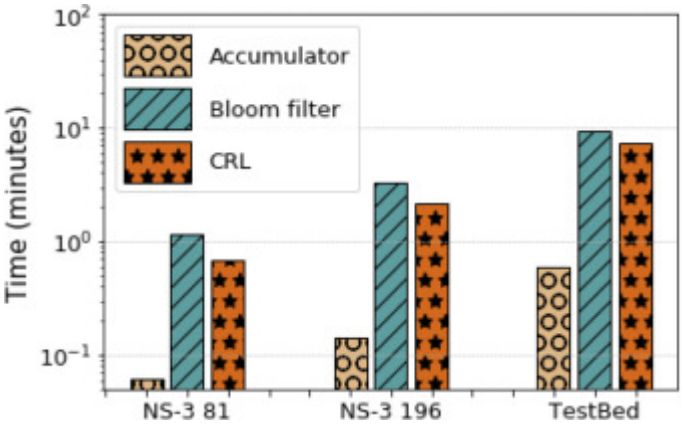


Fig. 6. CRL update overhead.

For the testbed results, we observe a similar which consistent with the simulations. Again, the completion time is more due to signal propagation and interference issues.

7.3. Computation overhead

We have demonstrated in the previous subsection that our approach significantly reduces the communication overhead. But, we need to also assess whether such a reduction introduces any major computational overhead. Thus, in this subsection, we investigated a detailed computational overhead of our approach. Specifically, we conducted two types of experiments: (1) We assessed the overhead of the computations due to the accumulation process in the Accumulator Manager. These experiments were conducted on a computer which has 64-bit 2.2 GHz CPU with 10 hardware cores, and 32 GB of RAM assuming that these are reasonable assumptions for the computer that will act as the Accumulator Manager. Moreover, we also investigated whether some of these computations can be parallelized to reduce the computation times through multi-thread implementations further; and (2) We assessed the computation time for the $RevocationCheck()$ function in meters by implementing it in a Raspberry Pi (smart meter).

7.3.1. Overhead results for the accumulator manager

In this subsection, we present and discuss the overhead at the Accumulator Manager by considering the functions below:

Computing Prime Representatives: To assess the computational overhead of prime representative generation, we computed prime representatives for different set sizes. Note that since both the valid and revoked certificates' serial numbers are used in our approach, the input size can become huge when AMI scales.

Therefore, we also conducted a benchmark test by using threads to show the parallelization ability of our approach. The results are shown in Fig. 7. As can be seen, the computational complexity of the prime representative generation is not overwhelming. 10^5 representatives can be computed nearly in 1 min even using a single core. Parallelization reduces the computational complexity by roughly 10 folds which allows computational times in the order of seconds.

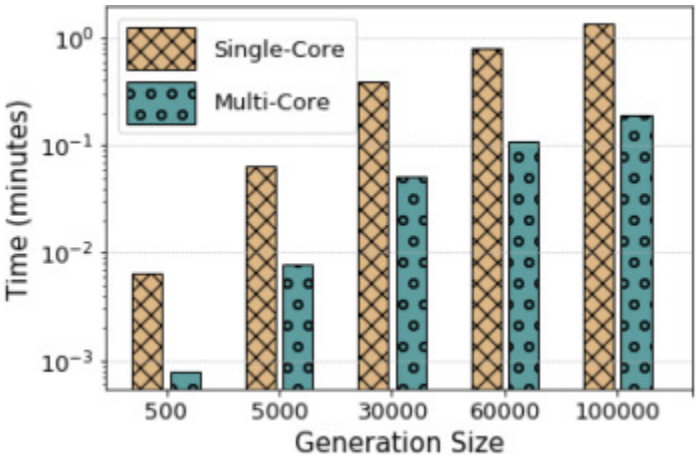


Fig. 7. Prime representative computation.

Computing the Accumulator Value: Next, we benchmark the computation cost of accumulator value according to different CRL sizes as used in the previous experiment. In addition, we also conducted tests to assess the computational difference between our setting (i.e., the Accumulator Manager has all aux_{info} information) and the case where the Accumulator Manager does not have aux_{info} as discussed in Section 5.

Note that for the computation of the accumulator value, a parallel implementation was not possible since each step in the computation depends on the previous operation. As seen in Fig. 8, the accumulator value is calculated under a minute for 10^5 revoked certificates even without using aux_{info}. However, the availability of aux_{info} significantly reduces the computation time making it possible to finish it milliseconds regardless of the size of the CRL.

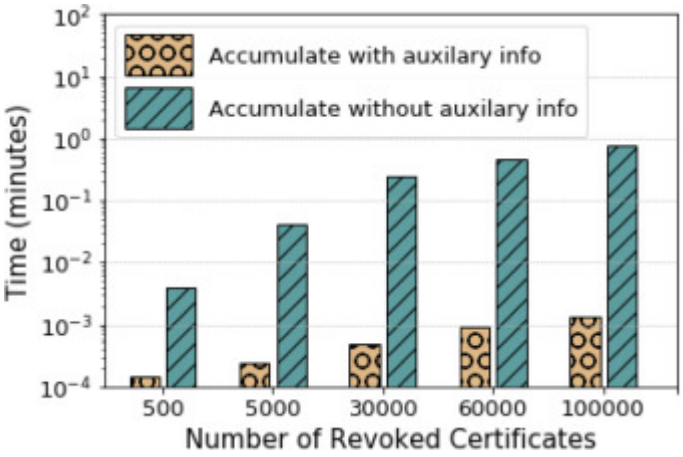


Fig. 8. Accumulator computation.

Computing Non-Revoked Values: Finally, we assessed the overhead of the computation of non-revoked proofs for both the first setup phase by using *full CRL* and the update phase by using *delta CRL*. Again, we conducted tests based on the availability/lack of aux_{info} and parallelization ability. Fig. 9 shows the computation overhead of this function according to different AMI sizes. As seen, aux_{info} makes a significant difference in this case.

Even with parallelization, the computational times are still in the order of days which may not be acceptable in an AMI setting. The results indicate that aux_{info} needs to be available for efficient computations. We repeated the same experiment for the $UpdateNonRevokedProof()$ function and observed the same trends since the only change was the size of the CRL (i.e., delta CRL is much smaller). These results were not shown due to space constraints.

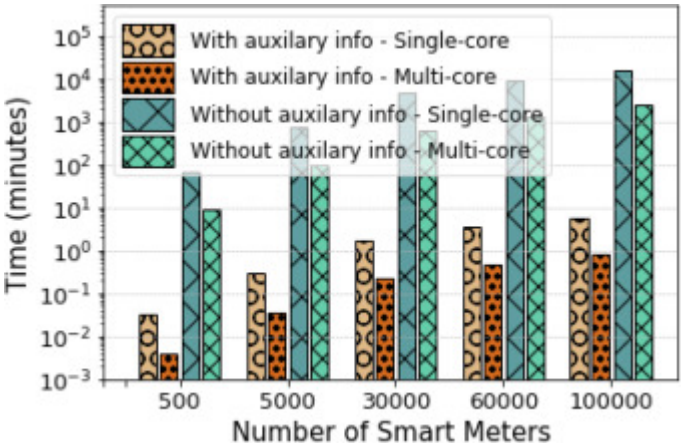


Fig. 9. nr_{proof} computation for *full CRL*.

7.3.2. Overhead results for smart meter

Revocation Check Overhead: We looked at the computational time of revocation check operations in smart meter based on the three approaches compared. This is an important experiment to understand the computation overhead of our approach on the smart meter, considering the fact that it has limited resources. As can be seen in Table 1, the elapsed time for a single revocation check is around 10 ms in our approach. Comparing with the other methods, the Bloom Filter has the best results as expected

because it enables faster checking by efficient hash operations. However, Bloom filter suffers from false-positives which degrades its efficiency by requiring access to the server [11]. Our approach does not have such a problem. While our approach doubles the revocation check time compared to the local CRL method, the time is still pretty fast as it is in the order of milliseconds which does not impact any other operation. This is a negligible overhead given that it brings a considerable space-saving benefit which affects both distribution and storage overhead.

Storage Overhead: To compare the storage requirements, we identified the needed revocation information size for our approach and compared it with the other approaches, as shown in Table 2. As expected, accumulator has a superior advantage since smart meters just need to store a small accumulator value and non-revoked proof value. Local CRL, on the other hand, keeps the whole CRL list and depending on the number of revoked certificates, it can be huge. For our scenario, the CRL size is around 0.7MB for 30K revoked certificates. While Bloom filter's performance is also promising, it is still not better than our approach and it suffers from false positives as discussed.

Table 1. Elapsed revocation check time.

	Local CRL	**Bloom Filter**	**Accumulator approach**
Average time (ms)	4.1	0.06	9.8

Table 2. CRL storage overhead.

	Local CRL	**Bloom Filter**	**Accumulator approach**
Required space (MB)	0.690	0.046	0.001

8. Security analysis

The security of the AMI depends on the secure implementation of our approach. Therefore, we considered the threats in Section 4 against the security of the proposed approach and identified the relevant security goals.

❶ **Compromised smart meter attack:** An adversary can accomplish this attack in two different ways. For the first one, the adversary compromises the smart meter, and then the smart meter may be used to perform various attacks to Smart Grid. The UC is responsible for detecting malicious activity by utilizing different tools and sources. After detection, the UC puts the serial number of smart meter's certificate to the accumulation process. The updated nr_{proof} and $\mathcal{A}$ values are distributed to the other smart meters, HES and gateways. This process basically detaches the compromised smart meter from the AMI. Every other non-compromised components which may interact with this smart meter are no longer be able to interact due to the our revocation check protocol. Once the information of the compromised smart meter is accumulated to obtain a new accumulator value, the attacker cannot successfully bypass the revocation check mechanism by using the compromised smart meter itself or its stolen private key and certificate in the future.

For the second one, the attacker can abuse a vulnerability in the certificate issuing process or steak smart meters' private keys from manufacturers. This time, the CA revokes certificates

of the corresponding devices and publish new CRLs. Those CRLs collected by our CRL collectors. These newly revoked certificates are then accumulated, and the corresponding non-revocation proofs are disseminated to AMI. With the updated accumulator values, an adversary cannot interact with any of the smart meters, HES, or gateways within AMI by using the revoked certificates.

❷ **Compromising the UC Servers:** In the event of an attack, the adversary's first target will be to compromise the accumulator manager to attack our revocation management. In our scheme, the accumulator manager plays a critical role but can be missed out easily because it is located within UC perimeters. However, the accumulator manager is the Achilles' heel of our approach and should be protected thoroughly. Thus, we investigate three possible attack scenarios and corresponding countermeasures within our revocation system.

(a) First, through the architectural design in Fig. 2 *accumulator manager* is protected from any attacks by not allowing direct communication from outside of the network through two different firewalls. For instance, the second firewall configuration just allows incoming traffic, which is directly started by the Accumulator Manager to collect CRLs. Thus, it is not easy to access to accumulator manager from outside of the perimeter.

(b) Considering the ever-increasing threat environment and improved skills of adversaries, no matter what level of protection our system has, the accumulator manager can get compromised by breaking a path the proposed defense layers. In such a case, the key factor will be how quickly our approach responds to the incident. After detection of the compromise (e.g., attacker steals RSA setting parameters such as aux_info and $p\&q$), the accumulator manager can be migrated to another server, and new nr_{proof} and $\mathcal{A}$ is computed from scratch by using different RSA primes p' and q' within minutes as shown in Section 7.3.1. Then, updated proofs are distributed to smart meters to prevent any further damage. So, our approach offers a pretty easy recovery capability which is important considering critical operations in AMI.

(c) Third, our scheme is also allowing computation of nr_{proof} without keeping critical security parameters of RSA accumulator settings RSA (i.e., aux_{info} and $p\&q$), since stolen aux_{info} and $p\&q$ values enable a malicious actor to prove any arbitrary statements. These parameters can be deleted once they are used in the setup phase. In such a case, compromising the accumulator manager does not give any advantage to the adversary to attack revocation management by abusing these parameters. Moreover, the computation of nr_{proof} can still be accomplished for new smart meters or/and in case of updated revocation information, but it is more computationally intensive as shown in the Experiments Section.

❸ **Compromising the Communication:** As stated before, the traffic of AMI is generally unencrypted and causes additional attack surface to our approach. In this subsection, we investigate two possible attack scenarios and countermeasures against them as follows:

(a) **Accumulator freshness attack:** An eavesdropping attack of AMI traffic poses a unique threat to our approach by combining public revocation information and circulating accumulator values. An attacker may perform a targeted attack if the UC has not

updated the accumulator value properly by pinpointing the smart meters that use old accumulator values. However, our approach is robust to this attack since while computing the accumulator value, we use a secret prime number r_k as a first exponent (g^{r_k}) in Eq. (2). This prevents inferring the freshness of accumulator value by combining publicly known revocation information and circulated values in unencrypted traffic.

(b) **Stolen non-witness attack:** One possible attack can be performed by using $nw_{1\&2}$ values to masquerade a valid smart meter since it can be obtained easily by eavesdropping. Our protocol is protected from this threat, even if the corresponding non-witness values $nw_{1\&2}$ of a smart meter are tried to be abused by attacker, the authentication during revocation check will still fail due to the multi-level signature checks. Thus, we relieve the attacks by abusing of stolen $nw_{1\&2}$ values through a multi-level authentication which combines the signature check with the accumulator check.

9. Benefits and limitations

There are several benefits associated with the use of the proposed approach for revocation management in AMI. Also, we highlighted some challenges and limitations of the approach.

9.1. Benefits

1. *Low overhead*: Our approach imposes minimal to no overhead to the smart meters deployed in the AMI and very low overhead to the central servers supporting the revocation management. In general, a revocation check contains at most one additional modular arithmetic operations if compared with the other revocation check methods (considering investigated methods realizes at least one modular arithmetic operation for signature check). Also, the overhead imposed by disseminating the revocation information to the smart meters is very low.

2. *Applicability and Security*: The steps used in our approach can be easily implemented to the current AMI infrastructure with few adjustments. We showed that which components of the current AMI setup will be affected and need to be updated with new functionality. To compare the applicability of our work with its alternatives, we determined four key benefits in total as shown in Table 3. Our approach collects the revoked certificates information without interrupting the current smart grid operational network setup. However, unlike OCSP or OCSP-stapled methods, it requires extra communication overhead to distribute revocation information. Still, AMI communication infrastructure is not natural to an off-the-shelf OCSP-based solution due to the frequent query requirement, so obviously, it does not carry any advantage for decreasing the communication overhead. On the other hand, our solution outperforms all other methods in terms of introduced distribution overhead. In brief, our conclusion from this comparative evaluation shows that our approach offers the same security benefits as other notable methods while keeping the overhead at the minimum level.

3. *A General Revocation Framework for Smart Grid*: Smart Grid is equipped with a myriad of various smart devices and sensors. This represents a new domain for security that is far beyond the traditional air-gapped operational network technology (OT) needs because of investments in distribution technologies such as renewable energy sources like rooftop solars and wind turbines. In

this context, our approach emerges as the first comprehensive solution that adapts the cryptographic accumulators to instrument lightweight revocation management and can be applied different domains in smart grid beyond AMI.

Table 3. High-level comparison of revocation management schemes.

	Applicability	**Storage Overhead Advantage**	**Communication Overhead Advantage**	**Security**
OCSP	○	●	○	●
OCSP-staple	○	●	○	●
CRL	◐	○	○	●
Delta CRL	◐	●	◐	●
Bloom Filter	◐	●	◐	●
Our approach	●	●	●	●

● = offers the benefit; ◐ = almost offers the benefit; ○ = does not offer the benefit.

9.2. Limitations

1. *Tight Synchronization Requirement*: Cryptographic accumulators are powerful tools for short set representation and secure non-membership proofs. However, a disadvantage of using an accumulator-based revocation scheme is that the non-revoked proof and accumulator value has to be synchronized between smart meters. This might occur in two ways. First, the accumulator value at the verifier's site is out-of-date, but the non-revoked proof of the prover is updated and vice versa. Asynchronous non-revoked proofs and accumulators between communicating smart meters may hinder the authentication operations; thus, AMI should ensure that all smart meters are updated and start to use the new proofs at the same time. Although the requirement for strict synchronization seems prohibitive, the AMI is a well-managed and synchronized network. Because of this characteristic of AMI, the synchronization requirement can be met easily.

2. *Not-allow to use Unreliable Distribution Methods*: Another limitation related to synchronization requirement is that an attacker may selectively drop the packets to cause a synchronization problem between smart meters. Thus, any unreliable method for the distribution of non-revoked proofs should be avoided and the UC should ensure that required values are completely reached to smart meters.

10. Conclusion

Considering the overhead of certificate and CRL management in AMI networks, in this paper, we proposed a one-way cryptographic accumulator based approach for maintaining and distributing the revocation information. The framework condenses the CRLs into a short accumulator value and builds a *secure*, efficient and lightweight revocation mechanism in terms of communication overhead. The approach is inspired by cryptographic accumulators and adopted based on the requirements of AMI. The experiment results indicate that the proposed approach can reduce the distribution completion time significantly for compared to CRL and Bloom filter approaches while introducing only minor additional computational overhead which is handled by the UC. There is no overhead imposed to smart meters.

As future works, we first aim to incorporate an improved accumulator scheme to relax the tight synchronization requirement. Since different smart meters may use different proofs and accumulator values, the proposed method may require some architectural modifications to enable relaxed revocation check but still ensures security.

Second, instead of using our in-house testbed, we will consider to use a real testbed such as EPIC [30] to be able to test our approach on a realistic AMI infrastructure with integrated power grid components.

CRediT authorship contribution statement

Mumin Cebe: Conceptualization, Methodology, Software, Writing - original draft.

Kemal Akkaya: Supervision, Writing - review & editing.

Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000779.

Declaration of competing interest

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

References

[1] H. Farhangi, The path of the smart grid, IEEE Power Energy Mag. 8 (1) (2010).
[2] N. Saputro, K. Akkaya, S. Uludag, A survey of routing protocols for smart grid communications, Comput. Netw. 56 (11) (2012) 2742–2771, http://dx.doi.org/10.1016/j.comnet.2012.03.027.
[3] NIST, Guidelines for smart grid cybersecurity, 2014, NISTIR 7628 Rev. 1. [4] landisgyr, Grid stream solutions overview, 2015, https://www.landisgyr.com/resources/gridstream-solution-security-2/.
[5] M. Nemec, M. Sys, P. Svenda, D. Klinec, V. Matyas, The return of coppersmith's attack: Practical factorization of widely used RSA moduli, in: Conference on Computer and Communications Security, ACM, 2017.
[6] Z. Durumeric, J. Kasten, D. Adrian, J.A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, et al., The matter of heartbleed, in: Proceedings of the 2014 Conference on Internet Measurement Conference, ACM, 2014, pp. 475–488.
[7] M.M. Mahmoud, J. Mišić, K. Akkaya, X. Shen, Investigating public-key certificate revocation in smart grid, IEEE Internet Things J. 2 (6) (2015) 490–503.
[8] S. Galperin, S. Santesson, M. Myers, A. Malpani, C. Adams, X. 509 internet public key infrastructure online certificate status protocol-OCSP, 2013.
[9] Y. Pettersen, The transport layer security (TLS) multiple certificate status request extension, 2013.
[10] J. Camenisch, A. Lysyanskaya, Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials, Springer, 2002, pp. 61–76.
[11] K. Akkaya, K. Rabieh, M. Mahmoud, S. Tonyali, Efficient generation and distribution of crls for ieee 802.11 s-based smart grid ami networks, in: Smart Grid Communications, IEEE, 2014.
[12] A.R. Metke, R.L. Ekl, Security technology for smart grid networks, IEEE Trans. Smart Grid (2010).

[13] M.M. Mahmoud, J. Misic, X. Shen, Efficient public-key certificate revocation schemes for smart grid, in: Global Communications Conference (GLOBECOM), 2013 IEEE, IEEE, 2013, pp. 778–783.
[14] K. Rabieh, M. Mahmoud, S. Tonyali, et al., Scalable certificate revocation schemes for smart grid ami networks using bloom filters, IEEE Trans. Dependable Secure Comput. (2015).
[15] M. Cebe, K. Akkaya, Efficient management of certificate revocation lists in smart grid advanced metering infrastructure, in: International Conference on Mobile Ad Hoc and Sensor Systems, 2017.
[16] C. Mumin, A. Kemal, Efficient publickey revocation management for secure smart meter communications using one-way cryptographic accumulators, in: ICC, 2018.
[17] J. Benaloh, M. De Mare, One-way accumulators: A decentralized alternative to digital signatures, in: Workshop on the Theory and Application of of Cryptographic Techniques, Springer, 1993.
[18] L. Reyzin, S. Yakoubov, Efficient asynchronous accumulators for distributed PKI, in: International Conference on Security and Cryptography for Networks, Springer, 2016, pp. 292–309.
[19] F. Baldimtsi, J. Camenisch, M. Dubovitskaya, A. Lysyanskaya, L. Reyzin, K. Samelin, S. Yakoubov, Accumulators with applications to anonymity preserving revocation., IACR Cryptol. ePrint Arch. 2017 (2017) 43.
[20] N. Barić, B. Pfitzmann, Collision-free accumulators and fail-stop signature schemes without trees, in: Advances in Cryptology—EUROCRYPT'97, Springer, 1997, pp. 480–494.
[21] J. Li, N. Li, R. Xue, Universal accumulators with efficient nonmembership proofs, in: ACNS, Springer, 2007.
[22] D. Cooper, Internet x. 509 public key infrastructure certificate and certificate revocation list (CRL) profile, 2008.
[23] IEEE, IEEE Standard for electric power systems communications-distributed network protocol (dnp3), 2012, https://standards.ieee.org/standard/1815-2012.html.
[24] K.H. Rosen, B. Goddard, K. O'Bryant, Elementary Number Theory and Its Applications, Pearson/Addison Wesley, 2005.
[25] C. Papamanthou, R. Tamassia, N. Triandopoulos, Authenticated hash tables, in: Proceedings of the 15th ACM Conference on Computer and Communications Security, ACM, 2008, pp. 437–448.
[26] W. Hart, F. Johansson, S. Pancratz, FLINT–Fast Library for Number Theory, Citeseer, 2011.
[27] ns-3, Ns-3: network simulator 3, 2016, http://www.nsnam.org/, Release 3.24.1.
[28] TinyMesh, Radio Frequency (RF) Datasheet, https://radiocrafts.com/uploads/RCxxxxHP-TM_Data_Sheet.pdf.
[29] J. Tourrilhes, Wireless extensions for linux, 1997, Linux.
[30] S. Adepu, N.K. Kandasamy, A. Mathur, Epic: An electric power testbed for research and training in cyber physical systems security, in: Computer Security, Springer, 2018, pp. 37–52.