Secure Data Collection in Constrained Tree-Based Smart Grid Environments

Haiming Jin[1], Suleyman Uludag[2], King-Shan Lui[3], and Klara Nahrstedt[1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
[2]Department of Computer Science, University of Michigan - Flint, MI, USA
[3]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong

***Abstract*—To facilitate more efficient control, massive amounts of sensors or measurement devices will be deployed in the Smart Grid. Data collection then becomes non-trivial. In this paper, we study the scenario where a data collector is responsible for collecting data from multiple measurement devices, but only some of them can communicate with the data collector directly. Others have to rely on other devices to relay the data. We first develop a communication protocol so that the data reported by each device is protected again honest-but-curious data collector and devices. To reduce the time to collect data from all devices within a certain security level, we formulate our approach as an integer linear programming problem. As the problem is NP-hard, obtaining the optimal solution in a large network is not very feasible. We thus develop an approximation algorithm to solve the problem. We test the performance of our algorithm using real topologies. The results show that our algorithm successfully identifies good solutions within reasonable amount of time.**

I. Introduction

One of the significant hallmarks of the Smart Grid (SG) initiative is the pervasive data sensing to facilitate a more efficient control. While the existing Supervisory Control And Data Acquisition (SCADA) system already collects data from various sensors, the scale and scope of the data collection in the SG are expected to pose new challenges. Applications of data sensing in the SG include conditional and structural monitoring of Distributed Energy Resources and renewables in the generation; State-of-Charge monitoring; substation, transformer, underground and overhead lines in the transmission and distribution; and collection of information from smart meters in the Advanced Metering Infrastructure (AMI) [1].

In addition to the emerging and new sensing in the SG, data collection from the legacy telemetric devices widely deployed in the field needs to be accommodated as part of the infrastructure, at the very least, in the transitional period. An example of such legacy telemetric collection need is provided in [2]. In this scenario, a mobile data collector (DC) moves along a road, or a certain path, to collect data from measurement devices (MDs). It can only connect directly to the MDs that are within DC's proximity. We call the MDs that can talk to the DC directly *root MDs*. Those MDs that are outside the communication range of the DC have to send data to the DC through root MDs in a multihop manner. Figure 1 presents a simple example where only root MDs (MD1, MD2, MD3) can talk to the DC directly. Other remote MDs should send their data to one of the root MDs. Note that the underlying data collection paradigm of the aforementioned case is present in many other emerging sensing and measurement scenarios [1], [3], especially when sequential data collection is needed over a multihop communications topology. To reduce the number of messages needed for the whole data collection, we propose generating several disjoint trees, each rooted at a certain MD, such that the trees reach all MDs. Fig. 2 shows an example, where bold edges show the trees formed from the roots.

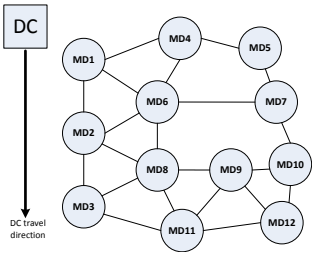


Fig. 1. An Example SG Network.

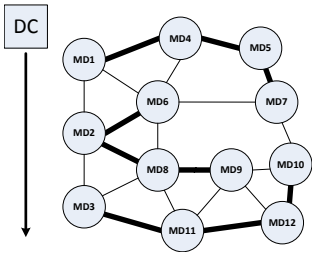


Fig. 2. Disjoint Trees.

In this paper, we are interested in secure, scalable, and optimized data collection in the SG. Our objective is to reduce the total data collection time while ensuring the confidentiality of the data. We encrypt data in a way that while allowing the relaying MDs to verify the message integrity, they cannot read the content under the *honest-but-curious* model. The data collection time is reduced through identifying data collection trees that have minimum sum of depths. We develop an optimization problem for the secure data collection, which is NP-Hard, and then develop an approximation algorithm.

The rest is organized as follows: Sec. II presents the related work. Tree-based secure data collection is detailed in Sec. III. Sec. IV develops the optimization problem and solution together with an approximation algorithm. Simulation results are given in Sec. V. Sec. VI concludes the paper.

II. Related Work

To the best of our knowledge, confidential data collection with integrity validation by means of forming disjoint trees to minimize the total collection time has not been studied in the literature. We dissect our problem into three constituents in order to present a discussion of the partially related work: (1) Multi-sink data collection in Wireless Sensor Networks (WSNs) and opportunistic networks [4], (2) Tree or cluster formation, and (3) Secure data collection.

There is some similarity in our formulation to data collection via mobile relay nodes [5], except that the objective of our approach is not energy efficiency Data collection by means of mobile sinks [6] is somewhat related in the sense that we are looking to choose the root of trees to relay the data to. The optimization part of our problem is similar to the Base

Station Problem from [7] where the problem of positioning data collecting nodes in a WSN is studied as a maximum flow problem with the objective of finding the optimal data rate. The NP-completeness proof of the problem is also given in [7]. While retaining the complexity[1], our problem is about the minimization of total time of the data collection and incorporates the security aspect. The tree formation of our approach connotes clustering algorithms, especially in WSNs [9]. However, unlike the goals of WSN clustering on node reachability and network longevity, we focus on security and data collection time minimization.

Finally, as for the security of the data collection task, there are two major approaches: One is to ensure the protection of the data content directly without regard to the data semantics. An approach presented in [10] is based on symmetric cryptography to provide data confidentiality and authentication between sensors and the base station. [11] describes a protocol for DC to collect data from an MD, but direct communication between DC and MD is assumed. Another category for providing security exploits the aggregate statistics of the sensed data, such as summation, average, minimum, maximum, etc. These approaches take advantage of in-network data processing (also referred to as *aggregation*) to induce some obfuscating operations on the transmitted data [12]–[20]. Our problem formulation does not assume any statistical property for in-network processing.

III. Secure Data Collection Via Trees

A. Overview

A Power Operator (PO) delegates a DC to collect data from a certain number of MDs. We assume each entity in the system possesses a pair of public and private key as long-term secrets. We denote the public key and private key of node A as A^+ and A^-, respectively. Before an MD is installed in the field, it is configured with its own public/private key pair and the public key of PO, but not the public key of DC. Our architecture does not require an MD to know the public keys of its neighbors, either. Apart from the keys, all entities are also configured with Diffie-Hellman (DH) parameters g and p for shared key establishment[2]. PO, DC, and MDs all agree to use the following basic cryptographic functions.

1) $PKE(K,M)$: Public key encryption on message M using key K
2) $SKE(K,M)$: Symmetric key encryption on M using K
3) $SIG(A,M)$: Signature of M by A (created using A^-)
4) $HASH(K,M)$: Compute the keyed-hash of M using key K

The security objective of the data collection is to protect the data reported by each MD such that only the PO can read the data generated by MDs. That is, even though the data reported has to be relayed by other MDs and the DC, these MDs and DC should not be able to read it. To achieve this efficiently, data should be encrypted by a shared key between a certain MD and the PO using symmetric key cryptography. We adopt the DH mechanism to develop the shared keys. On the other hand, to allow intermediate MDs to perform integrity checks, all the MDs within the same disjoint tree share a *Group Key GK* (to be explained below) with PO and DC.

We now briefly describe the whole procedure of data collection. First, the PO determines the disjoint trees to be used for data collection. Then, it provides the tree information and the necessary key information to the DC. The DC then talks to each root MD along its path to collect data. Root MD sends the key information along each branch on its tree to collect data. After it receives data from all MDs in its tree, it sends the data to DC.

B. Data Collection on a Branch

We first describe a simple situation that data is collected along a certain branch on the tree. We then describe the data collection of a whole tree. Fig. 3 presents how DC collects data on a tree branch spanning from MD_1, the root MD, to MD_2,

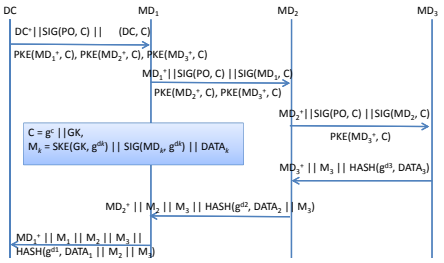


Fig. 3. Data Collection on a Tree Branch.

and then MD_3. Let the branch be $MD_1 \rightarrow MD_2 \rightarrow \ldots MD_l$, where MD_1 is the root MD. The public DH key of PO is g^c ($mod\ p$ is dropped for brevity), and the group key is GK. The key information created by the PO is denoted by $C = g^c||GK$, where $||$ represents the concatenation operation. Each MD on the branch must receive C to develop a shared key with the PO. The detailed procedure of data collection is as follows:

1) $DC \rightarrow MD_1$: $DC^+||SIG(PO,C)||SIG(DC,C)||$ $[PKE(MD_k^+,C), 1 \leq k \leq l]$
To protect C from eavesdroppers, C is encrypted using the public keys of the MDs. As only the PO knows the public keys of all the MDs, $PKE(MD_k^+,C)$ for $1 \leq k \leq l$ are created by PO and sent to DC, which will be discussed in Section III-C. MD_1 authenticates C by verifying the signature $SIG(PO,C)$. MD_1 verifies the legitimacy of DC by verifying $SIG(DC,C)$. MD_1 then signs C using its public key and sends the signature together with other information to the next MD along the branch.
2) $MD_1 \rightarrow MD_2$: $MD_1^+||SIG(PO,C)||SIG(MD_1,C)||$ $[PKE(MD_k^+,C), 2 \leq k \leq l]$
Note that as MD_1 also sends MD_1^+ to MD_2 so that MD_2 does not need to know any public key of any MD before the communication. When MD_2 receives the message, it performs similar operations as MD_1 that it authenticates C and MD_1. It

[1]Complexity of the optimization part of our problem may also be obtained from the classical *multi-facility location problem* [8].

[2]p is a prime number, and g is a primitive root $mod\ p$. Let A pick a secret a and B pick a secret b. A sends $g^a mod\ p$ to B, and B sends $g^b mod\ p$ to A. A can then compute the shared key by $(g^b mod\ p)^a mod\ p$. B computes the key by $(g^a mod\ p)^b mod\ p$.

then passes the key information to the next MD (more details will be provided in Section III-C).

3) $MD_i \rightarrow MD_{i+1}$: $MD_i^+||SIG(PO,C)||SIG(MD_{i+1},C)||$ $[PKE(MD_k^+,C), i+1 \leq k \leq l]$
Intermediate MDs on a tree branch keep forwarding the key information to the next MD after retrieving GK and g^c from C.

4) $MD_{l-1} \rightarrow MD_l$: $MD_{l-1}^+||SIG(PO,C)||SIG(MD_{l-1},C)||$ $PKE(MD_l^+,C)$
MD_l is the last MD on the branch. After verifying C, it prepares the reported data. It first generates its own DH half key g^{d_l} and develops the shared key between itself and the PO, which is g^{cd_l}. Let $DATA_l$ be the data encrypted and integrity protected using g^{cd_l}. MD_l then sends $DATA_l$ to MD_{l-1} for relaying to DC and finally to PO. To allow MD_{l-1} to perform authentication check on $DATA_l$, MD_l computes the hash of $DATA_l$ using g^{d_l}. MD_l should also let MD_{l-1} and the PO knows what g^{d_l} is. Let message $M_k = SKE(GK, g^{d_k})||SIG(MD_k, g^{d_k})||DATA_k$ for $1 \leq k \leq l$. M_l contains the DH half key and the encrypted data to be delivered to the PO.

5) $MD_l \rightarrow MD_{l-1}$: $MD_l^+||M_l||HASH(g^{d_l}, DATA_l)$
MD_{l-1} first retrieves g^{d_l} from $SKE(GK, g^{d_l})$ contained in M_l and verifies $SIG(MD_l, g^{d_l})$. It then can authenticate the message by verifying the hash. Note that although MD_{l-1} knows g^c and g^{d_l}, according to the property of DH protocol, MD_{l-1} still cannot compute g^{cd_l} to decrypt $DATA_l$. Thus, $DATA_l$ remains secret to MD_k for all $k = 1, \ldots, l-1$ and the DC. After verifying the message, MD_{l-1} prepares its own encrypted data $DATA_{l-1}$ and sends both $DATA_l$ and $DATA_{l-1}$ to MD_{l-2} together with the required key information.

6) $MD_{l-1} \rightarrow MD_{l-2}$:
$MD_{l-1}^+||M_{l-1}||M_l||HASH(g^{d_{l-1}}, DATA_{l-1}||M_l)$
MD_{l-2} processes the message in a similar manner as MD_{l-1} does. Finally, MD_1 would receive a message from MD_2 that contains all the data from MD_2 to MD_l. It can then prepare a message that contains all the data on the branch for the DC.

7) $MD_1 \rightarrow DC$:
$MD_1^+||M_1||M_2||\ldots||M_l||HASH(g^{d_1}, DATA_1||M_2||\ldots||M_l)$

C. Tree Structure Representation

To facilitate a root MD to collect data for its tree, it has to know which neighbors belong to its tree. The root MD also needs to inform its children of *their* children. In other words, the tree structure has to be embedded in the message from the DC to the root MD, and passed along to the MDs on the tree. $C = g^c||GK$ has to be encrypted using the public key of each MD on the tree. We use $PKE(MD_i^+, C)$ for all MD_i on the tree to represent the tree. Let $E(i) = PKE(MD_i, C)$, and let $T(i)$ be the tree representation rooted at MD_i. We further let MD_{ch_1}, ..., MD_{ch_m} be the children of MD_i if it is not a leaf. The representation is as follows:

$$T(i) = \begin{cases} [E(i), T(ch_1), \ldots, T(ch_m)] & \text{if } MD_i \text{ is not a leaf} \\ E(i) & \text{if } MD_i \text{ is a leaf} \end{cases}$$

Refer to the tree rooted at MD_2 in Fig. 2, $T(2) = [E(2), E(6), [E(8), E(9)]]$.

When MD_i receives $T(i)$, it can retrieve C from $E(i)$. It can also identify its children to forward key information. Let MD_j be a child of MD_i, it sends $MD_i^+||SIG(PO,C)||SIG(MD_i,C)||T(j)$ to MD_j.

An MD should wait for all its children to report data before sending its data to its parent. Data reporting does not have to follow the tree structure. A parent MD can simply append all M_k of descendant MD_k together. For example, MD_2 in Fig. 2 can send DC $MD_2^+||M_2||M_6||M_8||M_9||HASH(g^{d_2}, DATA_2||M_6||M_8||M_9)$.

D. Completing the Protocol

We now complete the protocol by describing the communication between the PO and the DC. Fig. 4 illustrates the information exchange. They first use the DH protocol to establish

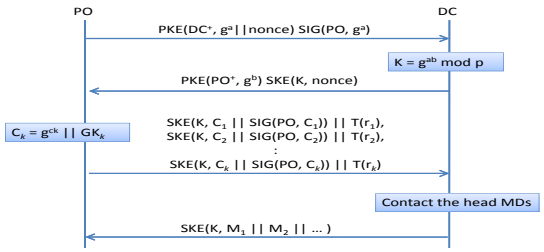


Fig. 4. Communication between PO and DC.

a shared secret K to secure the subsequent conversation. PO picks the group keys to be used. The tree structure and the key information are then encrypted using K. Suppose there are k root MDs, $MD_{r_1}, MD_{r_2}, \ldots, MD_{r_k}$. PO can use different DH half keys and group keys to develop shared keys with the MDs on different trees. Let the DH half key and the group key for the tree rooted at MD_{h_i} be g^{c_i} and GK_i, respectively. We further let $C_i = g^{c_i}||GK_i$. For root MD MD_{h_i}, PO sends $SKE(K, C_i||SIG(PO, C_i))||T(h_i)$ to DC. DC can then verify the signature for each C_i and talk to the root MDs to collect data as described earlier. After all data are collected, DC encrypts M_j for each MD_j it collects using K. Note that $M_j = SKE(GK_i, g^{d_j})||SIG(MD_j, g^{d_j})||DATA_j$, which belongs to the tree of MD_i, contains enough information for the PO to retrieve and verify the data.

E. Security Discussion and Its Complexity

In our protocol, the confidentiality of the data is ensured through the Diffie-Hellman keys. The DH half keys are transmitted under encryption. Eavesdroppers cannot read them. The data remains confidential to the *honest-but-curious* DC and intermediate MDs as well because, although they can read the public DH keys exchanged between the PO and the MD, they cannot establish the DH key themselves. Key information from the root to the leaves are authenticated through signatures, while data from the leaves to the root are authenticated through the hashes in a hop-by-hop manner. Although a tampering on data can be detected by the PO eventually, by authenticating the data in a hop-by-hop manner, tampering can be identified

early so that network resources would not have been spent on transmitting the tampered data from the leaf to the PO. Our protocol is thus secure from both active and passive attacks when the keys are not leaked. (We consider the risk of leaking the group key in our problem formulation in Sec. IV-A).

We now study the computational complexity of an MD. We only consider public key and DH operations since symmetric key and hash operations are not expensive. The leaf MD receives one message from its parent and sends one message to its parent in each data collection cycle. It has to decrypt the key information and verify two signatures for the message received. To prepare the reply, it generates a DH half key and signs it. Five operations are needed in total for a leaf node. Similar operations are needed for the messages to and from the parent for each non-leaf MD. To send messages to its children, an MD has to sign the key information. For each reply from a child, it has to verify the signature of the half key provided by the child. Therefore, the total number of operations a non-leaf MD needs to perform is 6 + number of children.

IV. Collection Tree Formation

A. Problem Description

In this section, we describe how PO constructs the disjoint data collection trees on the network connectivity graph $\mathcal{G}$ of MDs. Since the PO has complete information of the topology, graph $\mathcal{G}$, a connected vertex set of all MDs, is known *a priori*. We use set $\mathcal{M} = \{1, 2, \cdots\}$ to denote the index set of these MDs. Furthermore, if there is a direct connectivity between two MDs then an undirected edge exists between them in graph $\mathcal{G}$. Furthermore, we define the set of all MDs that can potentially be selected as the *root* MDs *candidate set*, and we use $\mathcal{R}$ to denote the set of indices of these candidate root MDs. The largest possible candidate set consists of all the MDs that can be physically within the communication range of the DC when it is traveling along the predefined data collection path. For example, in Fig. 1, $\mathcal{R} = \{1, 2, 3\}$.

Our objective is to minimize the sum of the data collection time over all trees. We further assume identical link delays. Then, we can use the depth of a data collection tree to represent the time needed to collect data from this particular tree and the summation of the depths over all constructed trees to represent the overall data collection time.

Apart from the time to collect data, we also consider the security level of the group key. As the MDs may not be in a very secure physical environment, there is a risk of leaking the group key. If the group key is stolen, the attacker can decrypt $SKE(GK, g^{d_k})$ in M_k to obtain g^{d_k} and create the correct hash of fake data. Although the PO can finally detect the data were not legitimate, network resource will be wasted in transmitting the message. Assume that every MD_i $(i \in \mathcal{M})$ leaks the group key with probability p_i. Then the probability that the group key of a tree $\mathcal{T}$ is leaked is $P_{leak}(\mathcal{T}) = 1 - \prod_{i:MD_i \in \mathcal{T}} (1 - p_i)$. To ensure the security level of every constructed tree $\mathcal{T}$, we limit $P_{leak}(\mathcal{T})$ to be no larger than some predefined threshold. We assume identical p_i's. Then, if we limit the group key leaking probability to be no larger than some threshold value $P_{threshold}$, we have:

$$P_{leak}(\mathcal{T}) = 1 - \prod_{i:MD_i \in \mathcal{T}} (1 - p_i) = 1 - (1 - p)^{|\mathcal{T}|} \leq P_{threshold}$$

In other words, the cardinality of every constructed tree should satisfy

$$|\mathcal{T}| \leq \frac{\log(1 - P_{threshold})}{\log(1 - p)} = N_{threshold} \tag{1}$$

As detailed in the next subsection, we formulate the optimization problem with a $\texttt{Min} - \texttt{Sum} - \texttt{Max}$ objective to minimize the summation of the depths over all constructed trees with the aforementioned security constraint.

B. Mathematical Formulation

We state the optimal collection tree formation (**CTF**) problem into a binary integer programming formulation as follows:

$$\min \sum_{j \in \mathcal{R}} \left(\max_{i \in \mathcal{M}} \sum_{k \in \mathcal{K}_{ij}} x_{ij}^k L_{ij} \right) \tag{2}$$

$$\text{s.t.} \sum_{i \in \mathcal{M}} \sum_{k \in \mathcal{K}_{ij}} x_{ij}^k \leq N_{threshold}, \forall j \in \mathcal{R} \tag{3}$$

$$\sum_{j \in \mathcal{R}} \sum_{k \in \mathcal{K}_{ij}} x_{ij}^k = 1, \forall i \in \mathcal{M} \tag{4}$$

$$x_{ij}^k \leq x_{mj}^l, \forall i \in \mathcal{M}, m \in \mathcal{N}_i, \mathcal{P}_{mj}^l \subseteq \mathcal{P}_{ij}^k \tag{5}$$

$$x_{ij}^k \in \{0, 1\}, \forall i \in \mathcal{M}, j \in \mathcal{R}, k \in \mathcal{K}_{ij} \tag{6}$$

$\mathcal{N}_i$: the set of neighbors of MD_i. $\mathcal{P}_{ij}^k$: the k_{th} shortest path from MD_i to MD_j in graph $\mathcal{G}$. There can be multiple equal length shortest paths between any pair of MDs in graph $\mathcal{G}$. For example, in Fig. 1 two shortest paths, $MD_8 \to MD_2 \to MD_1$ and $MD_8 \to MD_6 \to MD_1$ exist between MD_8 and MD_1. Then, we have $\mathcal{P}_{18}^1 = MD_8 \to MD_2 \to MD_1$ and $\mathcal{P}_{18}^2 = MD_8 \to MD_6 \to MD_1$. $\mathcal{K}_{ij}$: set of indices of the shortest paths between MD_i and MD_j. For example, in Fig. 1 because two shortest paths exist between MD_8 and MD_1, we have $\mathcal{K}_{81} = \{1, 2\}$. L_{ij}: number of MDs in the shortest paths between MD_i and MD_j. In the example in Fig. 1, we have $L_{18} = 3$. With Eq. 1, $N_{threshold}$: maximum number of MDs allowed in one tree.

We associate a binary variable x_{ij}^k with every shortest path $\mathcal{P}_{ij}^k$ between MD_i $(\forall i \in \mathcal{M})$ and MD_j $(\forall j \in \mathcal{R})$. For all MD_i's out of the candidate set and MD_j's in the candidate set, that is $\forall i \in \mathcal{M} \setminus \mathcal{R}$ and $\forall j \in \mathcal{R}$, we have

$$x_{ij}^k = \begin{cases} 1 & \text{if } MD_i\text{'s data is collected along } \mathcal{P}_{ij}^k \text{ to } MD_j \\ 0 & \text{if } MD_i\text{'s data is not collected along } \mathcal{P}_{ij}^k \text{ to } MD_j \end{cases}$$

Furthermore, for all MD_i belonging to the candidate set, that is $\forall i \in \mathcal{R}$, we have $x_{ii}^1 = 1$ if MD_i is selected as a root; otherwise, $x_{ii}^1 = 0$.

In the objective function (2), $\max_{i \in \mathcal{M}} \sum_{k \in \mathcal{K}_{ij}} x_{ij}^k L_{ij}$ for some fixed $j \in \mathcal{R}$ represents the depth of the tree rooted at candidate root MD_j. Furthermore, $\max_{i \in \mathcal{M}} \sum_{k \in \mathcal{K}_{ij}} x_{ij}^k L_{ij} = 0$ if MD_j is not chosen as the root of any collection tree. Therefore, by summing over $\forall j \in \mathcal{R}$, we have the objective function (2) representing the total depth over all constructed trees.

The inequality constraints (5) ensure that any MD_i can send data to MD_j over path $\mathcal{P}_{ij}^k$ only if one of MD_i's neighbors MD_m chooses to send data to MD_j over $\mathcal{P}_{mj}^l$, a sub-path of path $\mathcal{P}_{ij}^k$. Refer to Fig. 1, suppose $\mathcal{P}_{51}^1 = MD_5 \rightarrow MD_4 \rightarrow MD_1$ and $\mathcal{P}_{41}^1 = MD_4 \rightarrow MD_1$ with $\mathcal{P}_{41}^1 \subseteq \mathcal{P}_{51}^1$, then we have $x_{51}^1 \leq x_{41}^1$, which means MD_5 can choose to send data to MD_1 over path $\mathcal{P}_{51}^1$ only if MD_4 chooses to send data to MD_1 along path $\mathcal{P}_{41}^1$.

In a word, the single path constraints (4) and the sub-path constraints (5) together ensure that we construct multiple disjoint trees rooted at all or a subset of the MDs in the candidate set. Furthermore, the security constraints (3) further ensure that the number of nodes in every constructed tree is no larger than the threshold.

C. Solution and Analysis

To solve **CTF**, we firstly transform it into the standard format by rewriting (2) as follows:

$$\min \sum_{j \in \mathcal{R}} z_j \tag{7}$$

$$\text{s.t. } z_j \geq \sum_{k \in \mathcal{K}_{ij}} x_{ij}^k L_{ij}, \forall i \in \mathcal{M}, j \in \mathcal{R} \tag{8}$$

(7) and (8), together with constraints (3) to (6), form the **modified CTF**. The modified-CTF problem is equivalent to the original CTF problem in terms of the optimal objective function value. Nevertheless, the mixed-integer programming problem is an NP-hard problem. Thus, we propose to use an approximation algorithm by means of a linear relaxation based iterative rounding (LR-IR) [21] as shown in Algorithm 1. It is

Algorithm 1: LR-IR for Modified-CTF

Input: $\mathcal{G}$, $N_{thershold}$, $\mathcal{R}$
Output: Modified-CTF $\{x_{ij}^k\}$

```
while true do
    solve the LP relaxation of the modified-CTF with
    x_ij^k ∈ [0,1] and get optimal solution {x_ij^k*};
    round the largest fractional solution within {x_ij^k*} to 1;
    if x_ij^k* ∈ {0,1} (∀i ∈ M, j ∈ R, k ∈ K_ij) then
        return {x_ij^k*};
    end
end
```

obvious that the worst-case number of iterations of Algorithm 1 is $O(N)$ with respect to the number of decision variables x_{ij}^k. Also, Algorithm 1 terminates when each x_{ij}^k equals to 0 or 1.

V. Simulation

In our simulation, we use the SG data set for Washington, DC [22], which contains the exact positions of all the utility poles in the city. We extract the positions of 300 utility poles from two portions of the map, as shown in Fig. 5 and Fig. 6.

Specifically, there are 8 stars in both figures representing MDs in the candidate root set. In Fig. 5, we assume that the DC is traveling along the southernmost street. In Fig. 6, the DC is assumed to be traveling along the westernmost street. Then, we pick evenly distributed 8 MDs along the two streets

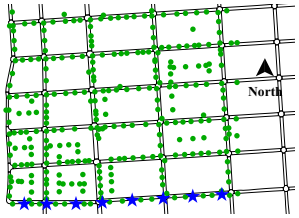


Fig. 5. MD Topology I.

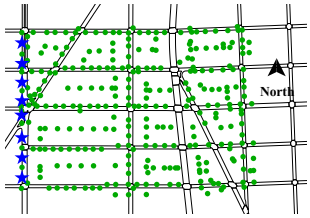
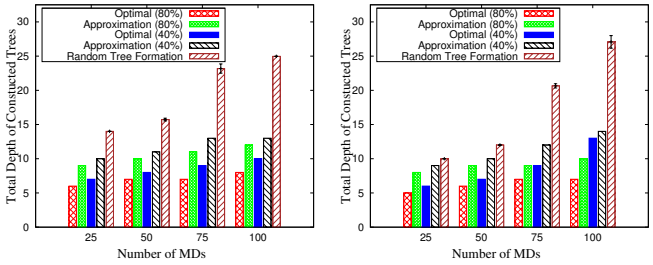


Fig. 6. MD Topology II.

to be the candidate root sets. Furthermore, we assume that MDs communicate with each other wirelessly with an identical communication range of $100m$. Also, topology II has a large MD density($826/km^2$) than than topology I ($722/km^2$). We used GUROBI solver [23] for our simulations.

In our simulation, we construct a scenario by choosing $|\mathcal{M}|$ MDs nearest to the reference streets. For example, in topology I, 50 MDs that are nearest to the southernmost street in Fig. 5 are selected to construct the scenario.



(a) MD Topology I (b) MD Topology II

Fig. 7. Comparison of Total Tree Depth.

In Fig. 7, we compare the total depth of the constructed trees in scenarios with different number of MDs generated by three algorithms: The optimal algorithm is the solution from GUROBI or the modified-CTF problem. The approximation algorithm refers to Algorithm 1. In the random tree formation algorithm, every MD randomly selects a neighbor which has been included in a tree to be its parent. Also, in Fig. 7, two cases are considered in both topologies, namely $N_{threshold} = 40\% \times |\mathcal{M}|$ and $N_{threshold} = 80\% \times |\mathcal{M}|$. For example, if we assume that every MD leaks the group key with probability $p = 0.01$ and $|\mathcal{M}| = 100$, then $N_{threshold} = 40\% \times |\mathcal{M}|$ means we set the threshold probability $P_{threshold} = 1 - (1-p)^{40\% \times |M|} = 1 - (1-0.01)^{40\% \times 100} = 0.331$.

In Fig. 7, our approximation algorithm yields a total tree depth which is much smaller than that of the random tree formation algorithm while staying fairly close to the optimal value. As the maximum tree size ($N_{threshold}$) increases, both the optimal and the approximation algorithm tend to yield decreasing values of the total tree depth. This observation captures the trade-off between efficiency and security of our optimization formulation. We will further show it in Fig. 9.

The comparison of the completion time (in terms of seconds) of the optimal algorithm and the approximation algorithm with different numbers of MDs is provided in Table I. We can readily observe that completion time of the optimal algorithm increases exponentially with the number of MDs. In contrast, our approximation algorithm has much lower

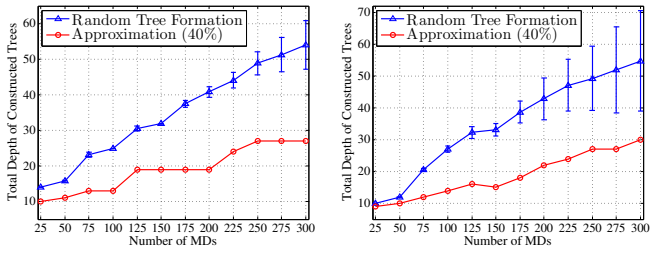
$\|\mathcal{M}\|$	25	50	75	100
Top. I Optimal	1.953	73.956	1418.134	6422.439
Top. I Approx. (40%)	0.048	1.892	12.309	15.319
Top. II Optimal	2.332	1650.730	6503.994	36009.112
Top. II Approx. (40%)	0.089	1.807	4.235	5.254

TABLE I
ALGORITHM COMPLETION TIME IN SECONDS.

completion time.

Furthermore, from Table I, we observe that when $|\mathcal{M}| = 100$, the optimal algorithm already has to take more than 6000s to complete. This indicates that when solving the modified-CTF problem for a network of more than 100 MDs, it is better to turn to our approximation algorithm. Hence, we carry out simulation in scenarios with up to 300 MDs whereby our approximation algorithm terminates in reasonable time period.

From Fig. 8, we can observe that the approximation algorithm outperforms the baseline random tree formation algorithm in terms of the total tree depth when the number of MDs is larger than 100.



(a) MD Topology I. (b) MD Topology II.

Fig. 8. Comparison of Total Tree Depth (Approximation vs. Random).

In Fig. 9, we fix the number of MDs to be $|\mathcal{M}| = 50$ and vary the maximum percentage of MDs in one tree, $\frac{N_{threshold}}{|\mathcal{M}|}$

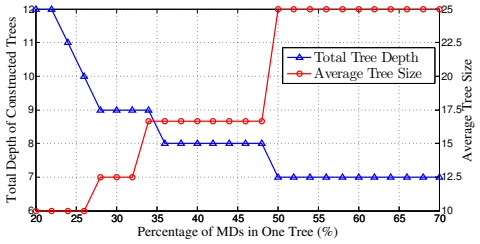


Fig. 9. Efficiency and Security Tradeoff

from 20% to 70%. As this ratio increases, we see an increasing trend of the average constructed tree size and a decreasing trend of the total tree depth. This observation reflects the trade-off between efficiency and security of data collection inherently in our problem formulation. More specifically, if we allow MDs to form trees with larger sizes then the total depth of constructed trees tends to be smaller.

VI. CONCLUSION

In this paper, we develop a secure and efficient protocol to collect data from measurement devices (MD) in a multi-hop manner through a mobile data collector (DC). MDs report data via trees rooted at the MDs that have direct communication with the DC. We formulate the secure and optimal tree construction problem as an integer linear programming, and develop an approximation algorithm to compute a solution. The simulations conducted using real topologies reflect that our algorithm performs well and efficiently.

ACKNOWLEDGMENT

This material is based upon work supported in part by the Department of Energy under Award Number DE-OE0000097 and the Ralph and Catherine Fisher grant. King-Shan Lui's work was funded in part by the Small Project Funds (Project No: 201109176048) of the University of Hong Kong.

REFERENCES

[1] N. Kayastha, D. Niyato, E. Hossain, and Z. Han, "Smart grid sensor data collection, communication, and networking: a tutorial," *Wireless Communications and Mobile Computing*, pp. n/a–n/a, 2012.

[2] R. Tabassum, K. Nahrstedt, E. Rogers, and K.-S. Lui, "Scapach: Scalable password-changing protocol for smart grid device authentication," in *IEEE ICCCN 2013*, July 2013, pp. 1–5.

[3] V. Jain and G. Chapman, "Massively deployable intelligent sensors for the smart power grid," in *IEEE VLSI*, Oct 2010, pp. 319–327.

[4] M. Conti, A. Passarella, and L. Pelusi, "Mobile-relay forwarding in opportunistic networks," in *Ch. 13 Adaptive Techniques in Wireless Networks*. CRC Press, August 2008.

[5] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.

[6] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, "A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 5, pp. 809–817, May 2012.

[7] A. Bogdanov, E. Maneva, and S. Riesenfeld, "Power-aware base station positioning for sensor networks," in *IEEE INFOCOM*, March 2004.

[8] R. Farahani and M. Hekmatfar, *Facility Location: Concepts, Models, Algorithms and Case Studies*. Physica, 2009.

[9] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 1415, pp. 2826 – 2841, 2007.

[10] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[11] G. Dan, K.-S. Lui, R. Tabassum, Q. Zhu, and K. Nahrstedt, "SELINDA: A secure, scalable and light-weight data collection protocol for smart grids," in *Proc. of IEEE SmartGridComm*, 2013.

[12] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *IEEE INFOCOM 2007*, May 2007, pp. 2045–2053.

[13] W. He, H. Nguyen, X. Liu, K. Nahrstedt, and T. Abdelzaher, "iPDA: An integrity-protecting private data aggregation scheme for wireless sensor networks," in *IEEE MILCOM 2008*, Nov 2008, pp. 1–7.

[14] T. Feng, C. Wang, W. Zhang, and L. Ruan, "Confidentiality protection for distributed sensor data aggregation," in *IEEE INFOCOM*, April 2008.

[15] M. Groat, W. He, and S. Forrest, "Kipda: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *IEEE INFOCOM*, April 2011, pp. 2024–2032.

[16] Y. Yan, Y. Qian, and H. Sharif, "A secure and reliable in-network collaborative communication scheme for advanced metering infrastructure in smart grid," in *IEEE WCNC*, March 2011, pp. 909–914.

[17] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies*. Springer, 2011, vol. 6794, pp. 175–191.

[18] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *IEEE SmartGridComm*, 2010.

[19] C. Rottondi, G. Verticale, and C. Krauss, "Distributed privacy-preserving aggregation of metering data in smart grids," *IEEE JSAC*, vol. 31, no. 7, pp. 1342–1354, July 2013.

[20] H. Nicanfar, A. Alasaad, P. Talebifard, and V. Leung, "Network coding based encryption system for advanced metering infrastructure," in *IEEE ICCCN*, July 2013, pp. 1–7.

[21] J. Cheriyan, S. Vempala;, and A. Vetta, "Network design via iterative rounding of setpair relaxations," *Combinatorica*, pp. 255–275, 2006.

[22] "Wash., dc data set," 2014. [Online]. Available: http://data.octo.dc.gov/

[23] "GUROBI Solver," 2014. [Online]. Available: http://www.gurobi.com/