Optimal Method for Detecting Collusive Saboteur Smart Meters in Smart Grid

El Yazid Dari[1,*], Ahmed Bendahmane[2] and Mohamed Essaaidi[3]

[1]*Faculty of Sciences, Abdelmalek Essaadi University, Tetuan, Morocco*
[2]*Laboratory of Applied Sciences and Education. Higher Normal School (ENS), Abdelmalek Essaadi University, Tetuan, Morocco*
[3]*College of IT (ENSIAS), Mohamed 5 University, Rabat, Morocco*
E-mail: da.elyazid@gmail.com; ab.dahmane@gmail.com; mohamed.essaaidi@um5.ac.ma
[*]*Corresponding Author*

Received 04 May 2019; Accepted 16 January 2020;
Publication 23 March 2020

Abstract

Smart grid is a system in which it is possible to use voting-based techniques to resist sabotage of several cyber-attacks. The adaptation of these techniques can be difficult and useless in the case when the malicious resources (i.e., smart meters) of this system can return wrong data in same time; however, the collusion problem is triggered. To detect and resolve the collusive issue, spot-checking technique has been proposed by sending randomly certain number of spotter queries to chosen resources with known correct data in order to estimate resource credibility based on the returned data. This work proposes an original method that resist against collusion attacks by using probability to solving a new spot-checking optimization problem for smart grid systems, with the objective to minimize probability of accepting wrong data (PAWD) while respecting an expected overhead constraint. The proposed solution contains an optimal combination of several parameters, the number of spotter queries sent, the number of resources tested by each spotter query, and the number of resources assigned to run the genuine query. The

Journal of Cyber Security and Mobility, Vol. 9_2, 237–264.
doi: 10.13052/jcsm2245-1439.923

optimization procedure includes a new method for evaluating performance metrics of PAWD and expected overhead in terms of the total number of query assignments. To demonstrate the proposed optimization problem and solution procedure, we have provided several illustrative examples.

Keywords: Smart grid, smart meters, cyber-security, vulnerability, data-integrity, probability, optimization.

Abbreviations

SGN: smart grid
MSM: malicious smart meters
PAWD: probability of accepting wrong data

Nominations

S_i: number of smart meters remained in the SGN list after running *i* spotter queries
M_i: number of smart meters remained in the MSM list after running *i* spotter queries
Q: number of spotter queries
T: number of smart meters tested in each spotter query
G: number of smart meters assigned to run genuine query
E_{Sp}: expected number of spotter query assignments during the sending of spot-queries
E_{Gu}: expected number of genuine query assignments
E: total expected number of query assignments $(E = E_{Sp} + E_{Gu})$
E_m: maximum expected number of query assignments
SQ_n: probability that exactly n MSMs are detected during the spot-query-sending process
C(G,m): conditional PAWD given that m MSMs are chosen for the genuine query
$P_i(n)$: probability that exactly n MSMs are detected after completion of i spotter queries
D_e: detection error, probability that the smart meters will be not detected after spotter queries.
W: PAWD

1 Introduction

Smart grid (Figure 1) offers a platform to execute intensive tasks by different shared resources, such as smart meters, in a distributed and parallel manner that makes it vulnerable to sabotage attacks. However, these sabotage attacks can be performed by data modification, data destruction and false data injection [1, 2]. The smart grid resources manipulation might be sabotaged by injecting false data and by attempting to get favourable results by accessing smart meters' applications and by modifying the generated data. This may have various effects such as changing the price of electricity consumed, decreasing the amount of electricity consumed or even causing a blackout. Thus, smart grid security requirements are really very high in spite of several existing of security mechanisms to provide confidentiality, to secure communication between all smart grid resources, and to provide the integrity of the data generated by smart meters.

Nowadays, several sabotage-tolerance techniques have been applied in many grid platforms such as volunteer computing systems [3, 4], desktop grids [5–7], and peer-to-peer grids [8–10], to deal with the verification of the reliability of job results in various grid systems. Voting-based techniques using spot-checking are the most applied ones to resist sabotage, assuring or at least improving reliability of computations and returned data [3], such

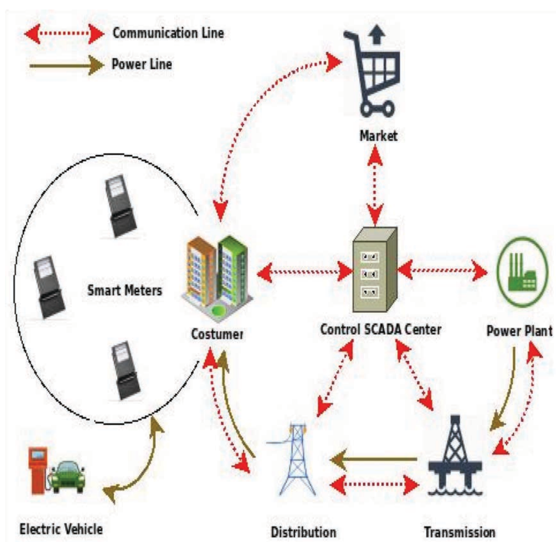

Figure 1 Smart grid.

as majority voting, m-first voting [11] and credibility-based voting [3] with reputation system [12]. However, these techniques rely on the assumption that the grid resources behave independently which is not valid where a number of collusive resources (smart meters in our work) collectively return the same wrong result (data in our work). Furthermore, sometimes the decision of voting-based techniques may not be exact as cited in [13–16]. Therefore, one group of smart meters in a smart grid might develop some form of collective misbehaviour to sabotage the circulated data by returning a wrong data that is considered the same wrong results in our work, failing the voting mechanism [17–20]. In fact, in order to deal with this threat it is necessary to explore the sabotage-tolerant techniques against collusive smart meters in a smart grid. To deal with collusion attacks, various approaches have been proposed, for example, there are approaches based on credibility [3, 9, 21], scheduling and certification [14, 22], reputation systems [12, 13, 16], and graph clustering [23]. To the best of our knowledge, no work has been realized by using probability to optimize the spot-checking policy to test the integrity of the data sent by smart meters, taking into account expected overhead for smart grid systems under the collusion attacks.

This work precedes other work, an overview of smart grid cyber-security [24]. Then it gives a new method by formulating and solving a new spot-checking optimization problem in smart grid, which finds an optimal combination of query distribution parameters including the number of deployed spotter queries, the number of smart meters tested by each spotter query, and the number of smart meters assigned to perform the genuine query. The objective of this optimization problem is to minimize probability of accepting wrong data (PAWD) taking into account on expected overhead.

The rest of the paper is organized as follows. Section 2 presents the system model and the problem to be addressed in this work. Section 3 describes the algorithm proposed for evaluating system performance metrics of PAWD and expected overhead. Section 4 presents the formulation of the considered optimization problems. Section 5 gives illustrative examples to demonstrate the proposed optimization problems. Finally, section 6 presents the conclusion of the work.

2 System Model and Problem Statement

In this section, we define our proposed system model, in which we describe all their components. In other hand, we describe in details the statement problem that to be dealt in this paper.

2.1 System Model

The smart grid (Figure 1) is an emerging technology that is revolutionizing the conventional electric grid by providing new services based mainly on Information and Communication Technologies (ICT) [14]. This is a very complex network for energy generation, transmission, distribution and consumption.

In Figure 2, we show the basic components of a smart grid system, which consists of N Smart meters. Each Smart meter is connected to a several local devices or appliances, which are able to communicate with other smart meters via a home area network (HAN). These smart meters analyze and measure data about energy usage, number of appliances, energy pricing, etc. and stores all these data periodically. In addition, these stored data must be returned to a Supervisory Control and Data Acquisition (SCADA) Center in order to monitor and control these collected data in real-time, and provide high security information communication between users and utilities.

The quantity of the energy consumed in a location is calculated by the smart meter linked to this location, taking into account the price of the energy on the market during the specific time of this consumption. In addition, the HAN can supply and sell extra energy to the power grid in order to benefit from high-energy prices in electricity markets that encourage competition among power suppliers [24].

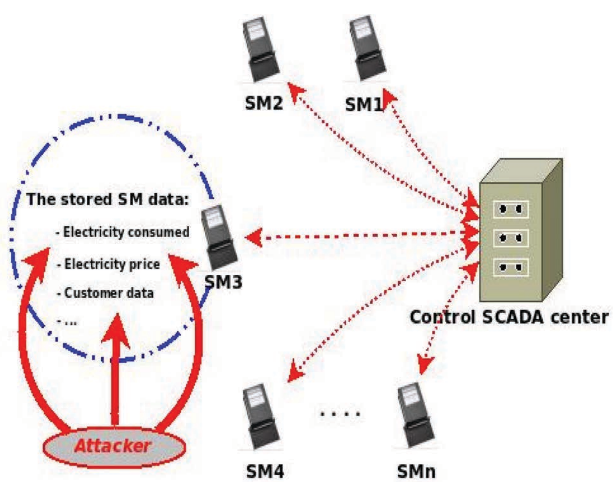

Figure 2 Basic components of a smart grid.

2.2 Problem Statement

In smart grid, malicious attackers or users who can trigger security attacks to its infrastructure, knowing that the attacker tries to corrupt as many smart meters as possible, but its capability is limited. The new approach we propose, has the objective to minimize probability of accepting wrong data (PAWD), and is based on a spot-checking technique.

In our grid, we focus on smart meters attacks that compromise the provided resources and services by exploiting some of their vulnerabilities. In this attack model, we have an adversary, which can compromise the smart meters, and instruct them to behave maliciously to tamper their data and to return wrong data to the SACDA control center. We assume that all malicious smart meters always collude with one another to return incorrect data, the "honest" smart meters always produce the same correct data for the same query at the same time t, and the SCADA control center sends randomly queries to smart meters to return their data at a time t. The smart meter memory stores both their data at random instances, and all data sent to the SCADA control center.

At first, SCADA control center sends the query 0 to all smart meters of SGN and stores the received data in order to initialize the data to be tested in the next query. Then, when a smart meter receives one of these queries, it executes respectively the following tasks:

- Step 1. It captures existing data at the same time t;
- Step 2. It returns this data captured at this time t and those stored at the time $t - 1$ (stored in the memory of the smart meter);
- Step 3. It stores the data captured at this time t in its memory.

Then, the SCADA control center receives the data sent by the smart meters, stores the data captured at time t, and checks whether the data captured at instant $t - 1$ is identical to the data already stored. If the data sent at time t and those sent at time $t - 1$ by the same SM and for the same request are the same, SCADA considers them as correct data and it stores them, otherwise, it considers them as wrong data and it refuses them. The new method we propose, has the objective to minimize probability of accepting wrong data (PAWD).

The smart grid consists of a static set of S0 smart meters, M0 of which are malicious. The malicious smart meters collude in producing wrong data to reduce the efficiency of the voting-based technique against sabotage. To detect the colluding malicious smart meters (MSM), the SGN sends Q spot-checking (spotter) queries with known output data to randomly chosen smart meters. In addition, the SGN cannot distinguish genuine and spotter queries and, therefore, generate to any query identical wrong outputs data.

In our grid, each spotter query *i* is sent to *T* smart meters, where *T* is a predetermined parameter. The smart meters for each spotter queries are chosen independently. If several smart meters produce identical wrong output data for the i-th spotter query, they are detected as MSMs and removed from the list of smart meters. Thus, after the i-th spotter query the list contains $S_i \leq S_0$ smart meters, $M_i \leq M_0$ of which are MSMs. Then a new $i+1$-th spotter query is sent to *T* smart meters. Let *G* represent the number of smart meters assigned to perform the genuine query. After completing *Q* spotter queries, the genuine query is sent to *G* smart meters randomly chosen from the list of S_Q remaining smart meters. The data that gets the majority of votes is accepted. If the wrong output data of the genuine query is accepted, the genuine query fails.

The problem addressed in this work is to find the combination of queries distribution policy parameters T, G and Q that minimizes the PAWD subject to constrained expected overhead. The overhead is proportional to the total number of queries assignments including the spotter query and the genuine query assignments.

3 Algorithm for PAWD Evaluation

Before the assignment of the *i-th* spotter query, the list contains $\mathrm{S_{i-1}}$ smart meters, $\mathrm{M_{i-1}}$ of which are MSMs. The probability that m MSMs are among $\mathrm{M_{i-1}}$ smart meters randomly chosen for the *i-th* spotter query is:

$$SQ(S_{i-1}, M_{i-1}, T, m), \tag{1}$$

where

$$\begin{aligned} SQ(a,b,c,d) &= \frac{C_b^d . C_{a-b}^{c-d}}{C_a^c}, \quad &&\text{if } c \geq d \\ SQ(a,b,c,d) &= 0, &&\text{if } c < d \end{aligned} \tag{2}$$

and *m* can vary from *0* to *T*.

Let $P_i(m)$ represent the probability that exactly *m* MSMs are detected after completion of *i* spotter queries. Initially there are S_0 smart meters in the list with M_0 of them being MSMs.

Thus,

$$P_1(m) = SQ(S_0, M_0, T, m) \tag{3}$$

Therefore, for $1 < i \leq Q$:

$$P_i(m) = \sum_{v=0}^{m} P_{i-1}(v) \cdot SQ(S_0 - v, M_0 - v, T, m - v) \tag{4}$$

Finally, $P_Q(m)$ for $m = 0, \ldots, M_0$ gives the probability that m MSMs are detected after the spot-checking procedure after *Q* queries.

On the other hand, the probability that the smart meters will not be detected is defined as follow:

$$D_e = 1 - P_Q(m) \tag{5}$$

In addition, the expected number of assignments during the *Q* spotter queries executions is:

$$E_{Sp}(T, Q) = T + \sum_{i=2}^{Q} \left(\sum_{m=0}^{M} P_{i-1}(m) \cdot T \right) \tag{6}$$

And, the expected number of assignments during the *Q* spotter query executions is:

$$E_{Gu}(G, T, Q) = \sum_{m=0}^{M} P_Q(m) \cdot G \tag{7}$$

According to the expressions above, the distribution of the number of detected MSMs after the execution of *Q* spotter queries and the expected number of assignments during the spot-checking can be obtained using the following procedure (Algorithm 1).

If after the spot-checking procedure *m* MSMs are detected, the SGN sends the genuine query to *G* out of S_0–m remaining smart meters with M_0–m of them being MSMs. The probability that w out of *G* chosen smart meters are MSMs is:

$$SQ(S_0 - m, M_0 - m, G, w), \tag{8}$$

If *w* MSMs and *G*–*w* honest smart meters are chosen, the wrong output data can be accepted if $w > G - w$. Thus, the conditional probability of accepting wrong data (PAWD) given that *w* MSMs are chosen for the genuine query is:

$$C(G, w) = \begin{cases} 0, & \text{if } w > G/2 \\ 1, & \text{if } w \leq G/2 \end{cases} \tag{9}$$

Algorithm 1: For detecting MSMs and the expected number of assignments

```
Begin
    For m = 0 to M do
        set P_1(m) = SQ(S_0,M_0,T,m);
    End For
    E_Sp = T;
    For j = 2 to Q do
        For l = 0 to M_0 do
            set P_j(l) = 0;
        End For
        For i = 0 to M_0 do
            set E_Sp = E_Sp + P_{j-1}(i)·T;
            For k = 0 to i do
                set P_j(i) = P_j(i) + P_{j-1}(k) · SQ(S_0 - k, M_0 - k, T, i - k);
            End For
        End For
    End For
End
```

The conditional PAWD given that *m* MSMs are detected is:

$$\sum_{w=0}^{G} SQ(S_0 - m, M_0 - m, G, w) \cdot C(G, w) \tag{10}$$

Thus, the overall PAWD is:

$$W(T, G, Q) = \sum_{m=0}^{M_0} \left(P_Q(m) \cdot \sum_{w=0}^{G} SQ(S_0 - m, M_0 - m, G, w) \cdot C(G, w) \right) \tag{11}$$

Notice that the algorithm above can be used to model a situation where the MSMs cannot distinguish spotter and genuine queries.

4 Optimization Problem Formulation

Based on estimation of attack parameter M_0, the SGN can solve the following optimization problem: find T, G and Q that minimize W(G,T,K) s.t. overhead constraint:

$$E(G, T, Q) = E_{Sp}(T, K) + E_{Gu}(G, T, Q) \leq E_{opt} \tag{12}$$

Algorithm 2: For obtaining the SGN policy parameters: G_{opt}, T_{opt} and Q_{opt}

```
Begin
    For k = 1 to M do
        set W_min(k) = 1;
    End For
    For g = 1 to S do
        For t = 1 to S do
            For q = 1 to Q do
                For m = 1 to M do
                    EGu = 0;
                    For i = 0 to m do
                        E_Gu = E_Gu + P_q(i) * g;
                    End For
                    E = E_Gu + E_Sp;
                    if W(t,g,q,m) < W_min(k) and E(g,t,q) < E_f then
                        G_opt = g;
                        T_opt = t;
                        Q_opt = q;
                        E_opt = E(g, t, q);
                        W_min(k) = W(t, g, q, m);
                    End if
                End For
            End For
        End For
    End For
End
```

Thus, the most conservative SGN policy for any M_0 is to minimize:

$$G_{opt}, T_{opt}, Q_{opt} = arg \min_{G,T,Q} (\beta(T, G, Q, M_0)) \tag{13}$$

s.t.

$$E(G, T, Q, M_0) \leq E_{opt} \tag{14}$$

Finally, the following procedure (cf. Algorithm 2) obtains the SGN policy parameters: G_{opt}, T_{opt} and Q_{opt}:

5 Illustrative Examples

Figures 3 and 4 present D_e and E_{Sp} after achievement of spotter queries. Figure 3 shows D_e as a function of the number of malicious smart meters varied between 10 and 90 for $N_0 = 100$, and different combinations of

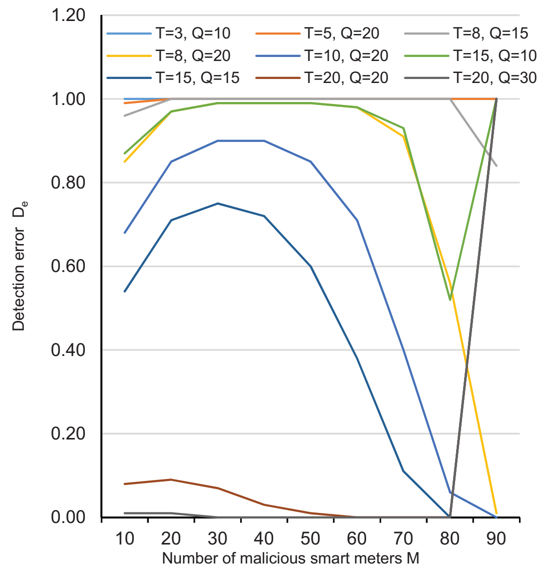

Figure 3 Detection error ε_s as a function of number of malicious smart meters for $S_0 = 100$ and different T and Q.

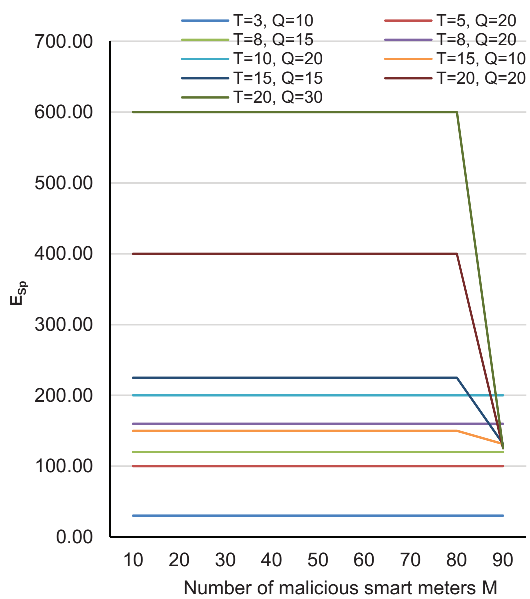

Figure 4 E_{Sp} as a function of number of malicious smart meters for $S_0 = 100$ and different T and Q.

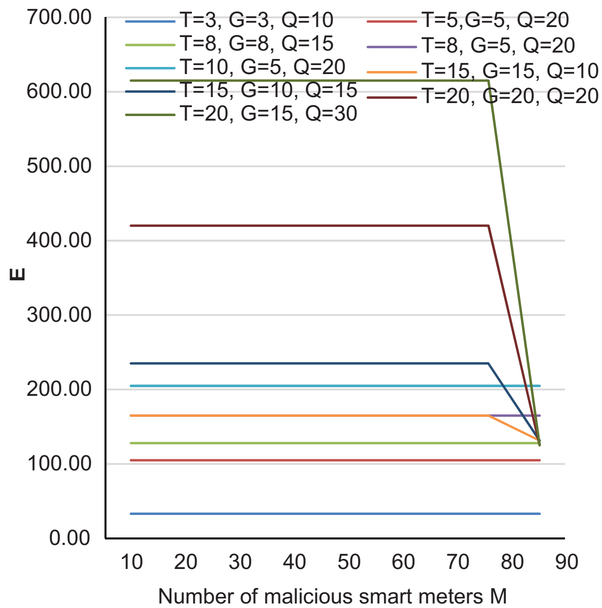

Figure 5 E as a function of number of malicious smart meters for $S_0 = 100$ and different T, G and Q.

T and Q. To detect all MSM it is necessary to choose carefully the good combination of T and Q, i.e. if we choose a small value of T, we must increase the number of sent queries, and otherwise, the D_e detection error will be remarkable. On the other hand, if we increase T, it will not exceed the number of MSM remaining in the list, otherwise, the probability of detecting all MSM tends to zero, i.e. D_e will be remarkable too, which explains the great values of D_e for $M = 90$.

We notice that all values of MSM have a combination of T and Q, which error detection value D_e tends to zero.

Figure 4 presents expected number of spotter query assignments as a function of malicious smart meters M, $E_{sp} = T \cdot Q$ always holds for the considered T and Q from $M = 10$ to $M = 80$, because the list contains more smart meters than those are used in any spotter queries. In the case $M = 90$, the list of smart meters can contain fewer than T, which explains the decrease of E_{sp}.

Figures 5 and 6 present E and W after achievement of genuine query. Figure 5 shows E as a function of the number of malicious smart meters

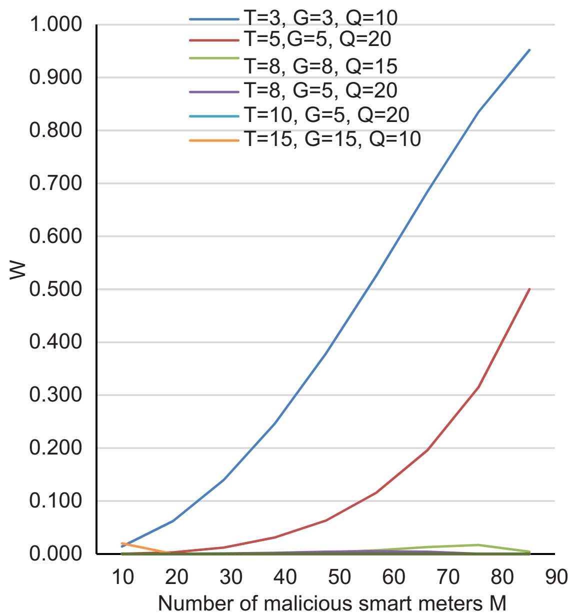

Figure 6 W as a function of number of malicious smart meters for $\mathrm{S}_0 = 100$ and different T, G and Q.

varied between 10 and 90 for $N_0 = 100$, $E = T \cdot Q + G$ always holds for the considered *T*, *G* and *Q* from $M = 10$ to $M = 80$, because the list contains more smart meters than those used. In the case $M = 90$, the list of smart meters can contain fewer than *T* and *Z*, which explains the decrease of *E*.

Figure 6 shows *W* as a function of the number of malicious smart meters varied between 10 and 90 for $\mathrm{N}_0 = 100$, all combinations have a value of W thanks to the null value of $\mathrm{D_e}$ and to a good choice of G. So, if $\mathrm{D_e}$ decreases and G increases then W decrease.

Finally, we deduce that in order to obtain a minimum value for all number of malicious smart meters, we must choose carefully the good values of the parameters T, G and Q.

6 Optimal Solutions for the Maximum Value $\mathbf{E_m}$

In this section we will present the optimization of the parameters *T*, *G*, *Q*, *E* and *W* as a function of malicious smart meters M for six values of E_m in three cases: $S_0 = 10$, $S_0 = 20$ and $S_0 = 50$, with $M_0 = S_0 - 2$ in the three cases.

For different E_m the optimal solutions behave differently. When E_m is small, the *SGN* can afford an increase in the number of spotter queries because T and G remain small. With an increase in S_0, the *MSM* tries to involve more smart meters into spot-checking, while the overhead constraint limits the number of spotter queries. The complex interaction between all *MSM* policy parameters makes intuitive choice of their optimal combination impossible and very complex. Thus, the suggested method is necessary to confront the collusive attacks in an efficient manner.

Figures 7–11 present respectively T_{opt}, G_{opt}, Q_{opt}, E_{opt} and W as a function of malicious smart meters *M* for $S_0 = 10$ with different values of E_m.

It can be seen that in five cases of E_m, T_{opt} decreases with the increase of *M*, Q_{opt} increases with the increase of *M* and G_{opt} remains always equal to 1 and *W* remains always zero, and we notice that in most cases we can reach a null *W* with an $E_{opt} < E_m$. on the other hand, if E_m is small, $E_m = 10$, we have a disturbance in the results and the increase of the G_{opt} values

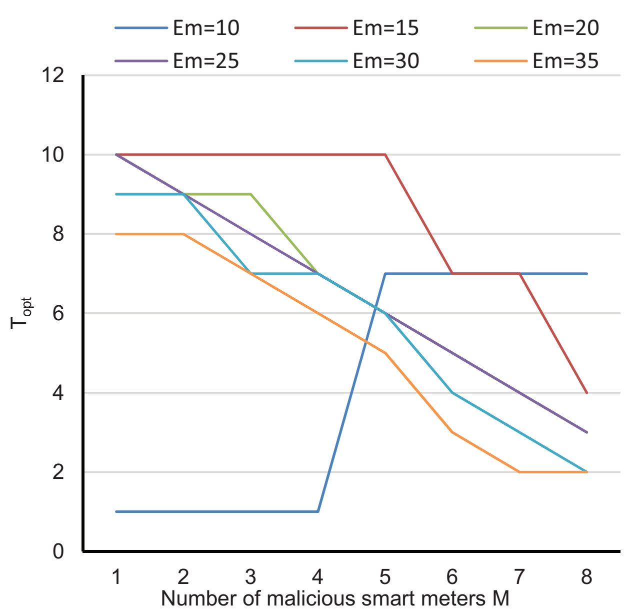

Figure 7 T_{opt} as a function of number of malicious smart meters for $S_0 = 10$ and different E_m.

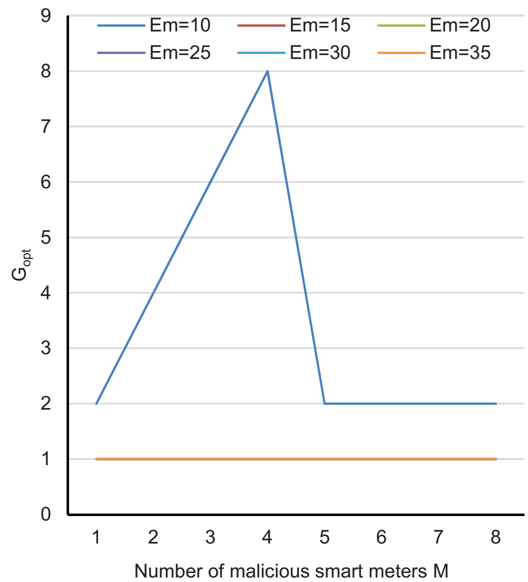

Figure 8 G_{opt} as a function of number of malicious smart meters for $S_0 = 10$ and different E_m.

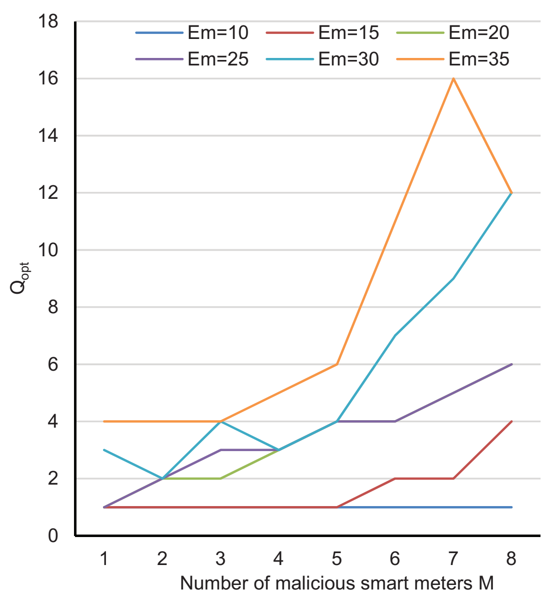

Figure 9 Q_{opt} as a function of number of malicious smart meters for $S_0 = 10$ and different E_m.

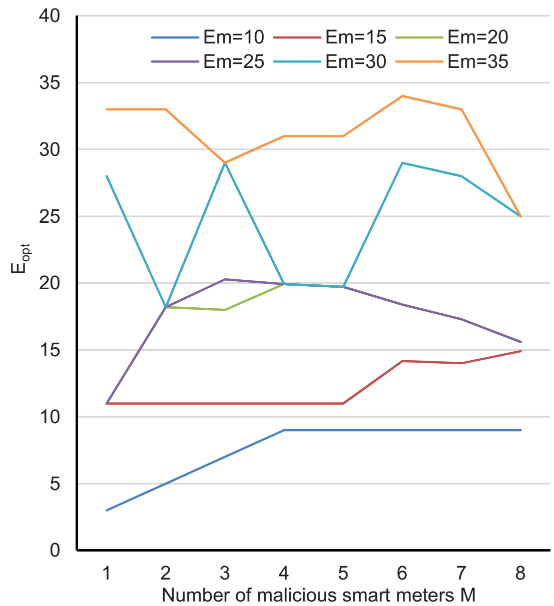

Figure 10 E_{opt} as a function of number of malicious smart meters for $S_0 = 10$ and different E_m.

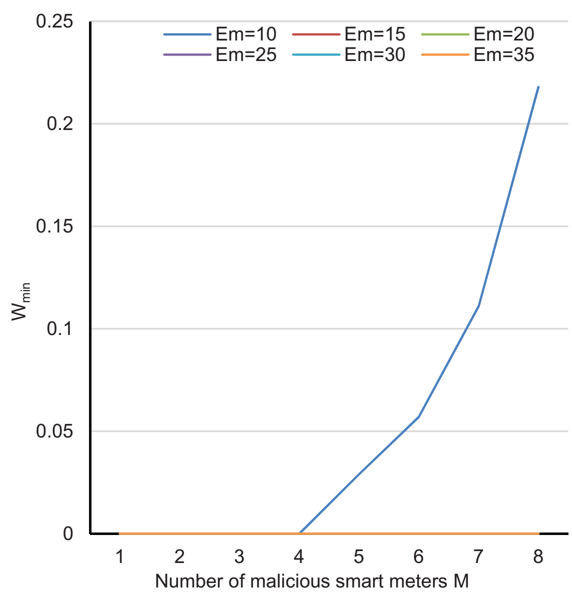

Figure 11 W as a function of number of malicious smart meters for $S_0 = 10$ and different E_m.

between $M = 4$ and $M = 8$ generates an increase in the values of *W* which reaches up to 0.218. However, the increase of the overhead can again lead to a considerable reduction of *W*, and an analysis of E_{opt} curve allows making the decision about the reasonable overhead level.

Figures 12–16 present respectively T_{opt}, G_{opt}, Q_{opt}, E_{opt} and *W* as a function of malicious smart meters *M* for $S_0 = 20$ with different values of E_m.

It can be seen that we can divide this case into two, case 1: $W = 0$ and case 2: $W > 0$:

Case 1: This case is valid for the values $E_m = 25$, $E_m = 30$ and $E_m = 35$, and T_{opt} decreases with the increase of *M*, Q_{opt} increases with the increase of *M* and G_{opt} is always equal to 1.

Case 2: This case is valid for the values $E_m = 10$, $E_m = 15$ and $E_m = 20$, T_{opt} increases with the increase of *M*, and we notice a disruption to the values of G_{opt} that generates an increase value of *W* which reaches $W = 0.7$.

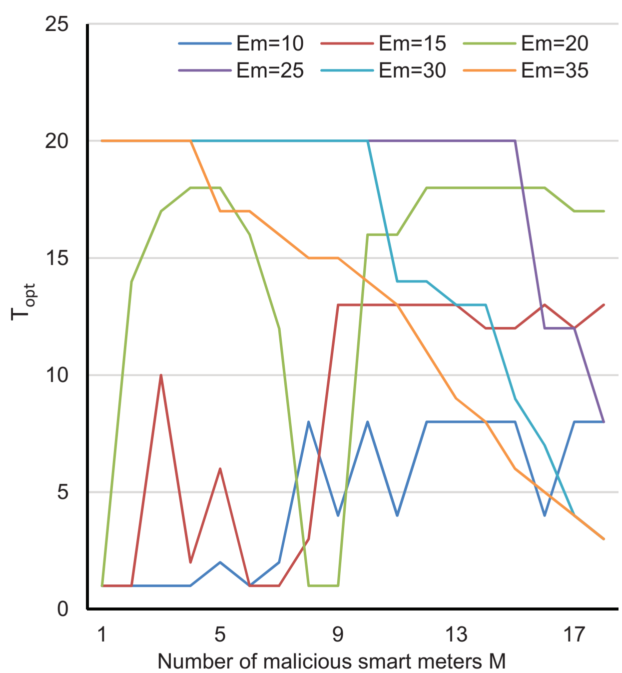

Figure 12 T_{opt} as a function of number of malicious smart meters for $S_0 = 20$ and different E_m.

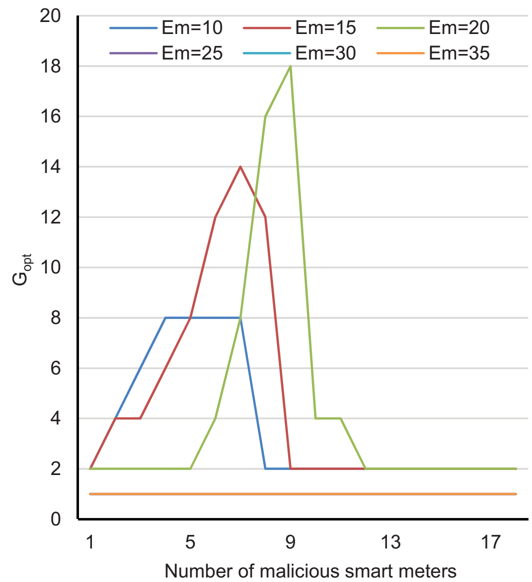

Figure 13 G_{opt} as a function of number of malicious smart meters for $S_0 = 20$ and different E_m.

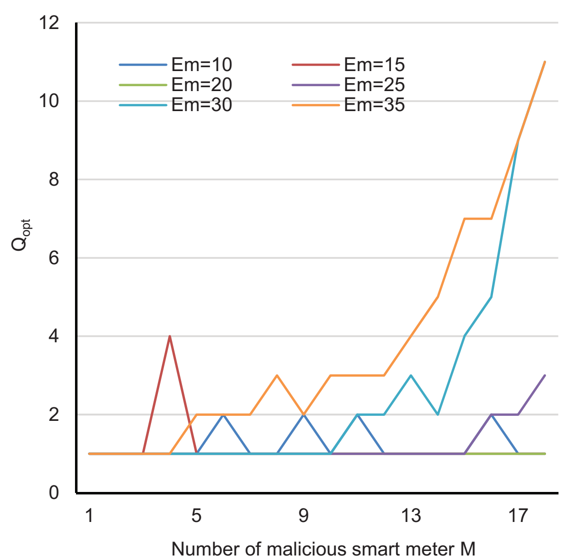

Figure 14 Q_{opt} as a function of number of malicious smart meters for $S_0 = 20$ and different E_m.

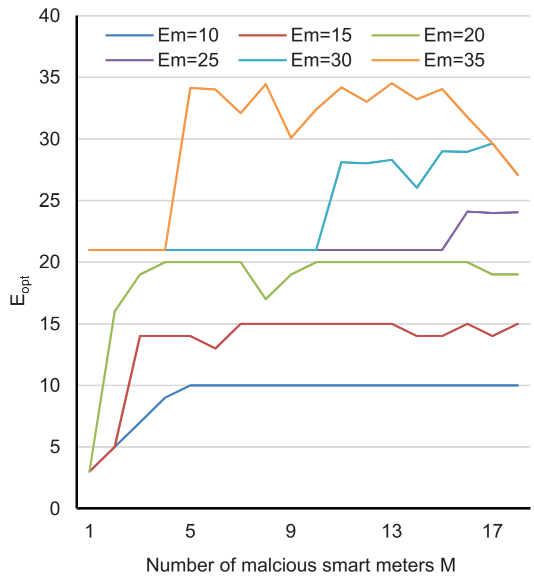

Figure 15 E_{opt} as a function of number of malicious smart meters for $S_0 = 20$ and different E_m.

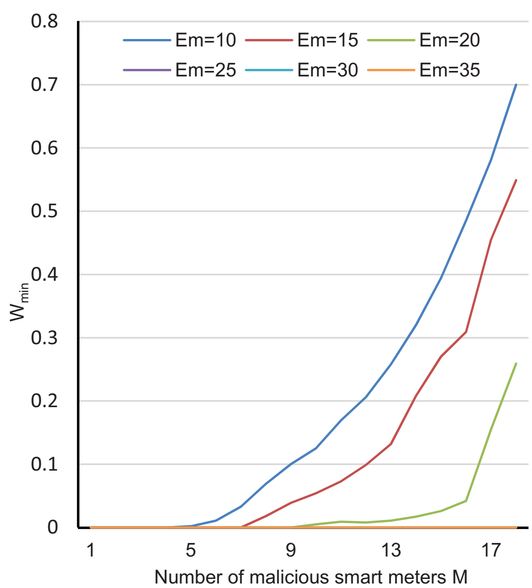

Figure 16 W_{min} as a function of number of malicious smart meters for $S_0 = 20$ and different E_m.

We notice in most cases we can reach a null *W* with $E_{opt} < E_m$, and the PAWD *W* is always decreasing because the probability of detecting *MSMs* by the spotter queries and removing *MSMs* from the list increases with *E*. On the other hand, it's clear that with an increase in the number of malicious smart meters, the *SGN* tends to increase the number of smart meters used for spotter queries and decrease the number of smart meters assigned to the genuine query.

Figures 17–21 present respectively T_{opt}, G_{opt}, Q_{opt}, E_{opt} and W as a function of malicious smart meters *M* for $S_0 = 50$ with different values of E_m.

In this case, when $S_0 = 50$, the number of *MSMs* can be very high. We can divide it into two cases: case 1: $1 \leq M < 25$ and $25 \leq M \leq 48$:

Case 1: $1 \leq M < 25$: in this case, we have a perturbation in the values of T_{opt} and Q_{opt}, G_{opt} and E_{opt} increases with the increase of *M*, but *W* remains almost zero.

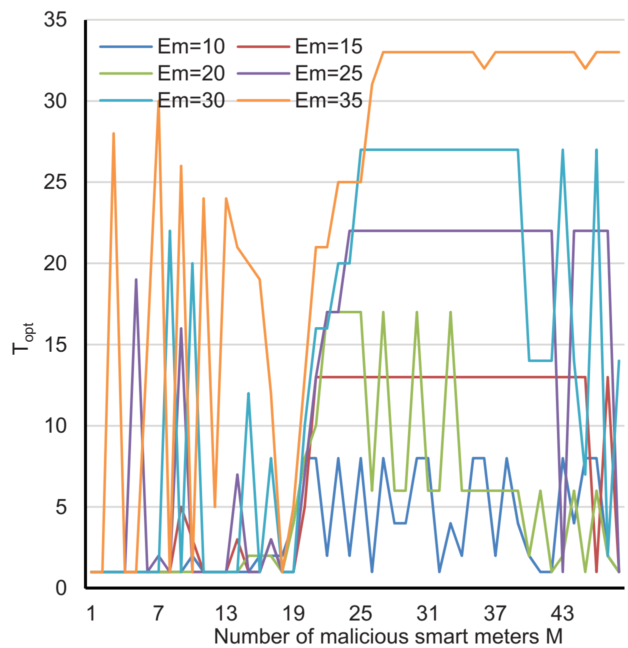

Figure 17 $\mathrm{T_{opt}}$ as a function of number of malicious smart meters for $\mathrm{S_0} = 50$ and different $\mathrm{E_m}$.

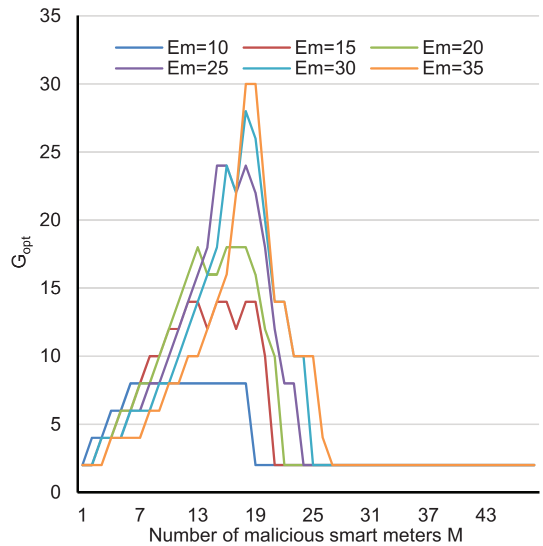

Figure 18 G_{opt} as a function of number of malicious smart meters for $S_0 = 50$ and different E_m.

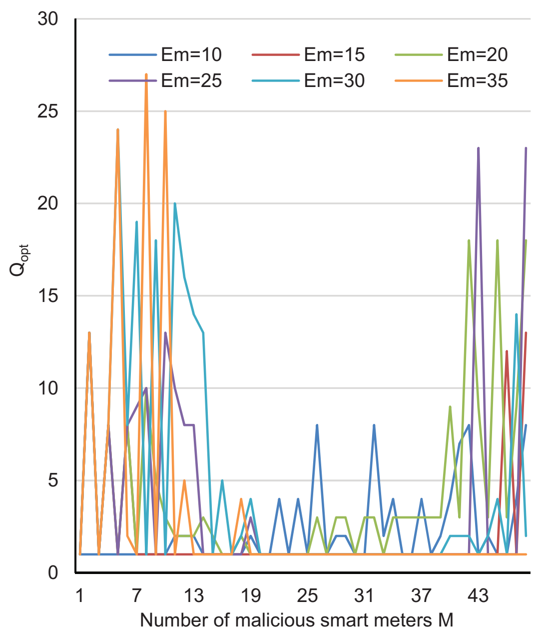

Figure 19 Q_{opt} as a function of number of malicious smart meters for $S_0 = 50$ and different E_m.

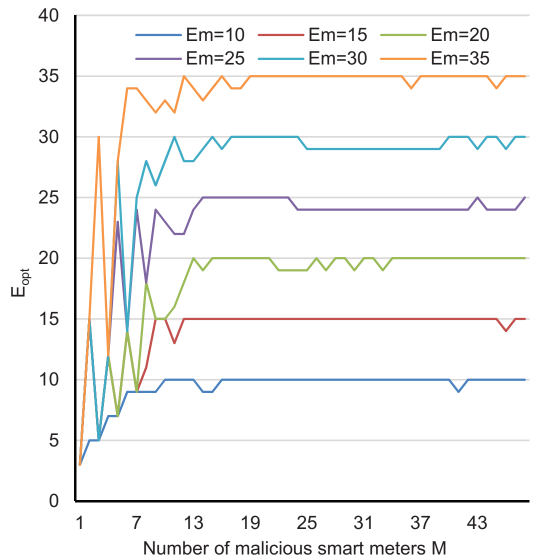

Figure 20 $\mathrm{E_{opt}}$ as a function of number of malicious smart meters for $\mathrm{S_0} = 50$ and different $\mathrm{E_m}$.

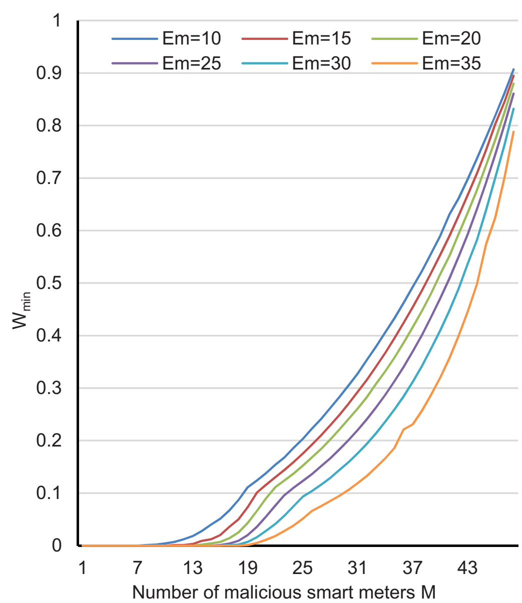

Figure 21 $\mathrm{W_{min}}$ as a function of number of malicious smart meters for $\mathrm{S_0} = 50$ and different $\mathrm{E_m}$.

Case 2: $25 \leq M \leq 48$: in this case, Q_{opt}, E_{opt} and *W* increases with the increase of *M*, G_{opt} decreases and is fixed in $G = 2$ with the increase of *M*, and Q_{opt} has varied values but with a low thickness.

The best strategy used by *SGN* is to try to detect all *MSMs* by sending spotter queries to as many smart meters as possible, reducing the number of smart meters allocated to the genuine query. However, the overhead *E* increases with the increase of *M*, and the *W* decreases with the increase of the overhead.

It's clear that with a reduction in M_0 the *PAWD* W decreases. Indeed, certain knowledge of M_0 allows *SGN* to choose optimal minimizing *PAWD* for this parameter.

7 Conclusion

Collusion detection and tolerance deserve great research interest from the various distributed computing systems such as smart grid. In this paper, we formulate and solve the spot-checking optimization problem to the collusive behaviour of the smart meters in smart grid.

A repetitive method is proposed, in this work, to measure and evaluate the probability of accepting wrong data (PAWD) and the expected overhead in terms of the total number of query assignments for our system. This method resolves the spot-checking optimization problem by finding the optimal combination of query send policy parameters (the number of deployed spotter queries, the number of smart meters tested by each spotter query, and the number of smart meters for performing the genuine query) minimizing the PAWD while satisfying the expected overhead parameter.

Finally, the obtained results can contribute in optimal distribution and assignment of all components of smart grid system for secure, reliable and cost-aware operation of this system.

References

[1] Y. Liu, P. Ning, and M. K. Reiter, False data injection attacks against state estimation in electric power grids, 16th ACM conference on Computer and communications security, 21–32, 2009.

[2] J. Hao, R. J. Piechocki, D. Kaleshi,, W. H. Chin and Z. Fan, Sparse Malicious False Data Injection Attacks and Defense Mechanisms in

Smart grids, IEEE Transactions on Industrial Informatics, 11(5), 1–12, 2015.

[3] L. F. G. Sarmenta, Sabotage-tolerance mechanisms for volunteer computing systems, Future Generation Computer Systems, 18(4), 561–572, 2002.

[4] M. Moca, G. C. Silaghi, and G. Fedak, Distributed results checking for MapReduce in volunteer computing, IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, 1847–1854, 2011.

[5] P. Domingues, B. Sousa, and L. M. Silva, Sabotage-tolerance and trust management in desktop grid computing, Future Generation Computer Systems, 23(7), 904–912, 2007.

[6] S. Choi and R. Buyya, Group-based adaptive result certification mechanism in Desktop Grids, Future Generation Computer Systems, 26(5), 776–786, 2010.

[7] K. Watanabe, N. Funabiki, T. Nakanishi and M. Fukushi, Optimal Spot-Checking for Delayed Attack on Desktop Grid Systems, 15th International Conference on Computer Modelling and Simulation, 600–605, 2013.

[8] A. C. Oliveira, L. M. R. Sampaio, S. Fernandes, and F. Brasileiro, Adaptive Sabotage-Tolerant Scheduling for Peer-to-Peer Grids, Fourth Latin-American Symposium on Dependable Computing, 25–32, 2009.

[9] S. Zhao, V. Lo, and C. GauthierDickey, Result verification and trust-based scheduling in peer-to-peer grids, Proc. of 5th IEEE Int. Conf. Peer-to-Peer Computing , Konstanz, 31–38, 2005.

[10] G. Silaghi, L. Silva, P. Domingues, A. E. Arenas, Tackling the Collusion Threat in P2P-enhanced Internet Desktop Grids, the CoreGRID Workshop on Programming Models Grid and P2P System Architecture Grid Systems, Tools and Environments, 393–402, 2007.

[11] Y. A. Zuev, On the estimation of efficiency of voting procedures, Theory Probability and Its Applications, 42(1), 73–81, 1998.

[12] J. D. Sonnek, A. Chandra, and J. Weissman, Adaptive reputation-based scheduling on Unreliable Distributed Infrastructures, IEEE Transaction Parallel Distributed Systems, 18(11), 1551–1564, 2007.

[13] G. C. Silaghi, F. Araujo, L. M. Silva, P. Domingues, and A. E. Arenas, Defeating colluding nodes in Desktop Grid computing platforms, Journal of Grid Computing, 7(4), 555–573, 2009.

[14] L. Canon, E. Jeannot, and J. Weissman, A Scheduling and Certification Algorithm for Defeating Collusion in Desktop Grids, 31st International

Conference on Distributed Computing Systems (ICDCS), 343–352, 2011.

[15] F. Araujo, J. Farinha, P. Domingues, G. C. Silaghi, and D. Kondo, A maximum independent set approach for collusion detection in voting pools, Journal of Parallel and Distributed Computing, 71(10), 1356–1366, 2011.

[16] A. Bendahmane, M. Essaaidi, A. El Moussaoui and A. Younes, The Effectiveness of Reputation-Based Voting for Collusion Tolerance in Large-Scale Grids, IEEE Transactions on Dependable and Secure Computing, 12(6), 665–674, 2015.

[17] Y.-S. Lee and T.-H. Chen, Insight into collusion attacks in random-grid-based visual secret sharing, Signal Processing, 92(3), 727–736, 2012.

[18] A. Estache, Emerging Issues in Competition, Collusion, and Regulation of Network Industries, Centre for Economic Policy Research, Published by London Publishing Partnership, 2011.

[19] A. Chakrabarti, Grid Computing Security, Infosys Technologies Limited, Springer-Verlag Berlin Heidelberg, 2007.

[20] F. Anta, C. Georgiou, M. A. Mosteiro and D. Pareja D, Algorithmic Mechanisms for Reliable Crowdsourcing Computation under Collusion, PLoS ONE, 10(3), e0116520, 2015.

[21] K. Watanabe, M. Fukushi and S. Horiguchi, Collusion-Resistant Sabotage-Tolerance Mechanisms for Volunteer Computing Systems, IEEE International Conference on e-Business Engineering (ICEBE'09), 213–218, 2009.

[22] L. C. Canon, E. Jeannot and J. Weissman, A dynamic approach for characterizing collusion in desktop grids, IEEE International Symposium on Parallel & Distributed Processing (IPDPS), 1–12, 2010.

[23] E. Staab and T. Engel, Collusion detection for Grid Computing, 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 412–419, 2009.

[24] El Yazid Dari and Mohamed Essaaidi, An Overview of Smart grid Cyber-Security State of The Art Study, 3rd International Renewable and Sustainable Energy Conference (IRSEC), 1–7, 2015.

Biographies

El Yazid Dari born in Nador, Morocco, in 1980. He received the “DESA” degree in Electrical Engineering from the university Abdelmalek Essaadi of Tetuan. Currently he is working toward the Ph.D. degree with the Information and Telecommunication Systems Group at Abdelmalek Essaadi University. His research interests include the smart grid security, Computer Sciences and Telecommunications.

Ahmed Bendahmane has received his Ph.D. degree in Computer Sciences from Abdelmalek Essaadi University at Information and Telecommunication Systems Laboratory, Faculty of Science, Tetuan, Morocco (2013). His main research interests include distributed systems, security of grid and cloud computing systems, Computer Networks, Intrusion Detection and Tolerance, and Multi-Agent Systems. Bendahmane has published a number of refereed research publications in this area.

Mohamed Essaaidi (SM'00) obtained the Doctorate of State degree in Electrical Engineering in 1997 from Abdelmalek Essaadi University in Tetuan, Morocco. He has been a Professor of Electrical and Computer Engineering in Abdelmalek Essaadi University since 1993 and the Deanof College of IT (ENSIAS) of Mohammed V University in Rabat, Morocco since 2011. He is the founder and the current Chair of the IEEE Morocco Section. He's been the co-founder and co-General Chair of several IEEE co-technically sponsored conferences such as Information and Communication technologies International Symposium (ICTIS) since 2005 and International Conference on Multimedia and Computing Systems (ICMCS) since 2009 and co-Director of NATO Advanced Research Workshop on Information Security Assurance, Tetuan, Morocco in June 3–6, 2005 and NATO ASI on Software Agents, Applications and Technologies, Tangiers, Morocco in September 2010. He is the author and co-author of more than 200 papers which appeared in refereed specialized journals and symposia. He was the co-editor of the book "Information Assurance and Computer Security", IOS Press, 2007 and the book "Software Agents, Agent Systems and their Applications", IOS Press, 2012. He was Editor-in-Chief of the "International Journal on Information and Communication Technologies", Serials Publications, India, from 2007 to 2010. Prof. Essaaidi is a Senior Member of IEEE and a member of IEEE Computer Society, IEEE Communication Society and IEEE Education Society.