A Witness-Based Approach For Data Fusion Assurance In Wireless Sensor Networks

Wenliang Du[†] , Jing Deng[†], Yunghsiang S. Han[§], Pramod K. Varshney[†]

***Abstract*— In wireless sensor networks, sensor nodes are spread randomly over the coverage area to collect information of interest. Data fusion is used to process these collected information before they are sent to the base station, the observer of the sensor network. We study the security of the data fusion process in this work. In particular, we propose a witness-based solution to assure the validation of the data sent from data fusion nodes to the base station. We also present the theoretical analysis for the overhead associated with the mechanism, which indicates that even in an extremely harsh environment the overhead is low for the proposed mechanism.**

***Index Terms*— wireless ad hoc networks, sensor networks, data fusion, information assurance, network security.**

I. Introduction

Recent advances in electronic and computer technologies have paved the way for the proliferation of ubiquitous wireless networks. Fast deployment of communication networks is highly desirable under many situations, such as establishing efficient, survivable dynamic communications for emergency and rescue operations. *Wireless ad hoc networks* can be formed almost instantly by deploying a number of nodes. The formation of ad hoc networks does not require the use of any pre-existing infrastructure. Applications of such networks range from battlefield communication networks to environment monitoring sensor networks.

A special type of ad hoc networks is a *wireless sensor network*. Contrary to more traditional computer networks, wireless sensor networks (WSN) consist of a large number of ultra-small autonomous devices. Each device, called a sensor node, is battery powered and equipped with integrated sensors, data processing capabilities, and short-range radio communications. In typical application scenarios, the nodes are spread randomly over the terrain under scrutiny and collect sensor data. Each node processes the data and coordinates with nearby nodes to combine their information (the process is called *data fusion*). The aggregate data is then forwarded to specialized gateway nodes or base stations. Examples of WSN projects include SmartDust [1] and WINS [2].

One of the main goals of WSN is to detect events of interest in a distributed manner. Due to the large number of sensors in WSN and limited communication range, data fusion is employed in order to reduce the traffic load from all the sensors to the base station. Even though this data collection and processing architecture drastically relieves the communication burden on the network, the nodes conducting data fusion are now vulnerable to malicious attacks. If a data fusion node is compromised, it can send bogus data to the base station; the base station has no chance to discover it since the data collected by sensors is not sent to the base station directly and the security association between sensors and the base station cannot be invoked. In this paper we propose to use the "witness" concept to solve the assurance problem between data fusion nodes and the base station. Some nodes around the data fusion node are selected as witnesses to monitor the data fusion results. The proposed method assures the validity of data fusion results received by the base station.

The rest of this paper is organized as follows. Section II briefly introduces the concept of data fusion and addresses the assurance of data fusion nodes in WSN. Section III presents a witness-based solution for the validation of the data sent from data fusion nodes to the base station. The overhead of the added security mechanism is studied in Section IV. Our concluding remarks are presented in Section V.

II. Data Fusion Assurance Problems

In order to avoid heavy traffic and conserve energy in a sensor network caused by the transmission of raw data back to the base station from each sensor, data fusion nodes can be deployed in the network. In the data fusion process, a data fusion node receives data from a number of sensors, conducts data fusion, and then sends the result (decision) to the base station. One example of such a system is distributed detection using multisensor networks as described by Varshney in [3] and further discussed in [4], [5], [6]. A general block diagram of this application with one data fusion node is given in Fig. 1. The sensor nodes $S_1, S_2, \ldots, S_n$ collect data $x_1, x_2, \ldots, x_n$ from the environment and make their binary decisions $b_1, b_2, \ldots, b_n$ based on some detection rules. Then they send these decisions to the data fusion node. The fusion node decides on the presence or absence of the event in that environment, based on the binary data it received, and then sends this result u_0 to the base station. One of the key advantages of this distributed detection and fusion scheme is that it reduces the transmission burden between sensor nodes and the data fusion node.

While much effort has gone into the design of fusion algorithms [3], to our knowledge, security and assurance aspects

This work is supported in part by the SUPRIA program of the CASE Center at Syracuse University, by the National Science Foundation under Grant No. 0219560, and by the National Science Council of Taiwan, R.O.C., under grants NSC 90-2213-E-260-007 and NSC 91-2213-E-260-021

[†]Dept. of Electrical Engineering and Computer Science at Syracuse University, E-mail: wedu, jdeng01, varshney@ecs.syr.edu; Fax:+1-315-443-1122.

[§]Dept. of Computer Science and Information Eng., National Chi Nan Univ., Taiwan, R.O.C. Han's work was completed during his visit to the CASE Center and Dept. of Electrical Engineering and Computer Science at Syracuse University, USA. (E-mail: yshan@csie.ncnu.edu.tw; Fax:+886-49-291-5226).

of data fusion systems have not been studied. The current data fusion system puts a great deal of trust on the nodes conducting data fusion. However, if the data fusion node is compromised and becomes malicious, it can send an arbitrary fusion result to the base station. Since the original data are not forwarded to the base station, it is difficult for the base station to verify whether the result is valid[1]. Moreover, sensors might also be compromised. If a sensor is compromised and becomes malicious, it can send incorrect sensing results to the fusion node. However, because some fusion algorithms can tolerate certain number of malicious sensors, we will assume that the number of compromised sensor nodes is tolerable.

The goal of this paper is to develop an approach to guarantee that only valid data fusion results are accepted by the base station, and invalid fusion results are rejected by the base station.

Problem 2.1: **(Data fusion node assurance)** Under the assumption that the data from the sensors can be trusted, how can the base station verify that the fusion result is valid? We also assume that each sensor shares a different secret key with the base station.

III. Witness-Based Data Fusion Node Assurance

We propose to use *witness nodes* to enhance the assurance of data fusion. In order to prove the validity of the fusion result, the fusion node has to provide proofs from several witnesses. A witness is one who also conducts data fusion like a data fusion node, but does not forward its result to the base station; instead, each witness computes the *Message Authentication Code* (*MAC*) of the result (we call the *MAC* a proof), and then provides it to the data fusion node, who must forward the proofs to the base station. If the data fusion node is compromised, and wants to send an invalid fusion result to the base station, it has to forge the proofs on the invalid result.

There could be various ways to achieve what we described above. We assume that the data fusion node and witness nodes share a secret key with the base station. Let F denote the data fusion node. Assume that we have chosen m witnesses, $w_1, \ldots, w_m$, and $k_1, \ldots, k_m$ represent the *MAC* keys they share with the base station. Without loss of generality, assume that m is even. After receiving the data from the sensor nodes, each witness i conducts data fusion, and obtains the result s_i; it then sends $MAC_i = MAC(s_i, w_i, k_i)$ to the data fusion node. We propose to use the n out of $m+1$ voting scheme to determine the validation of the fusion result. In the n out of $m+1$ (m witness nodes and one data fusion node being validated) voting scheme in our assurance system, when at least $n-1$ out of the m witnesses agree with the fusion result obtained at the fusion node, the base station accepts the fusion result to be valid; otherwise it discards the fusion result. Base station might poll one of the witnesses to send the result to it after it discards the received result. This action of the base station will be discussed and analyzed in Section IV. We present the proposed method starting with a special case of the general voting scheme.

[1]Here valid means that the result is the correct computation result based on the data received by the fusion node.

a) $m+1$ out of $m+1$ voting scheme or the AND rule [3]: After receiving $(MAC_1, \ldots, MAC_m)$ from the witnesses, the data fusion node computes

$$MAC_F = MAC(S_F, F, k_F, MAC_1 \oplus \cdots \oplus MAC_m)$$

where S_F is the data fusion result computed by node F, k_F is the *MAC* key shared by the base station and the node F, and $\oplus$ represents the XOR operation. F then sends (S_F, F, w_1, …, w_m, MAC_F) to the base station. After receiving the packet, the base station computes $MAC_i = MAC(S_F, w_i, k_i)$ for each w_i, and then computes $MAC'_F = MAC(S_F, F, k_F, MAC_1 \oplus \ldots \oplus MAC_m)$. If $MAC'_F = MAC_F$, the data fusion result is declared to be valid because the result is supported by all the witnesses. Since the witness identification information $w_1, \ldots, w_m$ is the same for all the messages sent to the base station by a data fusion node, it can be sent just once initially unless the witnesses are changed. Therefore, the advantage of this scheme is that the overhead caused by the MAC and the voting scheme is fixed regardless of how many witnesses are used.

b) n out of $m+1$ voting scheme $(1 \leq n \leq m+1)$: A disadvantage of the previous method is that even if one of the witnesses is malicious, and always sends an invalid *MAC* to the data fusion node, it basically achieves a successful denial of service attack. This is because one invalid *MAC* causes MAC_F to become invalid. The base station has no way to distinguish whether it is caused by the invalid S_F or by some invalid MAC.

To make the proposed scheme more robust against the above attack, node F should not merge all the MAC_i's; instead, it needs to forward all the MAC_i's. Therefore, node F sends $R = (S_F$, F, MAC_F, w_1, MAC_1, …, w_m, $MAC_m)$ to the base station (Once again, w_1, …, w_m need to be sent only once at the beginning unless the witnesses are changed). If at least n out of $m+1$ MAC's match, the result S_F is accepted; otherwise, the result is dropped because it is not endorsed by at least $n-1$ of the witnesses. Compared to the previous method, the overhead of this method is higher because multiple MAC's need to be sent to the base station.

Let $2^{-\delta}$ be the tolerance probability that an application accepts an invalid result. In a straightforward implementation, one may choose MACs with length δ and the size of R (the data sent to the base station) is proportional to δm. However, as we analyze in the following, to achieve the same tolerance probability, we can reduce the size of MACs when the number of witnesses increases. In particular, the increased overhead becomes at most $2(\delta + m)$ when the majority voting scheme is employed. An easy way to reduce the size of MACs is to use only a portion of a fixed-sized MACs, or use flexible data encryption algorithms, such as RC6 [7].

Let us assume that k bits are used for each *MAC*, and the number of witnesses is m. We determine the value of k according to the security requirement next. Let us assume that the malicious data fusion node F decides to send an invalid fusion result S'_F to the base station, and still wants to get an endorsement from $n-1$ of the witnesses. We also assume that none of the witnesses collude with F. In order to prove the validity of S'_F to the base station, F needs to forge at least $n-1$ valid endorsements. Because F does not know the *MAC* keys between the base station and the nodes whose endorsements are going to be forged, F can only randomly pick a number as the endorsement[2]. The probability for F to guess the endorsement from a witness correctly is $p = \frac{1}{2^k}$. If F uses this method to forge the endorsement, the probability that at least $n-1$ of the endorsements happen to be correct is:

$$Ps = \sum_{i=n-1}^{m} \binom{m}{i} p^i (1-p)^{m-i}. \tag{1}$$

The "tail" of a binomial distribution can be estimated by the following bound [8]:

$$\begin{aligned} &\sum_{i=\lambda m}^{m} \binom{m}{i} p^i (1-p)^{m-i} \\ \leq\ & \lambda^{-\lambda m}(1-\lambda)^{-(1-\lambda)m} p^{\lambda m}(1-p)^{(1-\lambda)m} \end{aligned}$$

given $\lambda > p$.

Thus,

$$\begin{aligned} Ps \ \leq\ & m^m (n-1)^{-(n-1)}(m-n+1)^{-(m-n+1)} \\ & \cdot p^{n-1}(1-p)^{m-n+1}. \end{aligned}$$

When the majority voting rule is applied, i.e., $n = \frac{m}{2}+1$, the above upper bound becomes

$$Ps = \sum_{i=\frac{m}{2}}^{m} \binom{m}{i} p^i (1-p)^{m-i} \leq 2^m p^{\frac{m}{2}} = 2^{-m(\frac{k}{2}-1)}.$$

In this case, when $2^{-\delta}$ is the tolerance probability that an application accepts an invalid result, the relationship between k, m, and δ is

$$m(\frac{k}{2}-1) \geq \delta. \tag{2}$$

The overhead defined as the total number of extra bits for all m MAC's is $mk \geq 2(\delta + m)$. As an example, let us calculate the overhead for the n out of $m+1$ scheme when the tolerance probability of an application is 2^{-10} ($\delta = 10$), namely, the probability of accepting an invalid result is at most 1 out of 1024. When $m = 4$ witnesses are used, k should be at least 7 according to Eq. (2), and hence the overhead of our scheme is $mk \geq 28$ bits. If the system has $m = 6$ witnesses, we have $k \geq 6$, with an overhead of $mk \geq 36$ bits. Note that one can always choose the smallest k that satisfies Inequality 2 in order to save computation and transmission loading.

Fig. 2 depicts the relationship between k, m, and δ when the majority voting rule is applied. From the figure, we can observe that, for a fixed number of witnesses (m), the tolerance probability as well as its upper bound decrease exponentially as the number of bits per MAC (k) increases.

Since $p = \frac{1}{2^k}$, according to Eq. (1), we can see that for a fixed m, when n increases, to achieve a fixed Ps, the key length k required for each witness to achieve the system tolerance decreases. Even though it reduces the overhead introduced by the security mechanism, as indicated by the previous discussion, the ability to resist a denial of service attack also diminishes. Therefore, the number n should be a reasonable compromise between the ability to protect the system from denial of service attacks and associated overhead. When n is specified, the minimum key length k for each witness to achieve the system tolerance can be determined.

IV. Polling Scheme and Associated Overhead

As indicated previously, when the base station does not receive a valid result from the current data fusion node, it polls one of its m witnesses to obtain a valid result. Obviously, this polling is an overhead of the system. We develop a polling mechanism and analyze its overhead in this section. In the first subsection, we determine the number of polling messages from the base station given probability of nodes being compromised. Then, we determine the number of expected polling messages given that ℓ nodes are compromised.

A. Overhead Based on the Probability of Node Being Compromised

In the n out of $m+1$ scheme, where $m+1 \geq n$, there are two scenarios in which the base station may not receive a valid result:

- Current data fusion node compromised but enough honest (uncompromised) witnesses;
- Not enough honest nodes (data fusion node or witnesses).

In the first scenario, in which the data fusion node is compromised or malicious, it may send an arbitrary fusion result together with the MAC messages to the base station, which rejects the result without enough endorsements from the witnesses. The base station then polls other witnesses and activates one of them as the new data fusion node and obtains its data fusion result.

In the second scenario, in which the number of compromised or malicious nodes (including the current data fusion node and witnesses) is larger than $m-n+1$, there are at most $n-1$ honest nodes in the system. So it is impossible for the base station to receive a valid result. Under such circumstance, the base station may still poll other witness nodes and activate one of them as the new data fusion node.

The difference between these scenarios is that, with the polling scheme, the base station will be able to obtain a valid result in the first scenario but not in the second scenario.

A simple round robin polling scheme in which the base station polls the rest of the un-polled nodes randomly is the best polling strategy. This can be justified by the following facts: First, in the first scenario, since the malicious data fusion node might modify any portion of the received result, there is no useful information for the base station to help it decide the next node to poll; second, in the second scenario, no matter what the strategy is, the base station should have polled all the nodes except the remaining $n-1$ to realize that no valid result can be obtained[3]. Hence, randomly choosing the next node to be polled is the best strategy.

We calculate the expected number of polling messages (or fusion result transmissions from either the data fusion node or

[2]We assume that the *MAC* algorithm is secure, i.e., F can not derive the key shared between a witness and the base station by looking at the *MAC* generated by the witness and the fusion result.

[3]When there are $n-1$ unpolled witnesses left, and the result is invalid, the base station understands that at least one node out of n nodes is compromised or malicious. It is impossible to obtain n valid endorsements.

activated witnesses to the base station) in this section. Since the base station may be far away from the data fusion node and witnesses, such operations are expensive in terms of delay and energy consumption. In our calculation, the unit of overhead is defined as the number of polling messages or data fusion result transmissions.

We assume that each node, including the data fusion node and all the witnesses, might be compromised or malicious with an i.i.d. probability p_c. We define $\mathcal{H}_j^i$ as the event that out of j nodes there are at least i honest nodes. We further define P_j^i as

$$P_j^i \;=\; \text{Prob}\{\mathcal{H}_j^i\} = \sum_{k=0}^{j-i} \binom{j}{k} p_c^k (1-p_c)^{j-k}$$

for $0 \le i \le j$, 0 otherwise.

Our goal is to calculate $T(m+1,n)$, the expected number of node to base station transmissions, i.e., the number of time steps that it takes for the base station to complete its task. When the current data fusion node is honest, there are two possibilities: enough honest witnesses and not enough honest witnesses. In the first scenario, it takes one step to accept the valid result. In the second scenario, it takes $m+1-n+1$ steps to find out that it is impossible to obtain a valid result. The number of time steps in this case is given by

$$\begin{aligned} T_1 &= 1\cdot(1-p_c)\cdot P_m^{n-1} + \\ &(m+1-n+1)\cdot(1-p_c)\cdot(1-P_m^{n-1}) \end{aligned} \quad (3)$$

If the current data fusion node is compromised or malicious, it will take $1+T(m,n)$ steps for the base station to accept a valid result or stop polling and the number of time steps in this case is

$$T_2 = [1+T(m,n)]\cdot p_c \quad (4)$$

Adding Eq. (3) and (4), we have,

$$T(m+1,n) = p_c\cdot T(m,n) + 1 \quad (5)$$
$$+ \quad (m-n+1)(1-p_c)(1-P_m^{n-1}). \quad (6)$$

The boundary condition is

$$T(j,i) = \begin{cases} 1 & 0 < i = j \\ 0 & j < i, \end{cases}$$

which satisfies Eq. (6).

In order to derive the closed form of $T(m+1,n)$, we define

$$f(m,n) = 1 + (m-n+1)(1-p_c)(1-P_m^{n-1}). \quad (7)$$

Eq. (6) becomes

$$T(k+1,n) = p_c\cdot T(k,n) + f(k,n) \qquad n \le k \le m. \quad (8)$$

We multiply Eq. (8) by p_c^{m-k-1} for all values of k that are in the range $[n,m]$:

$$\begin{aligned} &p_c^{m-(k+1)} T(k+1,n) \\ =\; &p_c^{m-k} T(k,n) + p_c^{m-k-1} f(k,n). \end{aligned} \quad (9)$$

Summing up Eq. (9) for all k in the range $[n,m]$, we have:

$$\begin{aligned} &p_c^{m-(m+1)} T(m+1,n) \\ =\; &p_c^{m-n} T(n,n) + \sum_{k=n}^{m} p_c^{m-k-1} f(k,n). \end{aligned}$$

Thus, we have derived the closed form of $T(m+1,n)$,

$$T(m+1,n) = p_c^{m-n+1} + \sum_{k=n}^{m} p_c^{m-k} f(k,n). \quad (10)$$

We have numerically calculated $T(m+1,n)$ for different values of m and n. In Fig. 3, we present the overhead in terms of expected number of polling messages from the base station based on a fixed probability of a node being compromised $p_c = 0.5$ [4]. From the figure, we can see that $T(m+1,n)$ increases as n increases for a fixed m, due to the number of affirmative endorsements needed. After a certain point, $T(m+1,n)$ decreases because the maximum number of polls is bounded by $m-n+2$. It is also observed that the expected number of polling messages is about 7 when there are 20 witnesses in the system, corresponding to $n = 13$.

B. *Overhead Based on Fixed Number of Compromised Nodes*

It is obvious that when there are compromised nodes in the network the expected number of polling messages will be greater than one. It is important to know the expected number of polling messages when there are ℓ compromised nodes.

When $\ell \ge m-n+2$, the base station takes $m-n+2$ steps to find out that there is no valid result. In this case, the expected number of polling messages is obviously $m-n+2$.

When $\ell < m-n+2$, the expected number of polling messages can be derived as follows:

Let $\mathcal{A}(j)$ be the event that all the nodes polled by the base station before jth poll are compromised and the jth polled node is not:

$$\text{Prob}\{\mathcal{A}(j)\} = (m-\ell+1)\cdot\frac{\ell!(m-j+1)!}{(\ell-j+1)!(m+1)!}, \quad (11)$$

where $1 \le j \le \ell+1$.

Then the expected value of the number of polling messages when $\ell < m-n+2$ is

$$T_3(m+1,n,\ell) \;=\; \sum_{j=1}^{\ell+1} j\cdot\text{Prob}\{\mathcal{A}(j)\}.$$

In summary, $T_3(m+1,n,\ell)$ is

$$\begin{cases} (m-\ell+1)\displaystyle\sum_{j=1}^{\ell+1} j\frac{\ell!(m-j+1)!}{(\ell-j+1)!(m+1)!} & 1 \le \ell \le m-n+1 \\ m-n+2 & \text{otherwise.} \end{cases}$$

We have numerically calculated $T_3(m+1,n,\ell)$ for n and ℓ for m equal to 10 and 20. The results are shown in Fig. 4. From this figure ($m = 20$), we may observe that even for $\ell = 10$ and $n = 11$, which is the worst case for the majority voting scheme while the base station can still get a valid result, the expected number of polling messages, 1.83, is only slightly greater than one. Furthermore, for any ℓ and n in this figure, the maximum of the expected number of polling messages is 11 that is only about half of the overall number of nodes.

[4]We are more interested in the performance of our scheme in a harsh environment, with a large p_c.

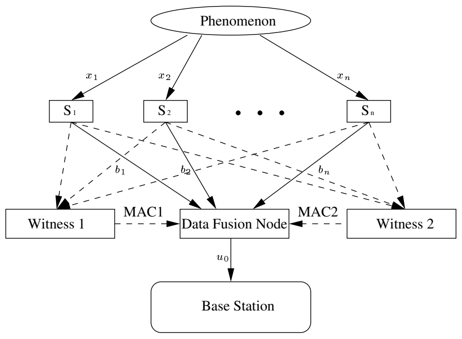

Fig. 1. Block diagram of a WSN for distributed detection using one data fusion node and two witnesses.

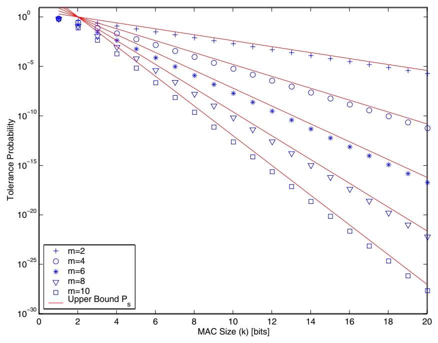

Fig. 2. Comparison between the tolerance probability and its upper bound.

V. Conclusions

We have studied the data fusion assurance problem in this paper. To ensure the validity of the data fusion result, we have developed a witness-based mechanism. The base station uses the n-out-of-$(m + 1)$ voting strategy to decide whether to accept a fusion result or not. Similar voting strategy is also proposed in [9], but our security mechanism is more efficient. To reduce energy consumption in our scheme, we have analyzed and computed the minimum length needed for the Message Authentication Code to achieve a pre-defined level of security. Our results show that the number of bits used for MACs does not increase linearly with the number of witnesses. We have also proposed a polling scheme and studied the overhead of our scheme when the invalid data from fusion nodes are rejected. Our results show that the expected number of polling messages of our scheme is much less than the number of witnesses even in a harsh environment.

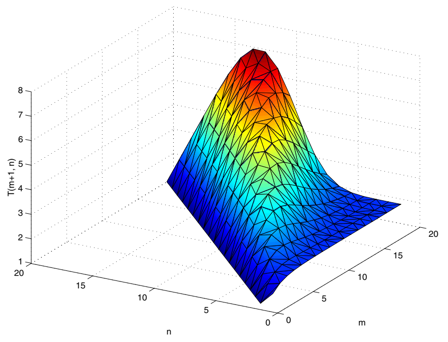

Fig. 3. Polling overhead when $p_c = 0.5$.

References

[1] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In *Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 483-92, 1999.

[2] Wireless Integrated Network Sensors, University of California, Available: http://www.janet.ucla.edu/WINS.

[3] P. K. Varshney, *Distributed Detection and Data Fusion*, New York, NY: Springer-Verlag, Inc., 1997.

[4] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors II. Advanced topics," *Proceedings of the IEEE*, Vol. 85, pp. 64-79, January 1997.

[5] P. K. Varshney, ed., Special Issue on Data Fusion, *Proceedings of the IEEE*, vol. 85, January 1997.

[6] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors Part I. Fundamentals," *Proceedings of the IEEE*, Vol. 85, pp. 54-63, January 1997.

[7] R. L. Rivest, M. J. B. Robshaw, R. Ridney, and Y. L. Yin. The RC6 Block Cipher. AES submission, June 1998. http://theory.lcs.mit.edu/ rivest/rc6.pdf.

[8] W. W. Peterson and E. J. Weldon, Jr., Error-Correcting Codes, 2$^{\text{nd}}$, Cambridge, MA: The MIT Press, 1972.

[9] B. Hardekopf, K. Kwiat, and S. Upadhyaya, "Secure and Fault-Tolerant Voting in Distributed Systems," *IEEE proc. Aerospace Conf., Vol. 3, pp. 1117-26, 2001.*

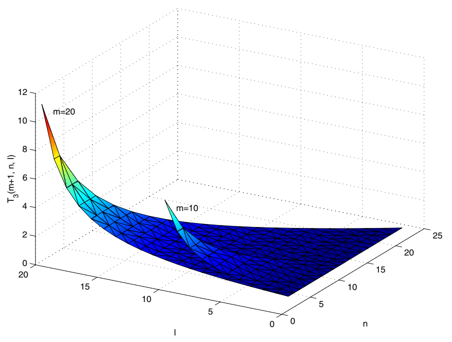

Fig. 4. Overhead for fixed number of compromised nodes ($m = 10, 20$ and $0 \leq \ell \leq (m - n + 1)$).