In-Network Aggregationin Wireless Sensor Networks

Mr. Harish G N
M.Tech. Student,
Department of CSE
AMC Engineering College,
Bangalore, India

Mr. Rabindranath S
Associate Professor,
Department of CSE,
AMC Engineering College,
Bangalore, India

***Abstract*—Wireless Sensor Networks(WSNs) is a collection of nodes organized into a cooperative network, In-network aggregation is usually required in many sensor applications to obtain the temporal variation information of aggregates. However, in a hostile environment, the adversary could fabricate false temporal variation patterns of the aggregates by manipulating a series of aggregation results through compromised nodes. In this paper, we identify distinct design issues for secure in-network aggregation in WSNs. An efficient verification scheme is proposed to protect the authenticity of the temporal variation patterns in the aggregation results. Compared with the existing secure aggregation schemes, our scheme only need to check a small portion of aggregation results in a time window and, thus, greatly reduces the verification cost. We define representative points and propose corresponding algorithms for representative point selection. By exploiting the spatial correlation among the sensor readings in close proximity, a series of security mechanisms are also proposed to protect the sampling procedure.**

Keywords—Wireless Sensor Networks; Continuous aggregation; authenticity; temporal variation patterns; spatial correlatio;

I. INTRODUCTION

A wireless sensor network is a collection of nodes organized into a cooperative network [10]. Each nodeconsists of processing capability (one or more microcontrollers, CPUs or DSP chips), may contain multipletypes of memory (program, data and flash memories), have a RF transceiver, have a power source (e.g., batteries and solar cells), and accommodate various sensorsand actuators.Wireless sensor networks (WSNs) are commonly used in pervasive and ubiquitous applications. WSNs are developed using both static (motes) and mobile (e.g. smart phone) sensor nodes for various applications such as smart homes, telehealth, surveillance, metering, and industry automation.

Data Aggregators can be called as organizers involved incompiling information from detailed database on individualsand selling information to others. For online purpose wheredynamic data is of prime importance, data aggregators cangather the information from designated websites and providingthe data to the user. The process of extracting raw statisticalinformation from the database or data repository, puttingit all together to produce statistical output that can be usedby the user and has relevance to statistical query it seeks to satisfy. Absolute difference in the value of data item at thedata source and the value known to the client.

In applications of wireless sensor networks (WSNs), the aggregations of sensed data, such as sum, average, and predicate count, is very important for the users to get summarization information about the monitored area. Instead of collecting all sensor data and computing aggregation results at the base station (BS), in network aggregation allows sensor readings to be aggregated by intermediate nodes, which efficiently reduces the communication overhead. Many in-network aggregation schemes have been proposed. However, since WSNs are often deployed in an open and unattended environment, an adversary could undetectably take control of one or more sensor nodes and subvert correct in-network aggregations by manipulating the partial aggregation results or reporting arbitrary readings through compromised nodes.

In this we consider the security of in-network aggregation in WSNs.In many WSN applications for environment monitoring, the users often need the temporal variation information in a series of aggregation results rather than an individual aggregation result. Thus, in-network aggregation of sensed data is usually desired. For ain-network aggregation query, a time interval, called epoch, is specified and the aggregation is evaluated in every epoch. The duration of every epoch specifies the amount of time sensor nodes wait before acquiring and transmitting each successive sample. In-network aggregation is not merely for one-shot responses to sporadic queries. It helps the users to understand how the environment changes over time and track real-time measurements for trend analysis.

A number of secure aggregation schemes have beenproposed [8], [9], [11]. SIA [8] addresses secureaggregation within the single aggregator network topology.A number of hierarchical secure aggregation schemes [9],[11] are proposed for aggregation in tree networktopology in which each node computes an intermediateaggregation result accounting for all sensing data of nodesin the sub-tree rooted at it. All these schemes aim to protect asingle aggregation computation. Directly using these schemes in ain-network aggregation results in individualverification for every aggregation result in every epoch,which will incur a great communication cost especiallyfor in-network aggregation having a long period or highfrequency (i.e., small epoch). The additional communicationcaused by interactive procedures between the base stationand sensor nodes for verification in every epoch also has anegative impact on the efficiency of transmission scheduling for ain-network data aggregation [12]. Besides, these schemes [8], [9] also are tightly coupled with the treetopology and, thus, unable to work with various other in-networkaggregation protocols [6], [7].

A number of hierarchical secure schemes [9], [11], [14], [16] have been proposed for in-network aggregation on tree topology, where each node computes an intermediate aggregation result accounting for the sensor readings of nodes in the subtree rooted at it. Hu and Evans [14] propose a secure aggregation scheme against one single malicious node in the network, in which each node checks the inconsistency of MACs from their children and grandchildren. Garofalakis et al. [16] propose to combine cryptographic signatures and Flajolet-Martin sketch [18] to achieve verifiable count aggregation.

Several secure hierarchical aggregation schemes [9], [11] follow an aggregation-commitment-attest framework.During the in-network aggregation, each node computesthe hash as commitment over the input of its aggregationcomputation, intermediate results, and data commitmentsfrom its children, and then sends the hash to its parent.Based on the commitments, interactive attest is performedbetween the base station and sensor nodes when aggregationcompletes. Yang et al. [9] propose a secure hop-by-hopdata aggregation protocol SDAP. The tree topology ispartitioned into multiple logical sub-tree groups, and sensordata are aggregated in every sub-tree separately to reducethe trust on high-level nodes. The groups returning outlierresults are attested by checking the aggregation correctnessalong a random path.

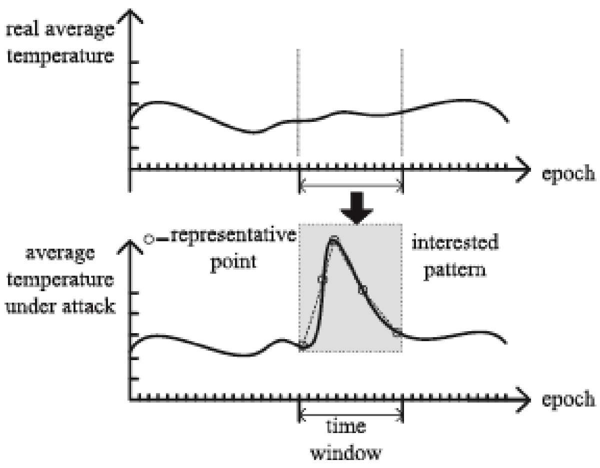


Fig. 1: The fabrication of the temporal variation pattern in ain-network aggregation.

Roy et al. [17] propose a scheme to verify the histogram computation to securely estimate the median. All these previous works address secure in-network aggregation within a snapshot query, so their approaches conduct verification for each single aggregation result. Unlike them, our work focuses on in-network aggregation and aims to protect the temporal variation patterns of aggregation results. To protect in-network aggregation, previous approaches would conduct individual verification in every epoch and, thus, can incur a significant communication cost. In contrast, our approach only selectively verifies a small part of aggregation results in a time window.

In this paper, we present an efficient scheme to detect false temporal variation patterns in ain-network aggregation. Our scheme verifies the correctness of the observed temporal variation pattern in a time window by checking only a small part of aggregation results termed representative points. The representative points are selected to capture the temporal variation pattern of the aggregate.

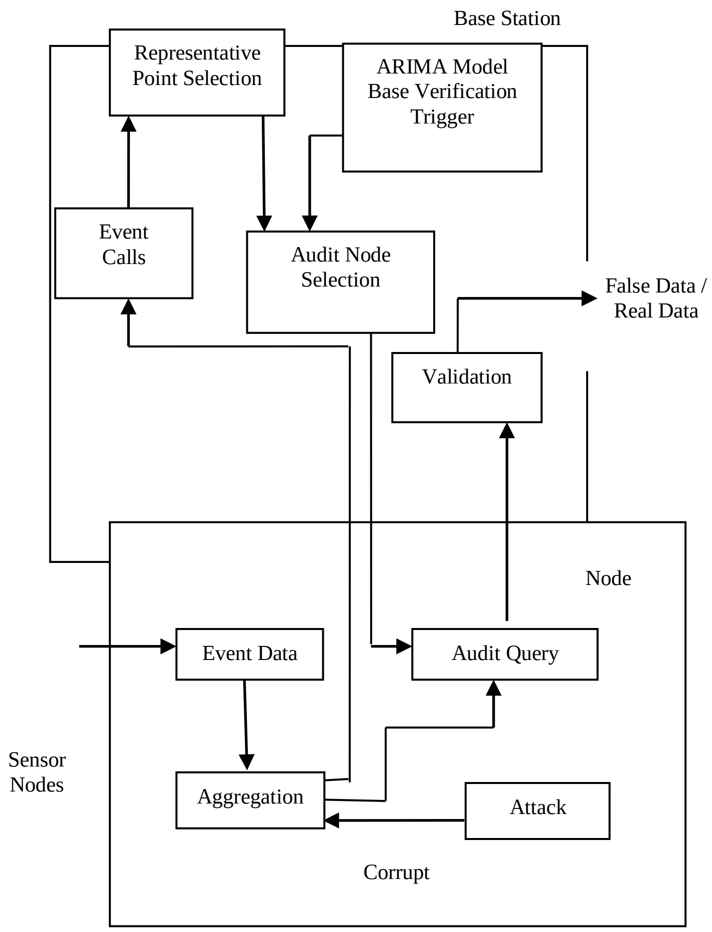


Fig 2: The system architecture

In our scheme, the correctness of representative points is checked by hypothesis testing techniques with samples from the WSN. While providing nice security properties, the sampling-based approach only requires a part of nodes to be involved in the verification, and enables verification not to rely on any particular in-network aggregation protocol. To protect the sampling procedure, verifiable random sampling is proposed to protect the legitimacy of sampled nodes, and local authentication based on spatial correlation among sensor readings is proposed to protect the validity of sample readings. As a result, our scheme can effectively verify the temporal variation patterns for in-network aggregation, while being able to achieve low additional energy cost and work with various in-network aggregation protocols.

II. PROPOSED ARCHITECTURE

The system architecture is shown in Fig 2. The modules in the architecture are shown below:

A. *RPS*

RPS requirement is to capture the temporal pattern of the whole aggregation result series with the help of sensor node input.

ISSN: 2278-0181
ICESMART-2015 Conference Proceedings

B. Aggregation

The sensor nodes were collected in a network.

C. Validation

BS verifies the legitimacy of sampled nodes; and, how to detect false samples provided by the malicious nodes.

D. Attack

The importance of temporal variation information of aggregation results, we focus on the attack against in-network aggregation that the adversaries attempt to distort the real temporal variation pattern of the aggregate by disrupting a series of successive aggregation results.

Notations: We list below notations used in this paper:

1) *u, v, w (in lower case) are sensor nodes.*
2) *N is the total number of sensor nodes.*
3) *N_u is the set of u's neighbors in u's communicationrange including itself.*
4) *N^2_u is the set of u's two-hop neighbors outside its communication range.*
5) *R_c is the communication radius of sensor nodes.*
6) *K_u is u's individual key shared between u and the BS.*
7) *MAC(k,m) is the message authentication code of message m generated with a symmetric key K.*
8) *$r_{u,t}$ is the sensor reading of u in epoch t.*

III. IN-NETWORK AGGREGATION

During the period of ain-network aggregation query, each sensor node caches l_{max} number of sensor readings thatcontribute to the aggregations in the latest l_{max} epochs. L_{max} determines the maximum length of the time window inwhich the temporal variation pattern of the aggregationresults can be verified.

Once the users observe an interesting temporal variationpattern of the aggregate, they can verify its authenticity on-demand.However, in the circumstance that the adversary isinterested in suppressing the real appearance of aninteresting temporal variation pattern, the users cannotdecide when to conduct verification because they do notknow when the interesting pattern really appears. Thus,periodic verification is required. To this end, the period ofthe aggregation query is divided into successive timewindows. Each time window consists of several successiveepochs. At the end of each time window, the temporalvariation pattern in this time window is verified.

Either in the on-demand verification or in the periodicverification, the BS selects some points from the series ofaggregation results in the time window to be verified, andchecks their correctness to detect any fabrication of temporalvariation patterns. Considering that the adversary canmanipulate only a small number of aggregation results suchas extreme points to tamper with the temporal variationpattern, it may be ineffective to check a set of randomlyselected points to detect forged patterns because the selectedpoints may not cover these manipulated points, whichcauses that the attack is not detected. Thus, to guaranteeeffective attack detection, the selected points should be ableto capture the temporal variation pattern in the time windowlike extreme points. We refer to these points as representativepoints and the epoch of a representative point as representativeepoch hereinafter. After the selection of representativepoints, the BS broadcasts a verification request, whichincludes the representative epochs, the sampling ratio ϱ,and a nonce number *nonce$_v$*, to the WSN. Once receiving theverification request, each node decides whether to act as asampled node. Before the sampled nodes send to the BS theirsensor readings of every representative epoch, their neighbouringnodes verify the correctness of sample data andauthenticate the sample messages.

With the sensor reading samples, the BS checks thecorrectness of the aggregation results of each representativeepoch by hypothesis testing. The general form of thehypothesis tests is

H_0: $A(t) = A_g(t)$ versus H_a: $A(t) \neq A_g(t)$.

If the aggregation results in all representative epochs areverified as correct, the temporal variation pattern in thetime window is assumed to be authentic.

IV. REPRESENTATIVE POINT SELECTION (RPS)

The definition of representative point toformally characterize the requirement that is to capture thetemporal pattern of the whole aggregation result series.Fig. 3 shows an example.

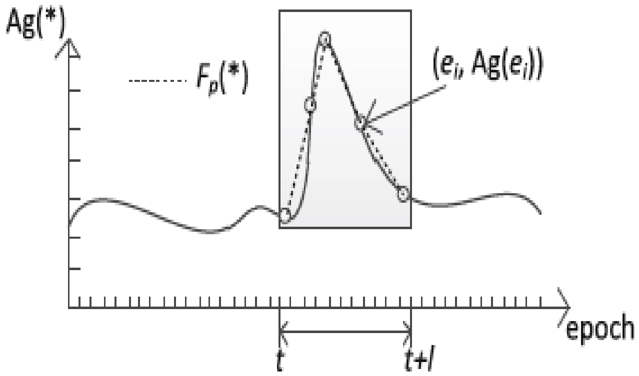


Fig 3: Definition of representative points.

Let $P = \{(e_i, Ag(e_i)) \mid 1 \leq i \leq p, e_1 = t < e_2 < \ldots < e_{p-1} < e_p = t + l\}$ be a set ofpoints in the time window [t,t+l], where $Ag(e_i)$ is the aggregation result in epoch e_i. Let $F_p(*)$ be the piece wise linear function consisting of connected line segments, each ofwhich is between point $(e_i, Ag(e_i))$ and $(e_{i+1}, Ag(e_{i+1}))$ for $1 \leq i \leq p - 1$. If F_p is a best approximation of the series of aggregation results Ag(*) within [t, t+l] among all possible $F_{p'}$ where $P' = \{(e'_i, Ag(e'_i)) \mid 1 \leq i \leq p, e'_1 = t < e'_2 < \ldots < e'_{p-1} < e'_p = t+l\}$, we say P captures the temporalpattern of aggregation results and the points in P arerepresentative points in the time window [t, t+l]. Here, thegoodness of approximation is assessed by the approximation error between $F_p(*)$ and Ag(*),which is measured by theirEuclidean distance

$$E(t, t+l) = \sqrt{\sum_{k=t}^{t+l} \{Ag(k) - F_p(k)\}^2}$$

A. Representative Point Selection

The RPS problem can be described asfollows: Given an integerp($p\geq2$), find a set of points $P = \{(e_i, Ag(e_i)) \mid 1\leq i\leq p, e_1 = t < e_2 < . . . < e_{p-1} < e_p = t + l\}$ such that the error of approximation of Ag(*) by F_p(*)in the time window [t,t+l] is minimized and |P| = p.

B. RPS with Prespecified Points (RPS-P)

With the knowledge of RPS algorithm and the ability ofpredicting the real temporal variation pattern of theaggregate, the adversary may try to forge a series ofaggregation results of which the selected representativepoints have aggregation values equal or close to the realones. If such attempt is successful, the check of representativepoints will not detect the fabrication of the temporalvariation. Fig. 3 shows an example of fabricated series ofaggregation results and the representative points selected byRPS algorithm over the fabricated series. The aggregationvalues of representative points are the same as the realaggregation results, which causes that the false patternbetween epoch 0 and 9 cannot be detected. Considering suchpossisbility, the randomness is introduced to make the outputof the selection algorithm unpredictable.To this end, eachdata point in (t, t+l) is prespecified as a representative point with a probability of qin our scheme. Then, the remaining number of representativepoints including the ones at two boundary epochs t and t+l are selected to minimize the approximation error. Onthe other hand, some points such as the maximum andminimum aggregation results, which describe the significantcharacteristics of the temporal variation pattern, should be always prespecified as representative points.

C. The Number of Representative Points

Selecting more representative points can further enhance thecapability of our scheme to detect forged temporal variationpattern because a larger number of representative points canbetter capture the variation pattern of aggregation resultsand have a higher probability to cover the manipulatedperiod. However, since each representative point needs tobe verified by collecting sensor reading samples in thecorresponding representative epoch from the WSN, morerepresentative points mean higher communication cost.Therefore, there is a trade-off between detecting capabilityand communication cost.Actuallyaddress the optimal representative point selection to minimizethe approximation error with a given budget oncommunication cost, i.e., a given number of representativepoints. On the other hand, the users would need to decide atleast how many representative points are required to achievethe desired detecting capability of the scheme. Thus, here weconsider the problem of minimizing the number of representativepoints given a certain degree of the approximationerror that the users can tolerate.

V. AGGREGATION VERIFICATION

Selecting more representative points can further enhance the capability of our scheme to detect forged temporal variation pattern because a larger number of representative points can better capture the variation pattern of aggregation results and have a higher probability to cover the manipulated period. However, since each representative point needs to be verified by collecting sensor reading samples in the corresponding representative epoch from the WSN, more representative points mean higher communication cost. Therefore, there is a trade-off between detecting capability and communication cost.

Once broadcasting the verification request, the BS waits for some time t_w to ensure the arrivals of all samples.Considering the network delivery time of the verificationrequests and sample messages, t_w should be at least twiceof the message delivery time from the network boundaryto the BS plus the time for the local sample authentication.According to the procedure of local sample authentication,the time required to complete it consists of the time of two-hopbroadcast from a sample node and two-hop broadcastfrom each of its neighbour nodes, and also the time foreach neighbour to collect sensor readings in its two-hopneighbourhood and for the sampled node to collect MACsfrom its one-hop neighbour's. These times can be easilyestimated and accordingly the time for the local sampleauthentication can be estimated. When time expires, the BSfirst checks the validity of every arrived sample and thesample size, and then verifies the aggregation results inrepresentative epochs.

A. Sample Message Verification

For every sample message, say S_v claimed from node v, theBS verifies its validity in two steps. First, the BS verifies thelegitimacy of the claimed sampled node v by checkingwhether Inequality holds because the BS knows h, nonce, $nonce_v$, and K_v. Then, the BS verifies XMAC in the samplemessage. Since the BS holds the seed key $K^s_u$of any node u, it can generate u's authentication key $K^a_{u,v}$. The BS generates $K^a_{u,v}$ for each node u_i in the ID list (u_1, . . . ,u_T) in S_v, recomputed XMAC, and compare it with the one in S_v for equality. If the verification in any step above fails, the BS drops S_v and raises an alarm. Otherwise, the BS accepts S_v. In this way, all invalid sample messages are dropped.

During the local sample authentication, a false samplemay pass the local verification and be successfully authenticatedby c neighbour's due to sufficient number of compromisednodes in the same neighbourhood. However, it isexpensive for the adversary to provide a large portion offalse samples because of the verifiable random sampling andlocal sample authentication. Thus, we assume the number offalse samples is relatively small to the total sample size and we can use Rosner's test to detect outlying sensor readings ineach representative epoch. The sampled nodes from whichoutlying sensor readings are detected are labelled as outlyingnodes and the hypothesis testing is conducted over thesamples excluding those from outlying nodes.

VI. EXPERIMENTAL EVALUATION

In this section, the representative epochs are uniformlychosen from a time window with an interval of 10 epochs.The performance of the local sample authentication isevaluated by the following two metrics:

Approval rate of real samples

The ratio of the number of nodes whose data can be successfully authenticated by at least c neighbors to the total number of nodes inthe benign environment. Even in benign environment,not all samples would be successfully authenticatedin practice. The samples that cannot be authenticatedwill not be accepted by the BS. This metric indicatesthe degree of influence of the local sample authenticationon the availability of real samples.

Disapproval rate of false samples

The ratio of the number of false samples that cannot be successfully authenticated by up to c neighbours to the total number of compromised sampled nodes in the hostile environment. It indicates the degree of the prevention of the false samples by the local sample authentication.

Fig. 4 illustrates the approval rate of real samples in each time window under different security threshold c. As we can see, the approval rate in each time window decreases as c increases. This is because the number of nodes having up to c neighbours decreases as c increases. When c = 1 and c = 2, the approval rate is higher than 90 and 85 percent, respectively. However, the approval rate is lower than 80 percent when c=3, which is because that the simulated network is sparse (the average degree of the nodes is 5). It indicates that with a reasonable value of c, here say 2, our local sample authentication approach have a small effect on the availability of real samples.

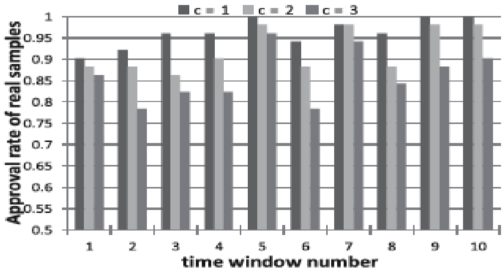
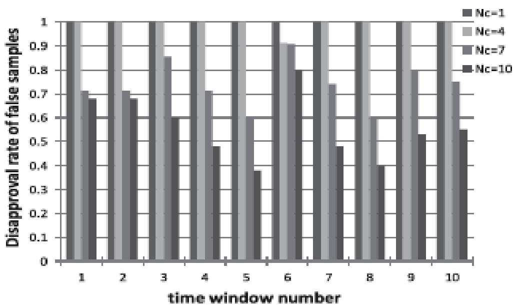
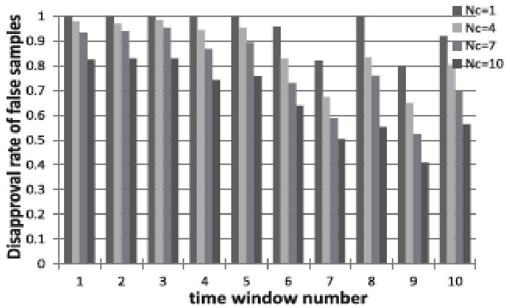


Fig 4: Approval rate of real samples in every time window.

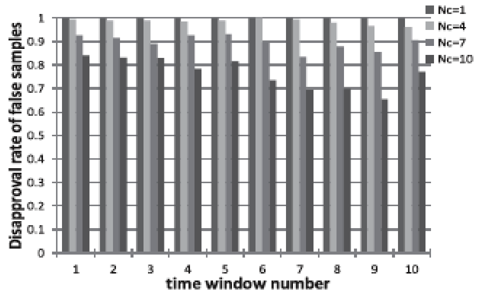
Figs. 5a, 5b, and 5c show the disapproval rate of falsesamples, respectively, generated by the above three manners under different N_c. The results are averaged over 50runs. In each run, N_c nodes are randomly selected ascompromised nodes. In each figure, we can see that thedisapproval rate of false samples decreases as N_c increases in every time window. This is because that more compromisednodes would incur a higher probability for that acompromised node providing false samples has c compromisedneighbour nodes to launch the collusion attack. When N_c= 1 and N_c= 4, the disapproval rate is higher than80 percent in all three figures. Since the network size is small(51 nodes), 10 compromised nodes make up a significantfraction of the network and cause the worst results.



(a)



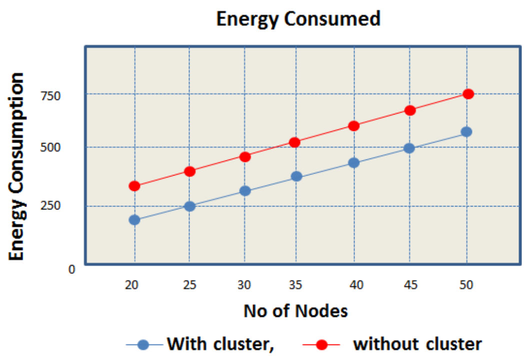
(b)



(c)

Fig 5: Disapproval rate under three manners of forging false samples: (a) Disapproval rate of false samples generated by the first manner. (b) Disapproval rate of false samples generated by the second manner. (c) Disapproval rate of false samples generated by the third manner.

Fig 7 shows the graph performance in which energy consumed verses number of nodes.



A. Fig 6: The performance of energy consumed.

VII. CONCLUSION

In this paper, we identify distinct design issues for secure in-network aggregation in WSNs. An efficient verification scheme is proposed to protect the authenticity of the temporal variation patterns in the aggregation results. Our scheme only need to check a small portion of aggregation results in a time window and, thus, greatly reduces the verification cost. We define representative points and propose corresponding algorithms for representative point selection. By exploiting the spatial correlation among the sensor readings in close proximity, a series of security mechanisms are also proposed to protect the sampling procedure. The correctness of representative points is checked by hypothesis testing techniques with samples from the WSN. While providing nice security properties, the sampling-based approach only requires a part of nodes to be involved in the verification, and enables verification not to rely on any particular in-network aggregation protocol. To protect the sampling procedure, verifiable random sampling is proposed to protect the legitimacy of sampled nodes, and local authentication based on spatial correlation among sensor readings is proposed to protect the validity of sample readings.As a result, our scheme can effectively verify the temporal variation patterns for in-network aggregation, while being able to achieve low additional energy cost and work with various in-network aggregation protocols.Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions. Authors are strongly encouraged not to call out multiple figures or tables in the conclusion—these should be referenced in the body of the paper.

REFERENCES

[1] Z. Cai, S. Ji, J.S. He, and A.G. Bourgeois, “Optimal Distributed Data Collection for Asynchronous Cognitive Radio Networks”, Proc. IEEE 32nd Int’l Conf. Distributed Computing Systems (ICDCS), 2012.

[2] S. Ji and Z. Cai, “Distributed Data Collection and Its Capacity in Asynchronous Wireless Sensor Networks,” Proc. IEEE INFOCOM, pp. 2113-2121, Mar. 2012.

[3] Clouqueur, T., Phipatanasuphorn, V., Ramanathan, P., Saluja, K.K.: Sensor De- ployment Strategy for Detection of Targets Traversing a Region. In: ACM Mobile Networks and Applications. Volume 8. (2003) 453–461.

[4] Cristescu, R., Beferull-Lozano, B., Vetterli, M.: On Network Correlated Data Gathering. In: Proc. of IEEE INFOCOM. (2004).

[5] Krishnamachri, B., Estrin, D., Wicker, S.: Modelling Data-centric Routing in Wireless Sensor Networks. In: Proc. of IEEE INFOCOM. (2002).

[6] K.-W. Fan, S. Liu, and P. Sinha, “On the Potential of Structure-Free Data Aggregation in Sensor Networks,” Proc. IEEE INFOCOM, 2006.

[7] A. Manjhi, S. Nath, and P.B. Gibbons, “Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 287-298, 2005.

[8] B. Przydatek, D. Song, and A. Perrig, “SIA: Secure Information Aggregation in Sensor Networks,” Proc. ACM First Int’l Conf. Embedded Networked Sensor Systems (SenSys), pp. 255-265, 2003.

[9] Y. Yang, X. Wang, S. Zhu, and G. Cao, “SDAP: A Secure Hop-By- Hop Data Aggregation Protocol for Sensor Networks,” Proc. ACM MobiHoc, pp. 356-367, 2006.

[10] J. Hill, R. Szewczyk, A, Woo, S. Hollar, D. Culler, and K. Pister, “System Architecture Directions forNetworked Sensors”, ASPLOS, November 2000.

[11] K.B. Frikken and J.A. Dougherty IV, “An Efficient Integrity- Preserving Scheme for Hierarchical Sensor Aggregation,” Proc. ACM First Conf. Wireless Network Security (WiSec), pp. 68-76, 2008.

[12] B. Yu, J. Li, and Y. Li, “Distributed Data Aggregation Scheduling in Wireless Sensor Networks,” Proc. IEEE INFOCOM, pp. 2159- 2167, 2009.

[13] R.G.M. Bellare and P. Rogaway, “XOR MACs: New Methods for Message Authentication Using Finite Pseudo-Random Functions,” Proc. Advances in Cryptology (Crypto), 1995.

[14] L. Hu and D. Evans, “Secure Aggregation for Wireless Networks,” Proc. Workshop Security and Assurance in Ad Hoc Networks, p. 384, 2003.

[15] F.E. Grubbs, “Procedures for Detecting Outlying Observations in Samples,” Technometrics, vol. 11, no. 1, pp. 1-21, Feb. 1969.

[16] M. Garofalakis, J. Hellerstein, and P. Maniatis, “Proof Sketches: Verifiable In-Network Aggregation,” Proc. IEEE 32nd Int’l Conf. Data Eng. (ICDE), pp. 996-1005, Apr. 2007.

[17] S. Roy, M. Conti, S. Setia, and S. Jajodia, “Securely Computing an Approximate Median in Wireless Sensor Networks,”Proc Fourth Int’l Conf. Security and Privacy in Comm. Networks, pp. 6:1-6:10, 2008.

[18] P. Flajolet, G.N. Martin, and G.N. Martin, “Probabilistic Counting Algorithms for Data Base Applications,” J. Computer and System Sciences, vol. 31, pp. 182-209, 1985.