ADAPTIVE MULTIPATH SELECTIVE SECURE DATA FORWARDING IN SENSOR NETWORKS

A. Maria Franclin Monisha

S.Varatharajan

Abstract

The wireless sensor network data aggregation significantly reduces the amount of communication and energy consumption. In the existing system a robust aggregation framework called synopsis diffusion which combines multipath routing schemes with duplicate-insensitive algorithms to accurately compute aggregates (e.g., predicate Count, Sum) in spite of message losses resulting from node and transmission failures. Lightweight verification algorithm is used in the existing system by which the base station can determine if the computed aggregate (predicate Count or Sum) includes any false contribution. However, this aggregation framework does not address the problem of false sub-aggregate values contributed by compromised nodes resulting in large errors in the aggregate computed at the base station, which is the root node in the aggregation hierarchy. A new novel technique called Selective forwarding schemes will depend on parameters such as the available battery at the node, the packet delivery ratio cost of retransmitting a message, or the importance of messages. Here we plan to design an efficient attack-resilient computation algorithm. This algorithm would guarantee the successful computation of the aggregate even in the presence of an attack. More sophisticated schemes will achieve better importance performance, but will also require information from other sensors. Suboptimal schemes that rely on local estimation algorithms and entail reduced computational cost are also designed.

Index terms: Base station, data aggregation, in-network aggregation, sensor network security, synopsis diffusion, selective forwarding scheme.

I . INTRODUCTION

In large WSNs, computing aggregates in-network i.e., combining partial results at intermediate nodes during message significantly reduces the amount of communication and hence the energy consumed. The important aggregates considered by the research community include Count, and Sum. Average can be computed from Count and Sum. A Sum algorithm can be also extended to compute Standard Deviation and Statistical Moment of any order. A robust and scalable aggregation framework called synopsis diffusion is used for computing duplicate-sensitive aggregates, such as Count and Sum. This approach uses a ring topology where a node may have multiple parents in the aggregation hierarchy, and each sensed value or sub aggregate is represented by a duplicate-insensitive bitmap called synopsis. To compute aggregates, such as Count and Sum, and to enable the base station to verify if the computed aggregate is valid. We call this algorithm the verification algorithm; it is an aggregate computation and verification algorithm. The key observation which we exploit to minimize the communication overhead of this algorithm is that to verify the correctness of the final synopsis (the aggregate of the whole network) the base station does not need to receive authentication messages from all of the nodes.

However, most of the existing in-network data aggregation algorithms have no provisions for security. of message losses resulting from node and transmission failures. However, this aggregation framework does not address the problem of false sub-aggregate values contributed by compromised nodes resulting in large errors in the aggregate computed at the base station, which is the root node in the aggregation hierarchy. Selective forwarding schemes will depend on parameters such as the available battery at the node, the packet delivery ratio cost of retransmitting a message, or the importance of messages. In the proposed system we are using an efficient attack-resilient computation algorithm. This algorithm would guarantee the successful computation of the aggregate even in the presence of an attack. More sophisticated schemes will achieve better importance performance, but will also require information from other sensors. Suboptimal schemes that rely on local estimation algorithms and entail reduced computational cost are also designed.

The rest of the paper deals with the following sections: section II explains the synopsis diffusion approach, section III discuss about attacks, section IV discuss about verification algorithm, section V describe the proposed work, section VI about the conclusion of the paper.

II. SYNOPSIS DIFFUSION

An aggregation framework called synopsis diffusion which uses a ring topology. During the query distribution phase, nodes form a set of rings around the base station (BS) based on their distance in terms of hops from BS. By Ti we denote the ring consisting of the nodes which are i hops away from BS. In the subsequent aggregation period, starting in the outermost ring, each node generates and broadcasts a local synopsis SG (v), SG () where is the synopsis generation function and is the sensor value relevant to the query. A node in ring Ti, will receive broadcasts from all of the nodes in its communication range in ring Ti+1.

It will then combine its own local synopsis with the synopses received from its children using a synopsis fusion function SF () and then broadcast the updated synopsis. Thus, the fused synopses propagate level-by-level until they reach BS, which first combines the received synopses using SF () and then uses the synopsis evaluation function SE() to translate the final synopsis to the answer to the query. We now describe the duplicate-insensitive synopsis diffusion algorithms for Count and Sum. These algorithms are based on a probabilistic algorithm for counting the number of distinct elements in a multiset.

A. Count:

The synopsis fusion function is the bitwise Boolean OR of the synopses being combined. Each node fuses its local synopsis with the synopses it receives from its children. Let denote the final synopsis computed by BS by combining all of the synopses received from its child nodes. We observe that will be a bit vector of length of the form , where is the lowest order bit in that is 0. BS can estimate Count from via the synopsis evaluation function. Algorithm for count,

Algorithm 1 $CT(X, \eta)$

```
begin
    i = 1;
    while $i < \eta + 1\ AND\ h(X, i) = 0$ do
        i = i + 1;
    end
    return i;
end
```

B. Sum:

The Count algorithm can be extended for computing Sum. The synopsis generation function for Sum is a modification of that for Count, while the fusion function and the evaluation function for Sum are identical to those for Count. A Sum algorithm can be also extended to compute Standard Deviation and Statistical Moment of any order. Algorithm for sum,

Algorithm 2 $SG_{\text{sum}}(X, v_X, \eta)$

```
begin
    $Q^X[j] = 0\ \forall j\ 1 \leq j \leq \eta$;
    i = 1;
    while $i \leq v_X$ do
        $X_i = \langle X, i \rangle$;
        $j = CT(X_i, \eta)$;
        $Q^X[j] = 1$;
        $i = i + 1$;
    end
    return $Q^X$;
end
```

III. ATTACKS

Since BS estimates the aggregate based on the lowest order bit that is “0” in the final synopsis B^C, a compromised node would need to falsify its fused synopsis such that it would affect the value of. It can accomplish this by simply inserting “1”s in one or more bits in positions, where $z \le j \le \eta$, in B^C which it broadcasts to its parents. Let $B\wedge^C$ denote the synopsis finally broadcast by C. Note that does not need to know the true value of; it can simply set some higher order bits to “1” with the expectation that this will affect the value of computed by BS.

Since the synopsis fusion function is a bitwise Boolean OR, the fused synopsis computed at any node which is at the higher level than node C on the aggregation hierarchy will contain the false contributions of node C. We observe that when a node X computes the fused synopsis $B\wedge^X$, X is not sure if contains any false “1”s contributed by a compromised node lower in the hierarchy. The observation is true also for the BS when it computes the final synopsis B□ .We call the “1” bits which are present in B□ but not B in the false “1”s.

A compromised node can introduce a false “1” at bit in by launching either of the following attacks.

1) Falsified sub aggregate attack: C just flips bit j in $B\wedge^C$ from “0” to “1”—not having a local aggregate justifying that “1” in the synopsis $B\wedge^C$.

2) Falsified local value attack: C injects a false “1” at bit j in its local synopsis, Q^C. The falsified synopsis, $Q\wedge^C$, induces bit j in $B\wedge^C$ to be “1”. Note that true local sensed value, V_C, corresponds to $Q^{C.}$

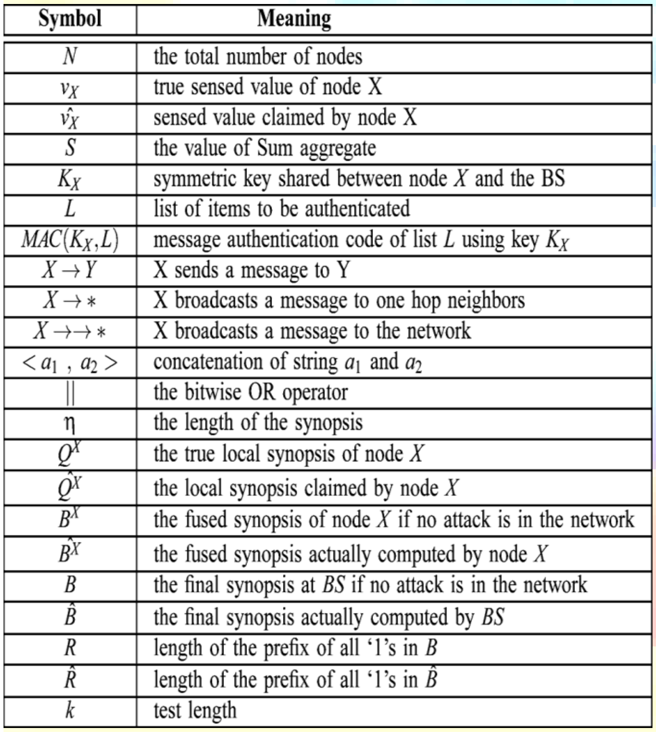
IV.VERIFICATION ALGORITHM

BS can verify the final synopsis if it receives one valid MAC for each “1” bit in the synopsis. In fact, to verify a particular “1” bit, say bit, BS does not need to receive authentication messages from all of the nodes which contribute to bit. As an example, more than half of the nodes are likely to contribute to the leftmost bit of the synopsis, while to verify this bit, BS needs to receive a MAC only from one of these nodes. Hence, it is sufficient for each node in the aggregation hierarchy to forward only one MAC corresponding to each “1” bit in the synopsis.

Our verification algorithm further reduces the communication overhead per node. In particular, each node forwards one MAC each for at most bits in the synopsis, where is a small constant. This ensures, as shown later, that BS will be able to authenticate the rightmost “1” bits in the final synopsis. Then, as proven later, BS can securely compute with very high probability, where is the length of the prefix of consecutive “1”s in the final synopsis .We remind the reader that determines the value of the final aggregate. The higher the value of, the greater is the probability that our scheme will detect a false “1” bit in the final synopsis.

TABLE

NOTATIONS TO DESCRIBE SUM VERIFICATION



Symbol	Meaning
N	the total number of nodes
v_X	true sensed value of node X
$\hat{v_X}$	sensed value claimed by node X
S	the value of Sum aggregate
K_X	symmetric key shared between node X and the BS
L	list of items to be authenticated
$MAC(K_X, L)$	message authentication code of list L using key K_X
$X \rightarrow Y$	X sends a message to Y
$X \rightarrow *$	X broadcasts a message to one hop neighbors
$X \rightarrow\rightarrow *$	X broadcasts a message to the network
$< a_1 , a_2 >$	concatenation of string a_1 and a_2
$\Vert$	the bitwise OR operator
η	the length of the synopsis
Q^X	the true local synopsis of node X
$\hat{Q}^X$	the local synopsis claimed by node X
B^X	the fused synopsis of node X if no attack is in the network
$\hat{B}^X$	the fused synopsis actually computed by node X
B	the final synopsis at BS if no attack is in the network
$\hat{B}$	the final synopsis actually computed by BS
R	length of the prefix of all ‘1’s in B
$\hat{R}$	length of the prefix of all ‘1’s in $\hat{B}$
k	test length

Protocol Operation:

The verification protocol runs concurrently with the original synopsis diffusion protocol described as follows. However, for ease of exposition, we describe our verification protocol with respect to one single synopsis. Each synopsis can be verified independently and hence our algorithm is readily applicable for computing multiple synopses.

1) **Query Dissemination**: In this phase, BS broadcasts the name of the aggregate to compute, a random number Seed and the chosen value of "test length", k . The query that BS broadcasts is as follows (F_{agg} is the name of the aggregate (e.g., "Sum")):

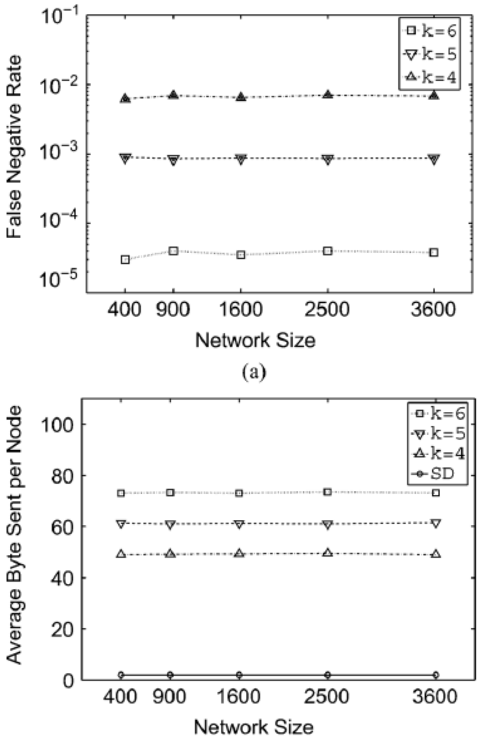
$$BS \rightarrow \rightarrow *: (F_{agg} , Seed, k)$$

During this phase, nodes form a set of rings around BS based on their distance in hops from BS.

2) **Aggregation Phase**: Each node executes the aggregation phase of the original synopsis diffusion protocol along with sending some authentication messages. Recall that during the falsified sub aggregate attack the fused synopsis, $B^{\wedge X}$ computed at a node X can be different from X's true fused synopsis $B^{X.}$

SIMULATION RESULT FOR VERIFICATION PROTOCOL

Our simulations were written based on the TAG simulator. In particular, we added the security functionality to the source code provided by Considine et al., which simulates their multipath aggregation algorithm in the TAG simulator environment. The simulation result of verification protocol is shown in the following figure,



(a)

(b)

Simulation results for the verification protocol. (a) False negative rate. (b) Bytes sent.

RESULTS AND DISCUSSION:

We now present the results of the experiments. As Count can be considered as a special case of Sum, here we discuss only the results related to Sum aggregate.

We did not study the false positive rate of the verification protocol. The integrity checks in node-to-node communication ensures that if no attack is launched, BS will receive at least one MAC for each of the rightmost "1"s in the final synopsis B. A corrupted MAC that is a consequence of something besides an attack (e.g., communication error) can reach the BS. However, this problem is not protocol-dependent. Since the verification protocol completes in one epoch irrespective of the final result (success or failure), we did not study the latency in our simulation. We present the following results for a single synopsis, which can be extended for multiple synopses.

False Negative Rate:

We considered the worst case attack scenario: The attacker knows the network topology and the synopsis computed by each node. That is, the attacker can compute the final synopsis received by the BS. So, the attacker is able to check if the following event E^K, occurs in the final synopsis: "1"s are present to the right of a "0" bit, say bit j.

The aim of the attacker is to increase the value of Sum as much as possible while remaining undetected. So, the attacker takes the following strategy: If E^K occurs, it changes all "0"s at positions $\leq$ j to "1"s; otherwise, it does nothing. In fact, if the attacker modifies a bit after the jth bit, that would be detected—the protocol verifies the MACs of the rightmost "1"s.

Communication Overhead:

We compare the communication overhead of the verification protocol to that of the original synopsis diffusion (SD) approach. Fig. Plots the number of bytes a node transmits on average during the verification protocol considering different network sizes.

This figure also shows the per-node byte overhead of the original SD approach. We assume that the size of a MAC is 8 bytes and the size of each synopsis is 2 bytes (compressed using run-length coding as used.. In our experiment, the size of a node ID is 2 bytes and a sensed value is represented by 2 bytes. We observe that the verification protocol costs roughly bytes of extra overhead for each node compared with the original SD approach. We also observe that the byte overhead does not increase with the network size, which shows the scalability of our approach.

V. PROPOSED WORK

Here a method called selective data forwarding includes, in a network of interconnected computer system nodes, receiving a request from a source system to store data, the request comprising an ownership and a data type, if the ownership and the data type match a corresponding entry in a store, directing the data to a computer memory, and continuously forwarding the data from one computer memory to another computer memory in the network of interconnected computer system nodes without storing on any physical storage device in the network.

An efficient attack-resilient computation algorithm is also used. This algorithm would guarantee the successful computation of the aggregate even in the presence of an attack.

VI. CONCLUSION

We discussed the security issues of in network aggregation algorithms to compute aggregates such as predicate Count and Sum. We discussed how a compromised node can corrupt the aggregate estimate of the base station, keeping our focus on the ring-based hierarchical aggregation algorithm.

Lightweight verification algorithm which would enable the base station (BS) to verify whether the computed aggregate was valid but does not compute aggregate in presence of attack. So we plan to design an efficient attack-resilient computation algorithm. This algorithm would guarantee the successful computation of the aggregate even in the presence of an attack and to reduce the communication overhead with less cost.

REFERENCES

[1] James Reserve Microclimate and Video Remote Sensing 2006 [Online]. Available: http://research.cens.ucla.edu 1052 IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO. 3, JUNE 2012

[2] S. Madden, M. J. Franklin, J. M. Heller stein, and W. Hong, “TAG: A tiny aggregation service for ad hoc sensor networks,” in Proc. 5th USENIX Symp. Operating Systems Design and Implementation (OSDI), 2002.

[3] J. Zhao, R. Govindan, and D. Estrin, “Computing aggregates for monitoring sensor networks,” in Proc. 2nd Int. Workshop Sensor Network Protocols Applications, 2003.

[4] J. Considine, F. Li, G. Kollios, and J. Byers, “Approximate aggregation techniques for sensor databases,” in Proc. IEEE Int. Conf. Data Engineering (ICDE), 2004.

[5] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, “Synopsis diffusion for robust aggregation in sensor networks,” in Proc. 2nd Int. Conf. Embedded Networked Sensor Systems (SenSys), 2004.

[6] M. Garofalakis, J. M. Hellerstein, and P. Maniatis, “Proof sketches: Verifiable in-network aggregation,” in Proc. 23rd Int. Conf. Data Engineering (ICDE), 2007.

[7] M. B. Greenwald and S. Khanna, “Power-conservative computation of order-statistics over sensor networks,” Proc. 23th SIGMOD Principles of Database Systems (PODS), 2004.

[8] P. Flajolet and G. N. Martin, “Probabilistic counting algorithms for data base applications,” J. Computer Syst. Sci., vol. 31, no. 2, pp. 182–209, 1985.

[9] D. Wagner, “Resilient aggregation in sensor networks,” in Proc. ACM Workshop Security of Sensor and Adhoc Networks (SASN), 2004.

[10] L. Buttyan, P. Schaffer, and I. Vajda, “Resilient aggregation with attack detection in sensor networks,” in Proc. 2nd IEEE Workshop Sensor Networks and Systems for Pervasive Computing, 2006.

[11] B. Przydatek, D. Song, and A. Perrig, “SIA: Secure information aggregation in sensor networks,” in Proc. 1st Int. Conf. Embedded Networked Sensor Systems (SenSys), 2003.

[12] H. Chan, A. Perrig, and D. Song, “Secure hierarchical in-network aggregation in sensor networks,” in Proc. ACM Conf. Computer and Communications Security (CCS), 2006.

[13] K. B. Frikken and J. A. Dougherty, “An efficient integrity-preserving scheme for hierarchical sensor aggregation,” in Proc. 1st ACM Conf. Wireless Network Security (WiSec), 2008.

[14] Y. Yang, X. Wang, S. Zhu, and G. Cao, “SDAP: A secure hop-by-hop data aggregation protocol for sensor networks,” in Proc. Seventh ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc), 2006.

[15] S. Nath, H. Yu, and H. Chan, “Secure outsourced aggregation via one-way chains,” in Proc. 35th SIGMOD Int. Conf. Management of Data, 2009.

Author's Detail:

A. Maria Franclin Monisha received the B.Tech degree in Information Technology from, PSN Engineering College, Tirunelveli in 2011. Presently pursuing M.Tech degree in Information Technology in PSN College of Engineering and Technology, Tirunelveli.