Brief Announcement: DoS-Resilient Secure Aggregation Queries in Sensor Networks

Haifeng Yu
National University of Singapore
haifeng@comp.nus.edu.sg

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications*

General Terms

Algorithms, Design, Security

Keywords

Secure aggregation queries, DoS attacks, sensor networks, multihop flooding attacks

1. INTRODUCTION

Wireless sensor networks are often queried for aggregates such as predicate count (e.g., number of sensors sensing fire) and sum. Since sensors may potentially be compromised, there is a clear need for security. Most previous work on secure aggregation queries focuses on defending against the so-called *stealth attack*, where the adversary's goal is to fool the base station into accepting an incorrect[1] result. Specifically, one recent effort [1] considers securely computing sum in a multihop sensor network using tree-based in-network aggregation. Intuitively, this would require the *inclusion* of all n readings from the n sensors and the *exclusion* of additional fabricated readings. Chan et al. [1] uses a commitment protocol to verify inclusion while incurring $O(\log^2 n)$ *edge congestion* (i.e., maximum number of messages on individual edges in the wireless topology) with $O(\log n)$ bits per message. To exclude fabricated readings, their approach assumes that all sensors are alive and reachable, and each reading is an integer between 0 and m. It then uses an elegant complementing technique and computes a sum for the complements (i.e., m minus the reading). The final result will be accepted only if the sum and complement sum add up to exactly $n \cdot m$.

This paper argues that it is crucial not to limit our attention to only stealth attacks. For example, in [1], even if the adversary compromises a single sensor, it can forever prevent the base station from getting correct aggregates, which is equivalent to a global network-wide DoS attack. The adversary can achieve the same global DoS effect by destroying a single sensor physically, or by radio jamming a single senor. Finally, even without an adversary, the base station will still reject all results as long as any single sensor naturally dies.

Our goal thus is to design efficient algorithms for multihop sensor networks to correctly compute predicate count and sum despite natural sensor deaths, DoS attacks (including the serious *multihop flooding attack* [2]), and compromised sensors. To the best of our knowledge, this paper is the first effort toward such a goal.

Our algorithms provide approximate answers instead of exact results. Notice that sensor readings are inherently approximate. Our first protocol, the *verifiable aggregate synopsis* (VAS) protocol, adopts ideas from synopsis diffusion [5] for aggregation queries in trusted sensor networks. In synopsis diffusion, the final (approximate) answer is computed from a small bit vector called *synopsis*. We observe that the nature of the synopsis makes it particularly suitable for security. Specifically, only readings from $O(\log n)$ (random) sensors are ultimately "accountable" for the final/aggregate synopsis. After (conceptually) verifying these readings and incurring $O(\log n)$ edge congestion, the base station is assured that fabricated readings (if any) did not contribute to the final result. This avoids the complementing approach [1], and enables the protocol to obtain a final answer despite natural sensor deaths. Second, each sensor can locally verify the inclusion of its reading by examining the final synopsis, incurring $O(1)$ edge congestion. With the presence of adversarial behavior, the VAS protocol will not generate a result but will produce a (distributed) audit trail, which enables later pinpointing and revocation of compromised sensors.

If the number of compromised sensors is large, revoking them may incur excessive overhead. Thus we have developed a second *broadcast sampling* protocol. This protocol is more powerful and can compute a final (approximate) answer with the standard (ϵ, δ) guarantee (i.e., with at least $1-\delta$ probability, the approximation result is within $(1 \pm \epsilon)$ multiplicative factor of the accurate result) despite natural sensor deaths, DoS attacks, and compromised sensors. To achieve this, we leverage sampling and depart from the traditional paradigm of in-network aggregation, which is inherently vulnerable to multihop flooding attacks. Additionally, while in-network aggregation is quite efficient in trusted environment, such advantage has almost vanished (e.g., with $O(\log^2 n)$ [1] edge congestion) in unsecured environment.

A major challenge in sampling is how to effectively sample and obtain a multiplicative-ϵ approximation when the predicate count or sum is small. Our broadcast sampling algorithm leverages the broadcast nature of sensor networks and produces multiplicative-ϵ approximation regardless how small the

[1]A *correct* result here allows compromised sensors to report arbitrary readings for themselves, but they are not allow to add additional fabricated readings or change the reported readings of other sensors.

PODC'07, August 12–15, 2007, Portland, Oregon, USA.
ACM 978-1-59593-616-5/07/0008.

count or sum is. For predicate count, the algorithm takes only $O(\frac{1}{\epsilon}\log(\frac{1}{\delta})(\log n + \frac{\log\log n}{\epsilon^2}))$ samples. With fixed ϵ and δ, the number of samples taken becomes $O(\log n)$ (and thus $O(\log n)$ edge congestion). Notice that this breaks the well known lower bound on sampling when the count/sum is small. For sum, the algorithm currently incurs $O(\log m \log\log m)$ times more samples and edge congestion. We are currently improving this $O(\log m \log\log m)$ factor.

The broadcast sampling protocol further uses a novel *keyed predicate test* as a building block, which makes the sampling process resilient to DoS attacks. A modified version of the broadcast sampling protocol is used to automatically revoke malicious sensors based on audit trails generated by the VAS protocol. Notice that the revocation protocol must itself be DoS-resilient, instead of generating additional audit trails.

Because the broadcast sampling algorithm is relatively intricate, we focuses on the VAS protocol in the remainder of this paper due to space limitations.

2. SYSTEM MODEL AND ATTACK MODEL

We consider a general multihop sensor network where the base station is completely trusted. The number of sensors is at most n, where n is known beforehand. The actual number of live sensors or deployed sensors is unknown.

The adversary has a network-wide presence and may record or inject messages at any point in the network. The adversary may also compromised *arbitrary* number of sensors. Sensors that are not compromised are *honest*. The adversary may further launch a wide range of DoS attacks, including physically destroying sensors, radio jamming, and multihop flooding. In multihop flooding [2], the compromised sensors generate a large number of spurious responses, which are forwarded by honest sensors to the base station. Because senors have limited forwarding capability, this serious attack effectively prevents real responses from ever reaching the base station.

We assume that all honest sensors (together with the base station) form a connected component even if we remove all sensors that have failed naturally, compromised sensors, and sensors that are physically destroyed or are radio jammed. We assume that each sensor shares a unique symmetric key with the base station, and the base station has a mapping between sensors' unique IDs and these symmetric keys. We assume that the base station knows an upper bound on the (multihop) round trip time of the sensor network.

3. VERIFIABLE AGGREGATE SYNOPSIS

Background on synopsis diffusion. For predicate count, each sensor satisfying the predicate uses the well-known Flajolet-Martin's (FM) algorithm [3] to translate its reading to a *synopsis* with $O(\log n)$ bits, where exactly one bit is "1". The position of "1" bit is the ith position in the vector with probability of $(\frac{1}{2})^i$. The base station uses the final/aggregate synopsis, which is the bit-wise OR of all synopses from all the sensors, to approximate the count (see [3] for the approximation error). Sum can be done using an extended version of the FM algorithm, which simulates a reading v by conceptually combining v synopses. It is also possible to use other kinds of synopses, such as the exponential synopsis [4].

Verifiable aggregate synopsis. We use the FM synopsis as an example, even though our approach is quite general and applies to other kinds of synopses as well. First, we can easily make the synopsis from a sensor verifiable. Namely, every sensor uses a pseudo-random number generator, which is seeded using its own ID, to create the synopsis according to its reading v. It also creates a MAC (message authentication code) for the synopsis, using its unique key K shared with the base station. The base station (knowing ID, K and v) can always verify the synopsis. A compromised sensor can generate no more than m different synopses (corresponding to m different readings) that can pass verification.

The next conceptual step is that for every bit in the aggregate synopsis, the base station asks (via authenticated broadcast) whether any sensor would like to set that bit to "1". Sensors with the bit set in their synopses will locally broadcast their responses. A response contains the synopsis, the sensor's ID, the reading, and the MAC. All other sensors will record and then forward (locally broadcast) the first synopsis they receive and *drop all others*. This process has the *one-time forwarding* property where each sensor sends at most one message, which obviously is robust against multihop flooding. One can show that if any sensor sends a message, the base station is guaranteed to receive some message (i.e., impossible not to receive anything). If the base station receives within some timeout a synopsis that verifies, it sets that bit to "1". If the synopsis does not verify, the sensors will have recorded an audit trail leading to the origin of that corrupt synopsis.

Optimizing for adversary-free scenarios. It is possible to achieve $O(1)$ link congestion when there is no adversarial behavior, while still being able to detect adversarial behavior (but without an audit trail). The idea is to use the exponential synopsis [4] where the aggregate synopsis is the minimum of all synopses from sensors. We leverage the fact that only one sensor (instead of $O(\log n)$) is "accountable" for the aggregate synopsis. The first phase of the protocol is similar as the original synopsis diffusion protocol [5], with minor modifications to deal with DoS attacks during ring formation. In the second phase, the base station broadcasts the (verified) final synopsis and waits for disagreements. Any sensor with a smaller synopsis will send back a disagreement using the previous one-time forwarding protocol.

4. ACKNOWLEDGMENTS

I thank Phillip B. Gibbons and Suman Nath for several helpful discussions when this work was started.

5. REFERENCES

[1] H. Chan, A. Perrig, and D. Song. Secure hierarchical in-network aggregation for sensor networks. In *CCS*, 2006.

[2] J. Deng, R. Han, and S. Mishra. Limiting DoS Attacks During Multihop Data Delivery In Wireless Sensor Networks. *International Journal of Security and Networks, Special Issue on Security Issues in Sensor Networks*, 2006.

[3] P. Flajolet and G. Martin. Probabilistic Counting Algorithms for Database Applications. *Journal of Computer and System Sciences*, 1985.

[4] D. Mosk-Aoyama and D. Shah. Computing Separable Functions via Gossip. In *PODC*, 2006.

[5] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson. Synopsis Diffusion for Robust Aggregation in Sensor Networks. In *SenSys*, 2004.