On agent-based aggregation schemes in networks with malicious nodes

Christian Sommer
The University of Tokyo
National Institute of Informatics
2-1-2 Hitotsubashi
Chiyoda-ku
Tokyo, Japan
sommer@nii.ac.jp

Shinichi Honiden
The University of Tokyo
National Institute of Informatics
2-1-2 Hitotsubashi
Chiyoda-ku
Tokyo, Japan
honiden@nii.ac.jp

ABSTRACT

Mobile agents can be used to efficiently aggregate information in sensor networks. The originator aims at obtaining the correct result, even in the presence of byzantine nodes or a malicious aggregator agent. Secure information aggregation considers these situations. We propose to use a mobile agent for the hierarchical in-network aggregation scheme of Chan *et al.* and we introduce a greedy strategy to their commitment forest in order to reduce the communication cost of the aggregation process. This approach is promising in particular for a sequence of aggregations.

1. INTRODUCTION

An agent is a software entity that represents its originator to achieve a predefined goal. A mobile agent is able to migrate to multiple hosts and it can perform different sets of control operations for each host it visits, asynchronously from its originator. It is a process that can transport its state and data while still being able to perform appropriately in the new environment. Mobile agents can be used as a replacement for message-based communication in network management services. A mobile agent's ability to perform multiple remote operations can significantly reduce the amount of traffic, e.g., by aggregating network data before sending it back to its originator [6, 8, 11, 12, 13, 14].

The data retrieved from sensor networks might influence critical decisions. However, basic aggregation protocols are extremely vulnerable to attacks. Capturing of sensor nodes allows an attacker to manipulate the result [15]. Chan *et al.* [4, 5] proposes a framework for the design of secure aggregation protocols. The aggregator commits to the data it processes and later proves the correctness of the results to its originator using an interactive proof system. Aggregation preserving confidentiality is analysed in [7, 9]. The approach of [16] conceals sensor data and still provides efficient and flexible in-network data aggregation. [3] introduces an energy-efficient security algorithm and data aggregation protocol, which, in contrast to conventional data aggregation protocols, avoids the transmission of redundant data from the sensor nodes to the cluster-head. Host nodes need to be protected against malicious mobile code, however, at the same time, the agent needs to be protected against its host [10]. The work of Bellare and Yee [1, 2, 17] applies cryptographic primitives to mobile agents in order to protect its code and data. Forward secure signatures are used to protect partial results.

1.1 Contribution

We propose to use a mobile agent for the hierarchical aggregation scheme of Chan *et al.* [5] to reduce the communication cost. We lower the size of intermediate constructions (called commitment forests), which have to be forwarded within a binary aggregation tree. As long as the size of the mobile agent code is not too large, this results in improved communication cost.

2. SECURE AGGREGATION...

In-network aggregation is performed using a tree structure. Leaf nodes provide data only (i.e., a value v_j), inner nodes also perform aggregation computations. For inner nodes, an additional virtual leaf node provides the sensor value. As soon as the inner node has received all values of its children, it computes the aggregation and forwards it to its parent node. In the SUM algorithm, the complement $\bar{v}_j := r - v_j$ of its value v_j is stored at every node and aggregated, too[1]. An important part is the construction of commitment trees using a hash function H. In a commitment tree, vertices are labelled as follows:

$$u_i := \left\langle c_i, v_i := \sum v_j, \bar{v}_i := \sum \bar{v}_j, H[N|c_i|v_i|\bar{v}_i|u_j] \right\rangle$$

where c_i is the number of nodes 'below' and j iterates over all child nodes, i.e., every aggregator node commits to the labels of all its children including their values.

Chan *et al.* [5] proposes a delayed aggregation algorithm. This strategy trades off increased communication during the aggregation phase in return for a more balanced commitment tree, which results in lower verification overhead in the result-checking phase. An inner node only aggregates two values and forwards other values to its parent node,

[1] r is the upper bound on allowable data values.

which behaves in the same way. Therefore, a complete binary commitment forest is generated. For a sensor network with n nodes, there are at most $\log n$ commitment trees to be sent to a parent node, resulting in a communication cost of $O(\log n)$.

3. ...WITHIN A MOBILE AGENT

Using a mobile agent, we move the computation to the data. The aggregation algorithms presented follow a similar strategy, however, in order to lower the verification overhead in the result-checking phase, a communication overhead is accepted. We propose to employ a mobile agent in order to reduce this overhead. If the mobile code is not too big, the presented strategy is more efficient while the verification complexity remains the same.

Within the mobile agent the resulting subtrees can be combined immediately, minimising the size of the commitment forest. The agent, while travelling through the sensor network graph, has a 'global' state and is able to reduce the size right away, i.e., it maintains a virtual aggregation forest, which is optimized at every node. In [5], the ability of an aggregator node to reduce the forest's size strongly depends on the tree structure. Building the commitment forest using a regular process is both more predictable and more efficient. Furthermore, the agent's tour can be determined before in order to reduce node congestion[2]. However, the improved node congestion results in a slower computation because the agent has to take a long tour. If, as in [5], the performance measure is congestion only, the mobile agent approach is better, as both the maximum and the average communication load on any node is smaller or equal. If the regular binary tree structure is needed for fast verification, the maximum load of $O(\log n)$ remains, however, it rarely occurs and the average load is much smaller. Especially, if multiple sequential aggregations are needed[3], the agent's tours can be scheduled such that another node receives maximum load and thus the total network lifetime is increased.

4. CONCLUSION

By combining the mobile agent framework with the hierarchical aggregation scheme of Chan *et al.* [5] we reduce the communication cost and thus mitigate the tradeoff between fast verification and low communication cost. If the size of the mobile agent code is not too large, this results in an improved communication cost and thus a lower congestion.

5. REFERENCES

[1] M. Bellare and B. Yee. Forward integrity for secure audit logs. Technical report, Computer Science and Engineering Department, University of California at San Diego, November 1997.

[2] M. Bellare and B. S. Yee. Forward-security in private-key cryptography. In *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference*, pages 1–18, 2003.

[3] H. Çam, S. Özdemir, P. Nair, D. Muthuavinashiappan, and H. O. Sanli. Energy-efficient secure pattern based data aggregation for wireless sensor networks. *Computer Communications*, 29(4):446–455, 2006.

[4] H. Chan, A. Perrig, B. Przydatek, and D. Song. Sia: Secure information aggregation in sensor networks. *Journal of Computer Security*, 15(1):69–102, 2007.

[5] H. Chan, A. Perrig, and D. Song. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, pages 278–287, 2006.

[6] M. Chen, T. Kwon, Y. Yuan, and V. C. Leung. Mobile agent based wireless sensor networks. *Journal of Computers*, 1(1):14–21, 2006.

[7] L. Hu and D. Evans. Secure aggregation for wireless network. In *Symposium on Applications and the Internet Workshops (SAINT 2003)*, pages 384–394, 2003.

[8] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pages 457–458, 2002.

[9] P. Jadia and A. Mathuria. Efficient secure aggregation in sensor networks. In *High Performance Computing - HiPC 2004, 11th International Conference*, pages 40–49, 2004.

[10] W. Jansen and T. Karygiannis. Mobile agent security. NIST Special Publication 800-19, 1999.

[11] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for TSP, Steiner tree, and set cover. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 386–395, 2005.

[12] B. Krishnamachari, D. Estrin, and S. B. Wicker. The impact of data aggregation in wireless sensor networks. In *22nd International Conference on Distributed Computing Systems, Workshops (ICDCSW '02)*, pages 575–578, 2002.

[13] A. Manjhi, S. Nath, and P. B. Gibbons. Tributaries and deltas: Efficient and robust aggregation in sensor network streams. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 287–298, 2005.

[14] K. Tei, C. Sommer, Y. Fukazawa, S. Honiden, and P.-L. Garoche. Adaptive geographically bound mobile agents. In *Mobile Ad-hoc and Sensor Networks, Second International Conference*, pages 353–364, 2006.

[15] D. Wagner. Resilient aggregation in sensor networks. In *Proceedings of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks*, pages 78–87, 2004.

[16] D. Westhoff, J. Girão, and M. Acharya. Concealed data aggregation for reverse multicast traffic in sensor networks: Encryption, key distribution, and routing adaptation. *IEEE Trans. Mob. Comput.*, 5(10):1417–1431, 2006.

[17] B. S. Yee. Monotonicity and partial results protection for mobile agents. In *23rd International Conference on Distributed Computing Systems*, pages 582–591, 2003.

[2]Congestion is the maximum communication load on any node in the network. It measures how quickly the heaviest-loaded nodes will exhaust their batteries.

[3]A common scenario for sensor networks.