Secure Distributed Data Aggregation
Secure Distributed Data Aggregation

Haowen Chan
Carnegie Mellon University
USA
haowenchan@cmu.edu

Hsu-Chun Hsiao
CyLab/Carnegie Mellon University
USA
hchsiao@cmu.edu

Adrian Perrig
CyLab/Carnegie Mellon University
USA
perrig@cmu.edu

Dawn Song
University of California, Berkeley
USA
dawnsong@cs.berkeley.edu

Full text available at: http://dx.doi.org/10.1561/1900000025
Editorial Scope

Foundations and Trends® in Databases covers a breadth of topics relating to the management of large volumes of data. The journal targets the full scope of issues in data management, from theoretical foundations, to languages and modeling, to algorithms, system architecture, and applications. The list of topics below illustrates some of the intended coverage, though it is by no means exhaustive:

- Data Models and Query Languages
- Query Processing and Optimization
- Storage, Access Methods, and Indexing
- Transaction Management, Concurrency Control and Recovery
- Deductive Databases
- Parallel and Distributed Database Systems
- Database Design and Tuning
- Metadata Management
- Object Management
- Trigger Processing and Active Databases
- Data Mining and OLAP
- Approximate and Interactive Query Processing
- Data Warehousing
- Adaptive Query Processing
- Data Stream Management
- Search and Query Integration
- XML and Semi-Structured Data
- Web Services and Middleware
- Data Integration and Exchange
- Private and Secure Data Management
- Peer-to-Peer, Sensornet and Mobile Data Management
- Scientific and Spatial Data Management
- Data Brokering and Publish/Subscribe
- Data Cleaning and Information Extraction
- Probabilistic Data Management

Information for Librarians
Foundations and Trends® in Databases, 2010, Volume 3, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.
Secure Distributed Data Aggregation

Haowen Chan¹, Hsu-Chun Hsiao², Adrian Perrig³, and Dawn Song⁴

¹ Carnegie Mellon University, USA, haowenchan@cmu.edu
² CyLab/Carnegie Mellon University, USA, hchsiao@cmu.edu
³ CyLab/Carnegie Mellon University, USA, perrig@cmu.edu
⁴ University of California, Berkeley, USA, dawnsong@cs.berkeley.edu

Abstract

We present a survey of the various families of approaches to secure aggregation in distributed networks such as sensor networks. In our survey, we focus on the important algorithmic features of each approach, and provide an overview of a family of secure aggregation protocols which use resilient distributed estimation to retrieve an approximate query result that is guaranteed to be resistant against malicious tampering; we then cover a second family, the commitment-based techniques, in which the query result is exact but the chances of detecting malicious computation tampering is probabilistic. Finally, we describe a hash-tree based approach that can both give an exact query result and is fully resistant against malicious computation tampering.
Contents

1 Introduction 1

2 Problem Definition 5
2.1 Network Model 5
2.2 Security Infrastructure 6
2.3 Node Congestion 7
2.4 In-Network Aggregation 7
2.5 Threat Model 9
2.6 Input-Resilient Functions 11

3 Early Work on Secure Aggregation 13
3.1 One-hop Verification 13
3.2 Witness-based Verification 15
3.3 Summary and Discussion 16
3.4 References and Further Reading 17

4 Resilient Estimation 19
4.1 Authenticated Single Data Values 19
4.2 Verifiable Sketches 21
4.3 Set Sampling 24
4.4 Summary and Discussion 27
4.5 References and Further Reading 29
## 5 Commitment-based Techniques

5.1 The Aggregate-Commit-Prove Technique 31
5.2 SIA: Secure Information Aggregation 32
5.3 Hash Trees for Hierarchical Aggregators 35
5.4 SDAP: Probing the Hierarchical Hash Tree 37
5.5 SecureDAV: Distributed Verification 40
5.6 Secure Hierarchical In-Network Aggregation 40
5.7 Summary and Discussion 46
5.8 References and Further Reading 47
5.9 Conclusion 48

### References 49
Introduction

Recent advances in technology have made the application area of highly distributed data collection networks increasingly important. One example of this is sensor networks [39], which are wireless multihop networks composed of a large number of low cost, resource constrained nodes. Another example occurs in distributed systems such as distributed databases [34], peer-to-peer networks [58] or “grid” computing, where a large number of nodes are distributed over the Internet while engaging in some shared data collection or processing task.

A common task in these systems is the transmission of data towards a designated collection point, or sink. In sensor networks, the sink is typically a wireless base station that relays the collected data to an off-site server; in other distributed systems the sink may be a designated coordinating server that is responsible for archiving the data and answering user queries. The most straightforward method for collecting data is for each node in the network to send their raw data directly to the sink, via multi-hop routes in which intermediate nodes act as passive message forwarders and neither inspect nor modify the data. However, this approach is communication inefficient since not all of the collected data may be relevant or necessary for the application.
An alternative method for data collection is to observe that, commonly, only very simple aggregation functions are queried on the data. An aggregation function takes as inputs all the data values of the nodes in the network, but outputs only a single scalar. Examples of common aggregation functions include the sum of all data values; the count of the number of nodes fulfilling a given predicate, the minimum or maximum data value over the nodes in the network, and various measures such as mean and median. Since the result of computing an aggregation function is only a single value, this computation can be efficiently distributed in the network by having intermediate nodes compute sub-aggregates, leading to an extremely communication-efficient protocol. This technique is known as in-network aggregation \[39\] and is briefly described in Section 2.4.

The efficiency of in-network aggregation comes at a price to resilience, however, since it relies on the honest behavior of intermediate nodes in terms of computing accurate sub-aggregates. For example, for a sum computation, a malicious intermediate node with two children each reporting a data value of ‘1’, could report an inaccurate sub-aggregate value of ‘100’ instead of the correct value of ‘2’, thus skewing the final result by a large amount. Such attacks are not easily preventable since the efficiency of in-network aggregation relies on intermediate sub-aggregators reporting only concise summaries of their received values; since a large fraction of the input data is hidden by necessity, this exposes the network to greater opportunities for attack.

In this article, we provide an overview of the known approaches towards combating such malicious mis-aggregation attacks. Ideally, a secure aggregation protocol should offer three key features: it should (1) produce accurate answers (typically, an accuracy guarantee bounded by some function of the number of malicious nodes in the network), (2) require only low communication overhead, and (3) be resilient against general node compromise models. We present a brief summary of several approaches drawn from a selection of the current literature as well as a more in-depth tutorial in one of the more important frameworks. In our selection of covered literature, our goal is to provide the reader with a general intuitive understanding of the field, rather than to bring the reader exhaustively up to date with all algorithms.
for the area. Towards this end, we have opted towards a more tutorial approach in terms of selecting the publications that most clearly exemplify a certain class of approaches (or which have been most influential historically), rather than focusing on breadth or depth of coverage in terms of the most effective or the most recent algorithms.

The remainder of the article is organized as follows. In Section 2 we define the problem and introduce the notion of in-network aggregation more rigorously. In particular, in this article we focus only on aggregation computations for which the secure aggregation problem is feasible: the family of such functions is examined and defined. In Section 3 we highlight some earlier work which show the basic flavors of integrity verification and result checking for secure aggregation. The existing literature on secure aggregation can be broadly divided into two categories: the first category uses verifiable sampling to provide resilient probabilistic estimates of the aggregate result; the second category uses commitment verification, which, unlike the first category, can provide highly precise results for which any malicious tampering is immediately evident, but at the cost of availability. We cover these approaches in Sections 4 and 5 respectively.
References


Full text available at: http://dx.doi.org/10.1561/1900000025
References


References


