Towards Flexible and Secure Distributed Aggregation

Kristján Valur Jónsson, Mads F. Dam

Towards Flexible and Secure Distributed Aggregation

Kristján Valur Jónsson[12*] and Mads F. Dam[1**]

[1] Royal Institute of Technology (KTH), Stockholm, Sweden
(kristjan@kth.se, mfd@csc.kth.se)
[2] Reykjavik University, Iceland (kristjanvj@ru.is)

Abstract. Distributed aggregation algorithms are important in many present and future computing applications. However, after a decade of research, there are still numerous open questions regarding the security of this important class of algorithms. We intend to address some of these questions, mainly those regarding resilience against active attackers, whose aim is to compromise the integrity of the aggregate computation. Our work is currently in its initial stages, but we have identified promising research leads, which we present in this paper.

1 Introduction

The past decade has shown distributed algorithms to be a practical and scalable approach to a wide range of applications, as demonstrated by various peer-to-peer systems, e.g. BitTorrent and Skype. We are interested in *distributed aggregation algorithms*, which efficiently aggregate local measurements in an efficient and scalable manner by means of in-network processing. These protocols can be roughly categorized by families, the most prominent ones based on gossiping and spanning-tree overlays. Our research is driven by network management applications, where distributed aggregation algorithms have been shown to increase scalability and efficiency in monitoring and management systems [1].

Reliance on distributed algorithms, for monitoring of critical networked systems, motivates a thorough review of their security properties. We hope to contribute to this field of research, as will be outlined in this paper.

We are currently working on countermeasures against active insider adversaries, whose objective is to compromise the integrity of the in-network aggregate computation. This particular subject has received considerable attention in the past few years [2, 3, 4], most prominently in sensor networks research, but important issues still remain open. In general, the prior research has focused on patching specific aggregation protocols to increase resilience, whereas we hope to develop more broadly applicable methods, focusing on secure management of dynamic networked systems.

* This work is supported in part by grant #080520008 from Rannís, the Icelandic research fund, and funding by Reykjavik University.

** Work was partially supported by the EU FP7 project 4WARD and a personal grant from the Swedish Research Council.

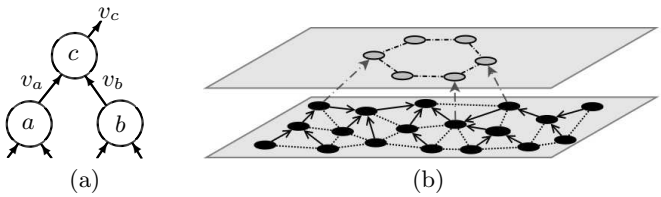

Fig. 1. (a) aggregation nodes, (b) Aggregation and supervisory layers.

2 A Motivating Example

Let us first present a small motivating example of a managed system of untrusted (compromisable) workstations, in which we require an aggregate view of some local input, perhaps for detection of anomalous events. We assume here a spanning-tree-based aggregation network, utilizing in-network aggregation by the managed nodes themselves for scalability.

A fraction of such a network is shown in Figure 1(a). Nodes a and b submit partial aggregate updates to their parent c, which is expected to correctly compute a update over its own state and received inputs, and forward a single aggregate message upwards. Now, consider the case in which c is compromised. The adversary may attempt to influence the aggregate computation by misrepresenting contributions, e.g. inflating or deflating the aggregate, manufacturing fictious inputs or ignoring legitimate ones. It is important to realize that the compromised node c does not risk detection of such actions in a typical unsecured network of the type described. It is obvious that a small number of compromised nodes, unrestrained against this class of attacks, may render an aggregation network ineffective, demonstrating the importance of countermeasures, such as the ones proposed in this paper.

3 Problem Statement

We consider means of increasing the resilience of dynamic distributed aggregation networks of untrusted nodes against inside attackers, whose objective is to stealthily compromise the in-network aggregate computation. Our goal is to ensure that, under suitable constraints on the network and distribution of adversaries, the network either performs the computation correctly, or else an adversary breaking the protocol is identified with some non-zero probability.

The problem has been considered in the distributed systems literature for the past decade, e.g. [2, 3, 4]. Significant results have been achieved, but the previous work generally involves considerable messaging overhead, and makes specific protocol and graph topology assumptions. The most voluminous body of work is on the popular research field of sensor networks, and addresses the particular challenges of these resource constrained networks.

In contrast to most previous work in the field, our objective is to develop protocols applicable to highly dynamic networks and a wide range of aggregation- and transport protocols in a network management context.

4 Distributed Security Layer

We approach the development of our solution in a methodical top-down fashion, employing sound systems design methodologies, with theoretical backing as warranted. The starting point is an idealized specification[1]. A well-known result from multi-party computing is that any distributed function can be computed securely via a trusted authority: every party simply submits its input to the trusted authority, which computes the function and hands the output back. This formulation can obviously be applied to our problem, which gives us the best case guarantees, as further defined by adversarial modeling and other systems assumptions. A trusted authority can indeed be implemented in form of a trusted server, accepting inputs from a population of managed nodes. This solution is the well-known centralized management model and suffers from obvious scalability problems. We intend to explore means of approaching the ideal functionality for a given set of assumptions, but to do so in a scalable and efficient manner.

We outline one possible approach to such an approximation in this paper: a distributed security layer $\mathcal{S}$ which accepts cryptographically secure commitments from the managed nodes and may interact with them to enforce a security policy. An example is shown in Figure 1(b). The objective is to remove or deduce the opportunities of compromised nodes to manipulate the aggregate computation undetected. Previous work on accountability systems [5] considers similar objectives. However, we plan to develop lower impact protocols, from the perspective of the managed nodes themselves. A practical approach to constructing $\mathcal{S}$ is to build on a secured distributed hash table (DHT).

A naive utilization of $\mathcal{S}$ is to require aggregation nodes to commit all messages sent and received. This method can be shown to be equivalent to the ideal specification, but the associated overhead is prohibitive. A more promising approach is to require smaller commitments and employing a spot-checking protocol in which $\mathcal{S}$ randomly selects nodes to audit – interrogating in some detail to ascertain that the node acted correctly in some past round. We believe this approach will result in an efficient protocol, tuneable to give acceptable detection probability of compromised nodes. We further believe that, in conjunction with robust reputation mechanisms, this method will prove to be an useful tool for increasing the robustness of in-network aggregation.

The approach described requires minimal changes to the basic aggregation system: we assume that each node has a unique and verifiable identity, as well as a set of cryptographic keys for signing or authenticating produced messages as

[1] We are inspired by Canetti's work on Universal composability and the ideal functionality modeling. However, we do not plan to apply this methodology literally at this time.

well as commitments sent to $\mathcal{S}$. Managed nodes must implement a small protocol to interact with $\mathcal{S}$, but little or no modifications to the aggregation protocol itself are required. A clear conceptual boundary can be drawn between the duties of the aggregation protocol and the security mechanisms. Spot checking behavior in past rounds also effectively decouples the security mechanisms from the graph structure, meaning that they are tolerant to churn and other dynamic effects.

Implementing efficient protocols for spot-checking requires distributing the secure storage duties into the population of untrusted nodes to some degree, e.g. using primitives for construction of a unmodifiable log [6]. This unavoidably blurs the boundaries between the trusted and untrusted overlays. Even further distribution of the security service into the untrusted node population may be considered, even to the extent of distributing $\mathcal{S}$ altogether into the general aggregation node population, in a similar manner to the proposals by [2, 4]. However, such approaches require stricter assumptions on the network type and topology, as well as considerable communications overhead on the aggregation nodes themselves.

5 Concluding remarks

Secure distributed aggregation has been our focus for the last few months and the work is still in its initial stages. Our focus is on the integrity of in-network aggregate computations and resilience against active insider attackers. Privacy is a closely related, but complimentary topic, which we may consider in parallel.

We have identified a promising research direction in the formulation of our distributed security service $\mathcal{S}$. Numerous variations and optimizations of this concept can be envisaged, and our proposed approach is sufficiently general to be applicable to a variety of distributed aggregation networks and security issues.

We plan to approach the design of such a system from a practical systems oriented angle, backed up by theoretical work as warranted. Next steps include a full design of $\mathcal{S}$ and formulation of commitment and checking protocols.

References

[1] Dam, M., Stadler, R.: A generic protocol for network state aggregation. In: RVK 05, Linköping, Sweden (2005)

[2] Chan, H., Perrig, A., Song, D.: Secure hierarchical in-network aggregation in sensor networks. In: CCS, New York, NY, USA, ACM (2006) 278–287

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

[4] Yang, Y., Wang, X., Zhu, S., Cao, G.: SDAP: A secure hop-by-hop data aggregation protocol for sensor networks. In: MobiHoc '06, New York, NY (2006) 356–367

[5] Haeberlen, A., Kouznetsov, P., Druschel, P.: PeerReview: Practical accountability for distributed systems. SIGOPS Oper. Syst. Rev. **41**(6) (2007) 175–188

[6] Chun, B.G., Maniatis, P., Shenker, S., Kubiatowicz, J.: Attested append-only memory: making adversaries stick to their word. SIGOPS Oper. Syst. Rev. **41**(6) (2007) 189–204