Brief Announcement: SuperTrust – A Secure and Efficient Framework for Handling Trust in Super-Peer Networks

Tassos Dimitriou
Athens Information Technology
19.5 km Markopulo Ave.
19002 Peania, Greece
tdim@ait.edu.gr

Ghassan Karame
Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland
karameg@inf.ethz.ch

Ioannis Christou
Athens Information Technology
19.5 km Markopulo Ave.
19002 Peania, Greece
ichr@ait.edu.gr

ABSTRACT

In this paper, we describe *SuperTrust*, a novel and efficient framework designed to handle trust relationships in Super-peer networks. What distinguishes SuperTrust from other works is that trust reports remain *encrypted* and are never opened during the submission or aggregation processes, thus guaranteeing privacy, anonymity, fairness, persistence and eligibility of transactions.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Data Sharing*; C.2.0 [**Computers-Communication Networks**]: General—*Security and protection*; C.2.4 [**Computers-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms

Algorithms, Design, Security

Keywords

P2P, Super-Peer, Trust, Security, Anonymity, Privacy

1. THE SUPERTRUST FRAMEWORK

SuperTrust is a *decentralized* framework that ensures the security of trust handling in K-redundant Super Peer networks and is in some sense *orthogonal* to the existing efforts for building trust among peers ([1, 2, 3, 4, 5]). Similar to most other contributions, SuperTrust assumes the existence of some certifying authority (CA) that can generate or certify special purpose keys and whose public key can be trusted as authentic.

Associated with each peer v in SuperTrust is a chosen set of n Super peers (*aggregators*) that are responsible for "collecting" the votes/reports of other peers that have interacted with v. The aggregators for each peer are chosen by the CA amongst the K super peers responsible for the various clusters. Furthermore, in each cluster, the CA delegates a *storage* node chosen amongst the K super peers to act as a storage facility for the reputations of the peers/resources located in the corresponding cluster (alternatively, this role can be assumed by the aggregators, thus eliminating single points of failure in the system). Such a semi-centralized, semi-distributed approach guarantees that each aggregator peer is within a fixed number of hops from each peer, thus improving the overall performance of the system. The various actions of a peer v in SuperTrust are outlined below:

Step 1. *Send a file request:* Peer v issues a request for resource r. Upon reception of v's request, one of the super peers responsible for v's cluster broadcasts this request to their neighbors.

Step 2. *Receive a list of peers that have the requested file, along with their global rating:* Upon reception of v's request, each super node checks whether the resource requested is within its cluster. Peer u issues a reply confirming his possession of the requested resource. In addition, each of n aggregators of u partially decrypt the encrypted trust value of u using a (t,n) Paillier-based threshold cryptosystem [6], and respond to v with their decrypted *shares* allowing v to compute the final trust value, as shown in Figure 1.

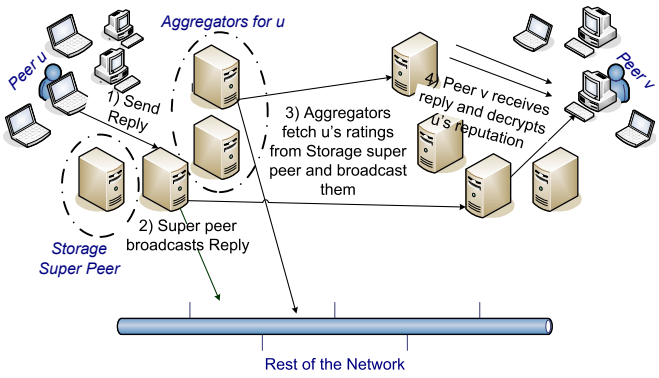

Figure 1: Reply Scenario in SuperTrust

Step 3. *Select a peer or a set of peers, based on a reputation metric in order to download the resource:* Once peer v receives the replies and the decrypted shares from *a sufficient number* t of aggregators, it calculates the global trust value of the replying peers and chooses to download the resource from the most reputable peer.

Step 4. *Send Vote:* Then, peer v rates the interaction it had with peer u. It first encrypts the report with the public key of the u's aggregators, thus encapsulating its rating for both peer u and its resource and then submits it to the designated Super peer. The latter forwards the encrypted vote to its neighbors. Upon reception of the trust report for v, the aggregators fetch from the storage Super peer v's

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

previous encrypted ratings, update it using a *homomorphic* encryption scheme based on [7], and submit the aggregation result back to the storage Super peer in order to guarantee the durability of the ratings in the system. In turn, the storage Super peer *only* stores the encrypted value that was advertised by the majority of the aggregators. Such a scheme protects against up to $n/2$ suspicious aggregators (where n is the total number of aggregators in some cluster) that are trying to cheat the system by submitting erroneous aggregation results. At this point, *the global trust value of v is updated by v' aggregators* without *decrypting the intermediate reports*, thus ensuring privacy and integrity of votes.

2. SUPERTRUST'S DESIGN GOALS

In this section, we present the first precise definition of security properties that *any* protocol for handling trust values must satisfy. We also show how our proposed model fulfills its various design goals.

1. *Anonymity, Privacy and Integrity of reports:* The system should support the peers' right to secrecy of their reporting ratings. Hence, peers should not lose their anonymity because their expressed opinions have been revealed, accidentally or not. In addition, an opinion expressed by some peer in the form of a report should be protected from disclosure and modification. A malicious node or a *collusion* of such nodes should not be able to eavesdrop or modify these ratings.

SuperTrust achieves the privacy property through the use of the threshold cryptosystem which guarantees that any $(t-1)$ faulty or malicious aggregators can not decrypt the report submitted by a peer. Additionally, the use of the homomorphic property to compute the final tally *without* decrypting individual reports strengthens peer's privacy and anonymity since at no point in the submission process will a report be decrypted. Furthermore, all communications in SuperTrust are made through secure channels since reports are sent encrypted using public key cryptography.

2. *Persistence:* All trust reports should be accounted in building the reputation of a peer even when these reporting peers are no longer in the network.

In SuperTrust, when a peer submits a trust report, its contents are accounted by the designated Storage super peer. If this peer ever leaves the system, its role will be assigned to a new super peer, thus guaranteeing maintenance of votes.

3. *Fairness and Soundness:* No one should be able to change, add, or delete trust reports without being discovered.

SuperTrust guarantees that a malicious node cannot affect the submission process by submitting invalid reports or by not following the protocol since it integrates a *proof of interaction* in the propagated reports. If a peer tries to submit an invalid report or if there is no associated proof of interaction, the report will be discarded by the system.

4. *Unreusability and Eligibility:* Only peers legitimate to express an opinion about some other peer should be able to do so. This protects from malicious nodes over-flooding the system with poor ratings for peers that have never interacted with.

SuperTrust ensures that only eligible peers are allowed to cast a report by including timestamps and proofs of interaction in the casted reports. This scheme guarantees that *authentic* reports can be submitted only once, thus preventing report duplication.

5. *Efficiency:* The entire process should be as efficient as possible in terms of messaging/computation overhead. This includes computation of the final trust value of a peer v, or the actual process of submitting a report by peer u.

In order to assess the performance of our scheme, we have simulated a realistic Super-peer network comprising of 1200 peers organized in 6 different clusters. As shown in Figures 2 and 3, SuperTrust takes advantage of the Super-peer model to optimize on messaging overhead and response time when compared to all other proposed contributions.

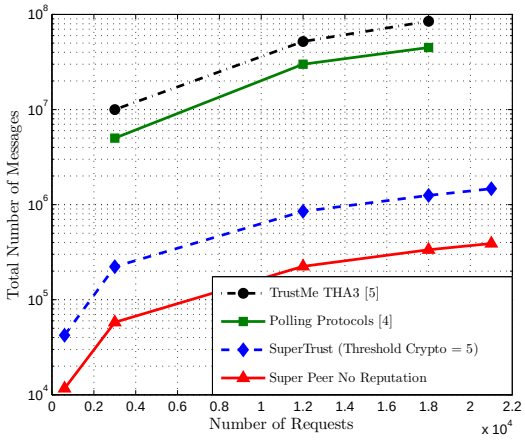

Figure 2: Messaging costs in SuperTrust.

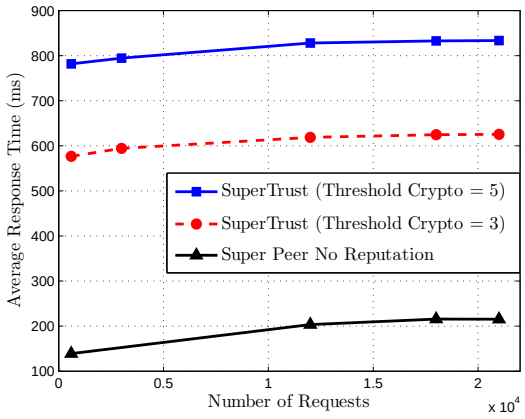

Figure 3: Response time in SuperTrust.

3. REFERENCES

[1] K. Aberer and Z. Despotovic, "Managing trust in a peer-to-peer information system," In *Proceedings of CIKM* 2001.

[2] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "Eigenrep: Reputation management in p2p networks," In *Proceedings of WWW* 2003.

[3] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities", In *IEEE Transactions on Knowledge and Data Engineering*, 2004.

[4] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," In *Proceedings of WWW* 2002.

[5] A. Singh and L. Liu, "TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems," In *Proceedings of P2P* 2003.

[6] I. Damgård and M. Jurik, "A Generalization, a Simplification and some Applications of Paillier's Probabilistic Public-Key System," In *PKC* 2001.

[7] P. Paillier, "Public-Key Cryptosystems Based on Discrete Logarithm Residues," In *Eurocrypt* 1999.