Beyond "web of trust": Enabling P2P E-commerce*

Anwitaman Datta, Manfred Hauswirth, Karl Aberer
Distributed Information Systems Lab
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
{anwitaman.datta, manfred.hauswirth, karl.aberer}@epfl.ch

Abstract

The huge success of eBay has proven the demand for customer-to-customer (C2C) electronic commerce. eBay is a centralized infrastructure with all its scalability problems (network bandwidth, server load, availability, etc.). In this paper we argue that C2C e-commerce is an application domain that maps naturally onto the emergent field of P2P systems simply by its underlying interaction model of customers, i.e., peers. This offers the opportunity to take P2P systems beyond mere file sharing systems into interesting new application domains. The long-term goal would be to design a fully functional decentralized system which resembles eBay without eBay's dedicated, centralized infrastructure. Since security (authenticity, non-repudiation, trust, etc.) is key to any e-commerce infrastructure, our envisioned P2P e-commerce platform has to address this adequately. As the first step in this direction we present an approach for a completely decentralized P2P public key infrastructure (PKI) which can serve as the basis for higher-level security service. In contrast to other systems in this area, such as PGP which uses a "web of trust" concept, we use a statistical approach which allows us to provide an analytical model with provable guarantees, and quantify the behavior and specific properties of the PKI. To justify our claims we provide a first-order analysis and discuss its resilience against various known threats and attack scenarios. In support of our belief that C2C E-commerce is one of the potential killer applications of the emerging structured P2P systems, we provide a layered model for P2P E-commerce, demonstrating the dependencies of various security related issues that can be built on top of a decentralized PKI.

Keywords: Customer-to-customer e-commerce, structured peer-to-peer systems, public key infrastructure, web of trust, quorum, distributed denial of service (DDOS) attacks.

*The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

1 Introduction

The demand for customer-to-customer commerce (C2C) has been proven by the huge success of eBay [11]. eBay provides a centralized trading platform to its customers which offers a certain degree of security that business transactions between partners that do not know each other are performed in a proper way, i.e., each partner obeys the rules. So, why should we not stay with this architecture? The advantage that eBay, like any other centralized system, can enforce rules easily turns into a severe problem if we switch the viewpoint to scalability. In each centralized system the center is a "hot spot" in terms of failure (no server, no system), network bandwidth, and server load among others.

In this paper we argue that a customer-to-customer system would actually map more naturally onto a P2P system by its very structure and interaction pattern. However, more effort has to be put into carefully designing and offering similar services and guarantees as a centralized infrastructure. Scalability can be achieved well as proven by successful P2P infrastructures such as Kazaa [16], Gnutella [9], or Freenet [8].

Equally important are the security aspects and service guarantees which are more difficult to achieve in a distributed environment, such as authentication of identities of the trading partners. Most of these services rely heavily on the existence of a public key infrastructure (PKI). Though PKIs exist for quite some time now and can be considered "out-of-the box systems" their centralized architecture contradicts the P2P approach and would introduce centralization again through the back door. So the only alternative so far would be the application of a PGP-like "web of trust" approach. However, it has been shown that this concept has several severe shortcomings as we will discuss in Section 2. Thus we follow a different strategy and propose a decentralized PKI based on a statistical (quorum-based) approach that bypasses these problems.

The paper starts with a taxonomy of existing PKI approaches in Section 2 which also gives an overview of the pros and cons of the different classes. Section 3 then presents a detailed description of the decentralized PKI architecture we propose by discussing all its building blocks and algorithms. To justify the validity of our model we give a first-order analysis of the incurred effort and security properties of our PKI in Section 4. We continue with a security analysis of our PKI in Section 5 in which we analyze common attacks in every stage of the systems lifetime. In Section 6 we provide a layered model for P2P E-commerce, demonstrating the dependencies of various security related features that can be built on top of a decentralized PKI. Related work is discussed in Section 7, which is relatively sparse in the relevant domain, due to the pioneering character of our work. Our conclusions in Section 8 round out the paper.

2 A case for harnessing structured P2P systems

This section defines an informal taxonomy to classify existing approaches and to position our proposed system therein. We argue for a distributed PKI based on efficient P2P access structures rather than using the web-of-trust model which has several drawbacks including inefficiency and lack of any proper model to provide quantifiable probabilistic guarantees.

2.1 Taxonomy

Essentially there are two dimensions in decentralized PKI management, namely "the discovery of peers who have the public key" and "trust on the peers from whom the public key is obtained". The fundamental difference between the two approaches (web of trust/statistical) is the order in which the two dimensions are navigated, i.e., the mechanism by which the relevant information is obtained: random walks in a trust graph or systematic, efficient access to relevant information, and then using a quorum (possibly weighted) for reliability(trust).

The inefficiency in web of trust based approaches arises primarily from the fact that the information is searched using random walks in the network. While random walks is the best one can do in structure-less P2P systems like Gnutella, with the emergence of structured P2P systems that support efficient searches, inserts and updates, we can now exploit these features and realize efficient distributed P2P PKI using statistical methods (or hybrid methods where the information is still searched efficiently using a structured P2P system, and not using the web of trust like transitivity). The reason a quorum based approach in structure-less P2P systems is impractical is that searches still depend on flooding (and thus continue to be inefficient), and also maintenance (inserts or updates) has high overhead.

Thus in the context of data management, current PKIs may be classified in two main groups:

Centralized: Confederation of trusted third parties (TTP), so-called certification authorities (CA), for example, VeriSign. The TTPs do not participate in the interactions of a system but act as facilitators of the activities. We omit these centralized solutions from the rest of our discussion since they are not in the scope of our discussion which focuses on only decentralized systems.

Decentralized: The public key infrastructure is maintained by the participants themselves without using central control and special roles such as CAs. Three main subclasses of this approach can be identified.

Web of trust: In this model, a participant (peer) of the system knows the public keys of some other peers, and considers this knowledge sacrosanct. It also relies on some of

these peers (with possibly varying degree of trust) to certify the public key of other peers. Thus the knowledge of peers' public key is obtained by finding a path in the peer acquaintance(trust) graph, thus forming a 'web of trust', where if P_A trusts that K_B is P_B's public key, and also relies (personally determined) on P_B to certify a third party's public key, then P_A will believe in K_C being P_C's public key if P_B certifies it. PGP [13] and variants belong to this group. PGP implicitly exploits the small world phenomenon of social acquaintance that is observed in the trust(certificate) graph [5] to create a web of trust of peers' public keys. Consequently, it obliterates the need of central authorities, and has been enormously successful as a freely available decentralized public key infrastructure. However, the strength of a chain is determined by the weakest link, and hence a simple transitive trust is highly vulnerable, and thus unreliable.

Statistical (quorum based) approaches: A statistical approach would involve obtaining the public key information from many peers and then forming a quorum. This will be elaborated in Section 3.2 where we describe our approach for P-Grid [1, 4]. The essential idea behind a statistical approach is to have multiple random and thus presumably independent peers to replicate the public key information, and retrieve the information from a random subset of these replicas. Such an approach relies on an efficient, decentralized storage infrastructure, for which we will use P-Grid, our P2P storage management system.

In this paper we restrict our discussion to the purely statistical retrieval and insertion of public keys in P-Grid (a hybrid variant will use a weighted quorum). Also, it may be emphasized that any other P2P system with similar guarantees of efficient and reliable search and updates as that of P-Grid may be used instead, and thus our proposal is generic and could be adapted to a group of structured P2P systems.

Hybrid approaches: A hybrid approach will involve obtaining public key information from many peers, and then forming a weighted quorum dependent on one's relative trust on various peers from whom the information is obtained.

An extension of the original PGP approach, which is presently put into practice, is to include multiple paths of trust transitivity [18] in an effort to improve reliability of authentication. Nevertheless, the reliability of such approaches is limited because of intersecting paths, and thus needs authentication metrics [18] to quantify the reliability of such multiple paths. Thus it consumes a lot of network and computational resources, worsened by the fact that there is no guarantee of the existence of such multiple independent paths, or a mechanism to discover them efficiently. It is again an attempt to navigate the two dimensions as described in the beginning of this section in a wrong order, since random walks are used for information discovery with the assumption of

finding multiple paths, trying to use the power-law distribution of trust graphs. Further, both the multiple paths and the metrics need to be evaluated at each peer, and thus the effort is not shared.

Apart from the inefficiency because of random walks, web of trust based approaches have further drawbacks:

1. Path discovery is inefficient because effort is not shared, and has high, unbounded latency.
2. Web of trust approaches (e.g., PGP) have primarily been used for privacy purposes. Typically, to completely believe in a person's public key, the information (at least the fingerprint of the public key) has to be obtained offline. Otherwise certification provided by only persons known in real life are accepted. It is thus premature to assume, particularly in the absence of any quantifiable guarantee, that the web of trust model that worked well for providing privacy for email users will translate well into a public key infrastructure in a P2P system.
3. Web of trust models fail to use the collective knowledge of the whole society, but uses only information available within a small number of transitive hops, determined by time to live, in the connectivity graph. However, the ultimate purpose of a decentralized PKI should be to provide a way to establish identity of stranger peers beyond a reasonable doubt. Web of trust models can not guarantee that (since transitive paths are not guaranteed). This is another important reason why we need to use efficient access structures to store public keys, and use a quorum based approach to reliably find public keys.
4. Finally, since the trust on other peers' certification is essentially ad-hoc, web of trust is susceptible to the treachery of even one (or a very few) trusted peers.

We argue that statistical (and hybrid) approaches are feasible and indeed better suited for systems where information can reliably and efficiently be obtained and updated, as is the case with many emerging structured P2P systems.

3 A quorum based decentralized PKI

As mentioned in the previous section, managing a PKI in a P2P way needs an efficient and reliable distributed information access structure, and also effective functionalities like updates of replicas even in the presence of frequent disconnections and possibly uncooperative peers. This is not possible in unstructured P2P systems like Gnutella [9], that is why the "web of trust" model, which essentially depends on random walks (basically flooding) for exploring the trust graph of a P2P network has gained more popularity. But with the recent development of efficient access

structures like CAN, Chord, Freenet and P-Grid among others, it is indeed possible to realize more systematic models, rather than relying on the ad-hoc web of trust model, for which no probabilistic guarantees have been provided so far. In this section, we first give a brief introduction of P-Grid [4], before elaborating our P-Grid based PKI.

3.1 P-Grid

P-Grid [1, 4] is a peer-to-peer lookup system based on a virtual distributed search tree: Each peer only holds part of the overall tree, which comes into existence only through the cooperation of individual peers. Searching in P-Grid is efficient and fast even for unbalanced trees [2] ($O(\log(n))$, where n is the number of leaves). Unlike many other peer-to-peer systems P-Grid is a truly decentralized system which does not require central coordination or knowledge. It is based purely on randomized algorithms and interactions. Also we assume peers to fail frequently and be online with a very low probability. Figure 1 shows a simple P-Grid.

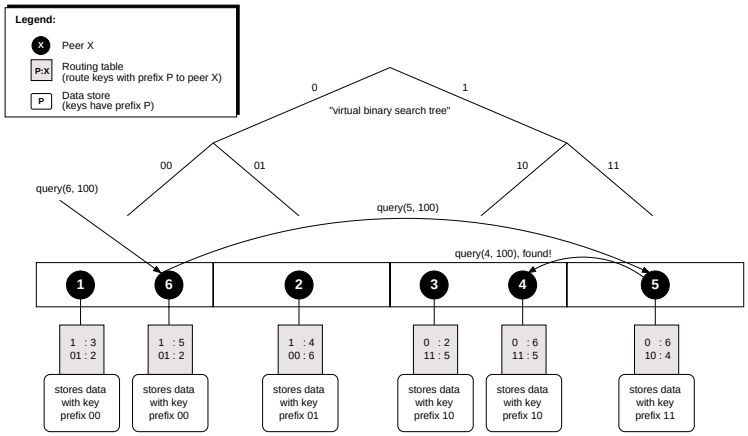

Figure 1: Example P-Grid

Every participating peer's position is determined by its path, that is, the binary bit string representing the subset of the tree's overall information that the peer is responsible for. For example, the path of Peer 4 in Figure 1 is 10, so it stores all data items whose keys begin with 10. For fault-tolerance multiple peers can be responsible for the same path, for example, Peer 1 and Peer 6. P-Grid's query routing approach is simple but efficient: For each bit in its path, a peer stores a reference to at least one other peer that is responsible for the other side of the binary tree at that level. Thus, if a peer receives a binary query string it cannot satisfy, it must forward the query to a peer that is "closer" to the result. In Figure 1, Peer 1 forwards queries starting with 1 to Peer 3, which is in Peer 1's routing table and whose path starts with 1. Peer 3 can either satisfy the query or forward it to another peer, depending on the next bits of the query. If Peer 1 gets a query starting with 0, and the next bit of the query is also 0, it is responsible for the query. If the

next bit is 1, however, Peer 1 will check its routing table and forward the query to Peer 2, whose path starts with 01.

The P-Grid construction algorithm [4] guarantees that peer routing tables always provide at least one path from any peer receiving a request to one of the peers holding a replica so that any query can be satisfied regardless of the peer queried. Additionally it guarantees that a sufficient number of replicas exist for any path and that the peers representing a certain path also know their replicas. Thus the routing tables will hold also multiple references for each level which the routing algorithm selects randomly [4].

Also, P-Grid, unlike most contemporary P2P systems, supports updates of the stored, replicated data via a push/pull strategy with probabilistic success and consistency guarantees in an unreliable environment [10].

3.2 The P-Grid based PKI

Each peer p is uniquely identified by a universally unique identifier (UUID) Id_p. This identifier is generated once when a peer joins the P-Grid community, by applying a cryptographicly secure hash function to the concatenated values of the current date and time, the current IP address $addr_p$ and a large random number. At bootstrap, each peer p also generates a private/public key pair D_p/E_p.

In P-Grid, routing tables and the index hold only these identifiers. Each peer p additionally has a cache of "identity to physical address" mappings $(Id_i, addr_i, TS_i)$ (TS_i denotes a time-stamp) that it already knows, in order to be able to communicate with other peers. Since disconnections of peers may lead to changing IP address, peers must update their latest "identity to physical address" mapping in P-Grid. The update functionality is provided in P-Grid as described in [10].

To correctly identify a peer it is essential to detect old mappings and retrieve and cache up-to-date ones.

Our algorithm for building a decentralized PKI on top of P-Grid is given below.

Bootstrap

Bootstrap is the phase when a new peer p joins the P-Grid.

1. p determines its current IP address. The IP address must be routable and reachable, i.e., not behind a firewall. The IP address is inserted in the P-Grid in order to handle possible changes of physical address of peers reconnecting after staying offline, and has been described and analyzed in [12].

2. p generates Id_p, D_p/E_p.

3. p inserts the *tuple* $(Id_p, addr_p, E_p, TS_p, D_p(Id_p, addr_p, E_p, TS_p))$ into P-Grid using Id_p as the key (TS_p prevents replay attacks). Inserting in P-Grid means that the request is routed to a peer $R_i \in \Re_p$. $\Re_p$ is the set of replicas responsible for the path using Id_p as the key value ($path(Id_p)$). If Id_p already exists in the P-Grid (though this is very unlikely) p is notified. If so, p generates a new Id_p and repeats this step.

4. p repeats the insertion operation at R_{min1} random and distinct P-Grid peers, so that the insertion request reaches an expected R_{min2} distinct replicas.

5. All R_i that receive the $insert(tuple)$ message initiate $update(tuple, R_i)$ among their replicas $\Re_p$. All replicas, including the ones that originated such updates locally store the tuple only if it receives and forms a quorum of $R_{min3} \leq R_{min2}$ distinct such update messages within a T_{out1} time.

   Peers who received the original insert then send a confirmation to p. This of course holds for the peers/replicas that are online during the update operation. Those peers that come online later use a quorum based pull (anti-entropy) to get a current view as described in [10]. If after T_{out1} since receiving the first update message, R_{min3} distinct messages have not been received, the peers discard the information.

   In the absence of Byzantine/malicious peers it would have been sufficient to make a single insert in P-Grid, since the update mechanism would have updated all replicas. However, malicious peers may initiate updates with false information. Since search and insert requests are routed to random replicas, we use multiple requests and then a quorum to address this properly.

6. As a result of the previous steps the mapping will be physically stored at peers in $\Re_p$. Based on the randomized algorithms that P-Grid uses we can assume that the individual replicas $R_i \in \Re_p$ are independent and they collude or behave Byzantine only to a degree that can be handled by existing algorithms.

7. If p receives $R_{min4} \leq R_{min3}$ confirmations (within some $T_{out2} > T_{out1}$), it is convinced (probabilistically) that its public key has been replicated amply for fault tolerance. Otherwise p generates a new Id_p and repeats the previous steps.

   Since only a new peer entering the P2P system needs to conduct the bootstrap phase, it is irrelevant which identity is successfully inserted. Also, generation and re-insertion of a new identity will be required only in the event of a distributed denial of service attack by malicious peers, more on which is discussed in Section 5.

$R_{min4} \leq R_{min3} \leq R_{min2} \leq R_{min1}$, and the exact numbers is a design issue for the P2P system. We will use these for our preliminary analysis in Section 4 of the effort required and performance (probabilistic) offered by our system. For simplicity we will consider $R_{min4} = R_{min3}$ without loss of generality of the results of our analysis.

Peer startup

Whenever a peer p rejoins the P-Grid it performs the following step.

1. p starts up and checks whether its $addr_p$ has changed. If yes, it initiates an update of its new physical address (signed with its private key). The complete algorithm for update along with the cost incurred and its reliability can be found in [12]. This step is necessary in order to make sure that the routing tables are correct.

Operation phase

This phase denotes the standard operation, i.e., p is up and running, has registered an up-to-date mapping of its identity/physical address $(Id_p, addr_p, TS_p)$ and is ready to process queries and update requests. Both queries and updates need to be routed to at least one replica peer responsible for the concerned key space. The following steps are to route it successfully despite frequent peer disconnections and changes in peers' physical address resulting in temporarily inconsistent routing tables.

By establishing the correct mapping, we ensure that operations (query/insert/update) may be successfully carried out. Then, such operations pertaining to either peers' public key, current physical address, reputation or any other kind of information may be conducted in a reliable manner. The steps incurred are given below.

1. p receives a request Q from a peer q.
2. In case p can satisfy Q the result is returned to q. Otherwise p finds out which peer p_f to forward the query to. It checks its routing table and retrieves $(Id_{p_f}, addr_{p_f}, E_{p_f}, TS_{p_f})$ which had been entered during the construction of P-Grid.
3. p generates a random number r, contacts p_f and sends $E_{p_f}(r)$. As an answer p_f must send $(D_{p_f}(E_{p_f}(r)))$ and q can check whether $D_{p_f}(E_{p_f}(r)) = r$. If yes, p_f is correctly identified, i.e., p really talks to the peer it intends to, and Q is forwarded to p_f.

4. If not, then p_f has a new IP address (the case that somebody tries to impersonate p_f is covered implicitly by the signature check above) and p sends a query to P-Grid to retrieve the current $addr_{p_f}$ using Id_{p_f} as the key.

   Since p_f may be offline multiple routing entries for each level are maintained to offer alternative peers to route to.

5. p collects all answers $t_i = (Id_{p_f}, E_{p_f}, addr_{p_f}, TS_{p_f}, D_{p_f}(Id_{p_f}, E_{p_f}, addr_{p_f}, TS_{p_f}))$ it receives from the $R_j \in \Re_{p_f}$

   If extended security is required then the R_j should sign their answers, i.e., send $(t_i, D_{R_j}(t_i)))$. p has to collect at least R_{min3} answers to detect misinformed or malicious peers, i.e., checks whether a certain quorum of the answers is identical (R_{min3} is defined by each individual p according to its local requirements for trustworthiness of the reply). Otherwise the query is repeated a certain number of times before aborting.

   (a) As an optimization the quorum can be avoided under certain circumstances. If p already knows E_{p_f}, e.g., from the construction of the P-Grid or because it has already done a certain number of (quorum-based) queries for E_{p_f} that have resulted in identical answers, so that it can assume that its E_{p_f}, then it can immediately check the validity of the answer by $E_{p_f}(D_{p_f}(Id_{p_f}, E_{p_f}, addr_{p_f}, TS_{p_f})).Id_{p_f} = t_i.D_{p_f}$.

   (b) The scheme can be further optimized (and made more robust and secure) by having all peers store the E_p's that they receive.

6. Now p can proceed with step 3. In case this is successful p enters $(Id_{p_f}, addr_{p_f}, E_{p_f}, TS_{p_f}))$ in its local cache.

Following the above steps, a peer p can obtain the latest physical address of other peers in a recursive manner and thus successfully handle the basic P-Grid operations of query, insert and update. As mentioned above, any information, including public key and reputation related information may then be accessed and maintained similarly in an efficient and reliable manner. A P-Grid based PKI is efficient because the basic operations such as search or insert in P-Grid take $O(log(n))$ messages to discover one random replica responsible for the relevant key. Since the routing process is randomized, reliability of results is then obtained by using quorum based techniques.

4 Analysis

This section gives a preliminary analysis of the cost of locating public key information reliably in P-Grid, that is to locate R_{min2} distinct replicas responsible for the particular key, and forming a

quorum of at least R_{min3}.

For the analysis, we use the following notation: There are R_{tot} replicas and R_{on} of these are online with their correct physical address references known to other peers who refer to them. All requests need to be routed to one of these responsible replicas. The effort to route an individual request in presence of stale caches and unavailable peers has been analyzed and shown to be efficient in [12], and update propagation within the subnetwork of replicas using a hybrid push/pull approach also has been shown to provide probabilistic success and consistency guarantees in presence of peer disconnections at a reasonable overhead in [10].

Thus, here we only need to analyze the expected number of independent requests R_{min1} in order to reach R_{min2} distinct replicas and form a R_{min3} quorum, where $R_{min3} \leq R_{min2} \leq R_{min1}$ as discussed in Section 3.2, out of a possible R_{on} replicas online, the underlying assumption being $R_{on} \geq R_{min2}$.

For the analysis, we assume that the network topology is relatively static, such that during the whole period while queries are being propagated, R_{on} does not change drastically. This is a realistic assumption because queries in P-Grid effectively means moving through a distributed binary search tree, requiring $O(log(n))$ messages, which means that a query is complete within a very short period of time. Thus the network topology may indeed be assumed to be quasi-static. Further, apart from replicas going off-line, replicas come online as well, thus the variation of R_{on} within a small period of time can be neglected for a first order analysis. Under these assumptions, we need to determine the expected number of requests at random peers in P-Grid, such that response is obtained from R_{min2} distinct replicas out of the R_{on} replicas online. This problem may be reduced to coupon collector's problem [15], such that R_{min2} coupons need to be collected out of R_{on} possible coupons. Under the assumption that all the online replicas are equally likely to be reached, which is guaranteed by P-Grid's randomized load balancing construction and routing algorithms, the expected number of requests R_{min1} is then a function of R_{on} and R_{min2}, represented as $R_{min1}(R_{on}, R_{min2})$ required for R_{min2} distinct responses, and can be formalized as (coupon collector's problem [15]):

$$R_{min1}(R_{on}, R_{min2}) \quad = \quad R_{on}(HarmonicNumber(R_{on}) - HarmonicNumber(R_{on} - R_{min2}))$$

Figure 2 shows the total expected number of P-Grid requests R_{min1} required to reach R_{min2} distinct replicas out of R_{on} online replicas. In this example R_{on} is chosen to be 20, and R_{min2} is varied between 0 and 20.

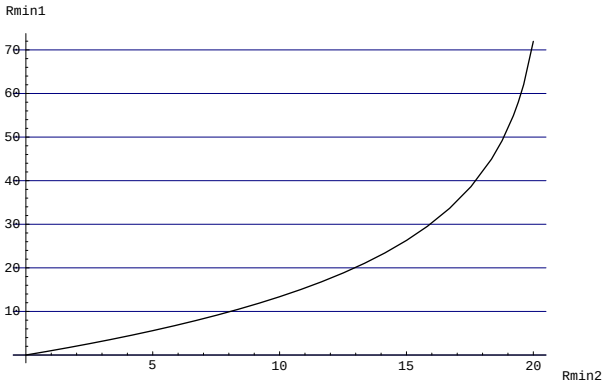

Figure 2: Expected effort(R_{min1}) for contacting R_{min2} distinct replicas.

Given any probability m of individual peers being malicious, the probability of successfully forming a quorum (and thus successful authentication), and the probability of a successful DDOS attack (if malicious peers act independently, where each of them return a false information, or do not reply at all) or successfully persuading an enquirer with a false information (with all malicious peers collaborating together, and thus replying the same false information) may be quantified, as given below.

Probability of correct authentication:

$$P_{Correct-Quorum} = \sum_{i=R_{min3}}^{R_{min2}} \binom{R_{min2}}{i} (1-m)^i m^{R_{min2}-i}$$

This is the probability that a quorum of at least R_{min3} correct replies are obtained.

Probability of successful attacks (as elaborated above):

$$P_{Attack} = 1 - P_{Correct-Quorum} = \sum_{i=0}^{R_{min3}-1} \binom{R_{min2}}{i} (1-m)^i m^{R_{min2}-i}$$

This is the case where sufficient correct replies could not be obtained to form an appropriate R_{min3} quorum.

Since peers are assigned a key space in a completely randomized manner in P-Grid's construction algorithm, it is unlikely that all malicious replicas collaborate, and thus DDOS attacks are more likely than malicious peers being able to persuade an enquirer with a false information successfully.

Below, in Figure 3 we show probability of successfully forming a quorum of at least 11 matching replies where 20 replicas are online, and thus can possibly be contacted. The X-axis represents the percentage of malicious peers. The probability of a successful DDOS attack is complementary to the probability of forming a quorum. As can be seen in the figure, there is a phase transition,

such that for a low percentage of malicious peers (in this example, if $m \leq .25$), the probability of successfully forming a quorum is close to one, and with increasing percentage of malicious peers, performance degrades rapidly, such that susceptibility to attacks increases. This example demonstrates that our mechanism provides a quantifiable performance for providing authentication beyond reasonable doubt in a predominantly well-behaving P2P society.

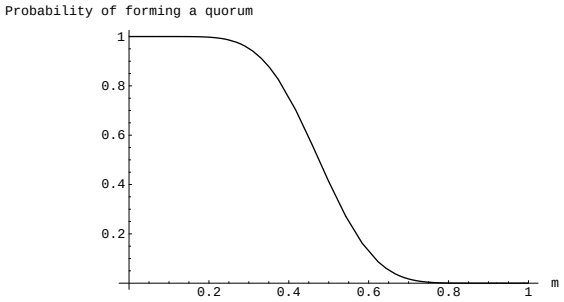

Figure 3: Probability of successfully forming a quorum with varying percentage m of malicious peers.

5 Attacks

Bootstrap phase

The most common attack during the bootstrap phase will be that a malicious replica inserts a wrong tuple locally instead of inserting the correct tuple, and sends an update message using this false tuple. By requiring a quorum at all well-behaved peers (replicas) we ensure that in a predominantly well-behaving P2P society, the correct information is stored.

A variant of this attack (where the percentage of malicious users is higher) will lead to a situation where well-behaved peers will possibly fail to form a quorum, and then after timeout, the operation has to be aborted and repeated with a freshly generated identity. Since routing is randomized, it is however likely that several repeated attempts will lead to eventual success.

The worst case will be a group of malicious users collaborating together will carry on a successful distributed denial of service attack (DDOS), i.e., they will successfully insert a wrong public key into the P-Grid on behalf of a peer p's identity. In this case, the peer p actually has nothing to loose (since it is new, it has no reputation yet), and will simply have to abandon that identity, and restart with a new one, or maybe, it will be unable to join the system. Philosophically speaking, it is not a bad option either, since most members of the system have to be malicious and collaborating in this case, so joining may not be a good idea anyway.

Peer startup phase

If a peer has successfully registered itself, then it has its correct public key registered in P-Grid. When a peer p rejoins the network, it has to communicate with other peers, and p's identity will be authenticated using its public key. Here, a possible attack can be done if a peer p_v queries P-Grid to verify p's public key. Malicious peers may provide false information, thus trying to deny service. But with multiple queries starting at random peers (e.g., from peer p_v's routing table), this attack can be thwarted, particularly because queries are routed randomly to different replicas. Thus a quorum-based authentication will again work in a predominantly well-behaved P2P society.

Operational phase

During the operational phase, a peer p needs to first authenticate the identity of another peer p_c by conducting queries about p_c's public key. After authenticating the identity, queries related to reputation information can again be made in a similar fashion. Also, after concluding business, digital signatures may be used as proof, thus providing non-repudiation. Apart from attacks during insert or query operations, as discussed above, impersonation attacks may also be attempted.

Impersonation is prevented using public key based authentication, but depends on the percentage of malicious peers in the system. If a significant population of peers are malicious and they all collaborate together, impersonation can not be ruled out, as was discussed in the analysis (Section 4).

6 Enabling P2P E-commerce

In order to enable P2P e-commerce it is essential to provide security functionalities, many of which can only be realized if a PKI is available, for example: authentication, confidentiality and trust. Figure 4 shows the necessary functionalities which all rely on a PKI in the context of our envisioned mc

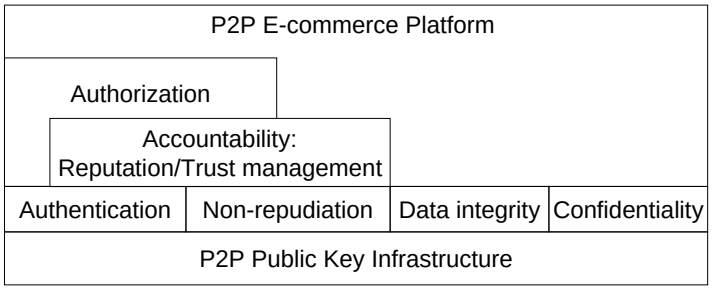

Figure 4: Layered model for P2P E-commerce

Authentication: Verification of the identity of a participant. Authentication of an entity's identity is typically done using digital signature, which uses a public key infrastructure (PKI). While other means for authentication exist, for example username/password schemes, we exclude such approaches in our discussions because of their centralized architecture and consequent incompatibility with the P2P approach.

Non-repudiation: To provide undeniable proof of any operation conducted by an entity, it is again necessary to apply digital signatures.

Accountability: Past actions of participants need to be taken into account in the present, thus penalizing for past misbehaviour or rewarding for past compliance with the (possibly implicit) rules. This is done using reputation management of peers. A P2P trust model is provided in [3]. For this, it is essential to have reliable authentication and non-repudiation schemes, which themselves rely on a reliable PKI.

Authorization Participants may decide to authorize other participants to use certain resources after having authenticated their identity, and possibly after making a judgement on their trustworthiness.

Confidentiality and data integrity: Participants in an activity may require confidentiality out of privacy concerns or for preserving digital rights. Message digesting and digital signatures can be employed to prevent data corruption. These in turn again rely on the existence of a PKI.

These basic security services are needed to implement any E-commerce platform (on top of a peer-to-peer network). In this section we have shown a layered model to integrate the PKI with other services like reputation management, work on which has previously been done, where peer reputation is also managed in P-Grid [3]. Distributed access structures such as P-Grid not only require authentication of peers for reliability, but also can serve as a new means for authentication, and provide maintenance of other resources and services like trust (reputation) information, and thus be used as platforms for C2C commerce in P2P. This intriguing combination motivated us to come up with the decentralized PKI based identification scheme. Additionally this fits well with the P2P design principle of avoiding any kind of centralization or specialized roles, and thus we argue that our P-Grid based PKI is an important step towards enabling E-commerce in P2P.

7 Related work

Most of the work done in the context of decentralized PKI has used PGP [13] like web of trust models, trying to exploit small-world certificate graphs [5], some of which uses computationally

intensive authentication metrics [18]. We elaborated the drawbacks in detail of such approaches in Section 2. We have then advocated the use of structured P2P systems' efficient searches in order to employ a statistical or hybrid model as a means to realize reliable PKIs, and given a preliminary analysis to demonstrate that it is possible to quantify the probabilistic guarantees in quorum based techniques, unlike the web of trust based approaches.

To the best of our knowledge, we have pioneered in the introduction of a layered model capturing the interdependencies of various security related issues that need to be addressed in order to enable E-commerce in P2P. We have outlined how other security issues will depend on and need an underlying PKI. For example, work done in the context of decentralized trust management ([3], Poblano [6]) often assume an extrinsic mechanism for authentication, and do not clearly address the essential issues of identification and non-repudiation, which is essential for any reputation management scheme. Our model formalizes these dependencies, and thus provides a better understanding as to how to implement future P2P E-commerce systems.

Our proposed decentralized PKI is generic, and in principle our approach could also work on top of other structured P2P systems. However, careful analysis is required for each of these systems to judge their suitability. For example, our approach would not work with the existing features supported by CAN [17] or Chord [19] because they do not provide explicit statements on the management of replicas and do not address the issue of updates, which is very essential for our PKI to work. Freenet [8, 7] may serve as a possible platform but would again require a detailed analysis as to whether its model can provide sufficient guarantees especially with respect to update propagation to cached copies of information.

8 Conclusion

In this paper we advocated the need of a distributed PKI for P2P systems and presented our approach based on the efficient search mechanism provided by structured P2P systems such as P-Grid. While in an unstructured P2P system like Gnutella, web of trust is the only available option, the scheme is inherently inefficient and ad-hoc. It is difficult to give probabilistic or any other quantification of the performance and cost of this approach. In comparison, the statistical approach we employ has various advantages: It helps to share the effort (unlike in web of trust where effort is not shared), has low latency and guaranteed result (unlike the case of trying to establish a web of trust by conducting random walks on a trust graph), and provides mathematically provable guarantees. We also provided arguments proving that our approach is resistant to various kinds of attacks.

Despite several advantages of the statistical approach, the fact remains that the web of trust approach is already in existence, and hence it is our belief that future systems will typically employ

a hybrid version of the web of trust and statistical approaches for maintaining PKIs. Our work on identity management solely by the participating peers rather than relying on trusted third parties is a step towards enabling e-commerce in a totally P2P way. To that end we have also described a layered model for enabling E-commerce. It paves way to support other services like reputation management apart from ensuring reliable functioning of the P2P system itself.

Our work can possibly be applied in domains other than C2C E-commerce particularly that of the emerging web services arena. Each such service may be considered as an "entity" or "peer" which cooperate in a P2P way. Then service discovery [14], as well as keeping track of quality and integrity of such services may be achieved in a completely decentralized manner, thus opening a vista of new opportunities, particularly because even "small players" can participate in such an open P2P society.

References

[1] Karl Aberer. P-Grid: A self-organizing access structure for P2P information systems. In *Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS 2001)*, Trento, Italy, 2001.

[2] Karl Aberer. Scalable Data Access in P2P Systems Using Unbalanced Search Trees. In *Proceedings of Workshop on Distributed Data and Structures (WDAS-20 02), Paris, France*, 2002.

[3] Karl Aberer and Zoran Despotovic. Managing Trust in a Peer-2-Peer Information System. In *Proceedings of the 10th International Conference on Information and Knowledge Management (2001 ACM CIKM)*, pages 310–317. ACM Press, 2001.

[4] Karl Aberer, Manfred Hauswirth, Magdalena Punceva, and Roman Schmidt. Improving data access in P2P systems. *IEEE Internet Computing*, 6(1), Jan./Feb. 2002.

[5] L. Buttyan, S. Capkun, and J. P. Hubaux. Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph. In *Proceedings of The ACM New Security Paradigms Workshop*, 2002.

[6] Rita Chen and William Yeager. Poblano - A Distributed Trust Model for Peer-to-Peer Networks. http://security.jxta.org.

[7] Ian Clarke, Scott G. Miller, Theodore W. Hong, Oskar Sand berg, and Brandon Wiley. Protecting Free Expression Online with Freenet. *IEEE Internet Computing*, 6(1), Jan./Feb. 2002.

[8] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval sy stem. In *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, number 2009 in LNCS, 2001.

[9] Clip2. The Gnutella Protocol Specification v0.4 (Document Revision 1.2), Jun. 2001. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[10] Anwitaman Datta, Manfred Hauswirth, and Karl Aberer. Updates in highly unreliable, replicated peer-to-peer systems. *To appear in the Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, 2003.

[11] eBay Inc. ebay, 2003. http://www.ebay.com/.

[12] Manfred Hauswirth, Anwitaman Datta, and Karl Aberer. Handling Identity in Peer-to-Peer Systems. Technical Report IC/2002/67, École Polytechnique Fédérale de Lausanne (EPFL), 2002. http://www.p-grid.org/Papers/TR-IC-2002-67.pdf.

[13] PGP Homepage. Pretty Good Privacy: PGP. http://www.pgpi.org/.

[14] Wolfgang Hoschek. A Unified Peer-to-Peer Database Framework and its Application for Scalable Service Discovery. In *Proc. of the Int'l. IEEE/ACM Workshop on Grid Computing (Grid'2002), Baltimore, USA, November 2002. Springer Verlag.*, 2002.

[15] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*, chapter 3.6, "The Coupon Collector's Problem", pages 57–64. Cambridge University Press, 1995.

[16] Sharman Networks. Kazaa, 2003. http://www.kazaa.com/.

[17] Sylvia Ratnasamy, Paul Francis, Mark Handley, Rich ard Karp, and Scott Shenker. A Scalable Content-Addressable Network. In *Proceedings of the ACM SIGCOMM*, 2001.

[18] M. Reiter and S. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2), 1999.

[19] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM*, 2001.