Abstract. On the Internet we often see a various systems for creation and maintenance trust between participants on network. There is, for example, a Reputation system, where all participants are building-up his reputation by honest handling with others. Users evaluate each others with points after each successful transaction. Here may occur some problems, like one or more of users intentionally give negative feedback (bad-mouthing) or wants earn positive feedback even if he don’t accomplish requirements (ballot-stuffing). Nice example of such reputation system is Ebay.com. But there is interaction between people, and this system we can not use in automated environment like p2p networks or distributed computing markets without some modifications. So in first section of this paper we describe some classic Reputation systems, and then we introduce one automated system for reputation based trust in Distributed Computing Markets. There Servers provides service – complicated computations and Clients pays for this service. Here we solve problem of security by a Threshold witnessing where by Clients and Servers are number of Witnesses (their count is more than some threshold), which evaluate and verify work of Server and at the same time verify each other.

Keywords: Trust, Reputation Systems, Electronic Commerce, EBay, Auction Systems.

1 Introduction

1.1 What it is Reputation Based Trust?

One of most characteristic property of the Internet is anonymity of his users. It is very convenient but often very dangerous property. For example advertisement - Download this program, it is fantastic, or Click Here you can win a million dollars, and what happens next we all know very well – virus infection, Trojan horses, spyware, adware a password stealing etc.

How we can avoid this? In classic physical world we trust all our friends, because we know them. They has a name, we know them a long period of time, they never did something wrong to us, so we trust them. This experience we can apply to environment of Internet. Every participant gets a name (with some password of
course) and we must provide that all activities that relates to that name will be recorded. Was he honest? Did he what he promises? Can we trust him? We can make some rating of this, store this information somewhere and.. Bingo! We have system for Reputation Based Trust!

1.2 EBay

With such systems we can on Internet found relative long time (what is long time on Internet? Year? Three years?) One of most known is Auction system EBay (www.ebay.com, www.ebay.co.uk, www.ebay.de, …) For example there is this situation: I want to buy a mobile phone, I know the type, his features, I like it very much and I want it for as low price as is possible. And here is some man from Australia, he tell himself that, there is new kind of this mobile phone and what with old one? It is easy to make an advertisement: I want to sell mobile phone.. ,but there is a problem, who can guarantee that some person from Australia sends a packet with a phone that wrote in advertisement? And who guarantee that some boy from somewhere in Europe sends money to Australia? We don’t know even his true name. So and here is EBay, with its “Feedback forum” If I fairly pay for the item, and he sends what he advertised, I can receive positive feedback from him, and he can get positive point from me and it all with short description: item such as described, payment comes quickly, fast delivery, good packaging, nice handling, recommended buyer etc. Score grows or falls, every participant can see all positive, negative or neutral feedback from seller’s past transactions, and every seller can check score of his potential buyer. I will newer send my item somewhere to member from Nigeria which will pay by personal cheque, because he has negative or no feedback and I don’t believe, that he send me any cheque. This way EBay internationally transmute trash into treasures. Of course EBay is Auction system, it means that if I want to sell something, I give starting price, and who make highest bid gets my item. And even more, many people makes on EBay their business, they sell internationally, (for example feedback score 22000 points). Don’t you want to buy a memory card from Honk Kong? 20% cheaper than here in Slovakia 😊.

1.3 Other Examples

There are other auction websites with similar philosophy like EBay. For example Amazon auctions (auctions.amazon.com), or there was Yahoo auctions, (unfortunately retired from June 2007). This websites have little modifications for example averaging of feedback score, with scale from 1 to 5, miscellaneous details like friendliness, prompt response, quality of product, etc.
Reputation systems we can found in various other uses. We can mention the website www.Allexperts.com or www.AskmeHelpdesk.com this so-called “expert sites” you can ask questions to self-proclaimed experts from various areas of knowledge. They provide answers in exchange for reputation points and comments from other users. Or website www.Bizrate.com rates registered shops by asking consumers to complete a survey from after each purchase, rated is customer support, time of delivery, shipping charges and many more. Other example is www.Epinions.com which offers rating service for product reviewers, the better the review is the more points the reviewer receives.

1.4 P2P Networks and Automated Environments

Reputation systems can be used also to establish trust among members in P2P networks. There can identify malicious peers and can prevent the spreading of malicious content. In such P2P network is not available a central server, so information about reputation of members must be shared between peers. For example every peer maintains a rating of every other peer that it has dealt within the past.
Similar situation is in computing markets and grids, where Servers offer compute services and Clients pays for this service. Here also centralized trust authority is not available and distributed alternative is required. Here is not acceptable ad-hoc manner of handling like in human driven reputation systems, and we need stronger, secure mechanisms of trust. And about one solution of this problem will be the next section off this paper.

2 Uncheatable Reputation for Distributed Computation Markets

Here we describe one solution for secure reputation management in a distributed computing environment. This solution is composed of two major elements: a proof of computation ("ringers") and a “threshold witnessing” mechanism (that means that there is a set of sufficient witnesses, which sign a document to certify a new rating). Market for computation here means that participants export CPU cycles that others can use in exchange for payment.

2.1 Overview of the Solution (Fig 2.)

Bob wishes to get a given computation $f$ executed over a set of input data items $\{x_1, \ldots, x_A\}$ in exchange for payment. Both the payment and the amount of time he is willing to wait for service completion are upper-bounded.

- (step 1) Initial witness selection phase: Bob randomly selects a set of $2c + 1$ computation "witnesses" $W_j$ (this provides a threshold-secure way of avoiding illicit ratings).
- (step 2) He then sends to all of them (via multicast) a service request including: $f$, the inputs, the payment and target execution time upper bounds.
- (step 3) The witnesses then perform a distributed server selection process, at the end of which the least-costly, best-reputation, available server is selected to perform $f$ for Bob. As the adversary model guarantees a majority of the witnesses are honest and non-colluding, this process is to complete successfully. Let the selected server be "Alice". Note that the selection of Alice is not under the control of Bob.
- (step 4) Alice is provided $f$ and the input data set and the witnesses then initiate the process of threshold witnessing by sending (each in turn) a set of challenge ringers to Alice.
- (step 5) Upon executing the computation Alice completes the witnessing process by returning the execution proofs associated with the challenge ringers to the witnesses, as well as the actual computation results back to Bob.
- (step 6) Finally, depending on the proof correctness, the witnesses sign (using verifiable threshold signatures) a new rating (a combination of the previous rating and "good" if correct proofs or "bad" otherwise) for Alice and distribute (broadcast) it. If the rating does not change it is not distributed.
Securing reputation mechanism in computing markets

Fig. 2. Overview of the solution
2.1 Building Blocks

Rating Store Management. Reputation value can be created or changed only if at least \( \geq c+1 \) participants agree. Every participant stores the most recent reputation for every other participant (signed with the secret master key) together with time per instruction and cost per time (signed with the participant's private key).

Witness Selection. B (client) select \( 2c+1 \) witnesses randomly. Then creates a multicast channel for the witnesses and sends the (signed) job description: \( f \), the set of input values \( \{x_1...x_a\} \), the maximum time B is willing to wait for job completion, the maximum amount B is willing to pay for the computation, and signed digest of this information, along with a certificate containing B's public key.

Server Selection. The \( 2c+1 \) witnesses selects the most suitable service provider. It will be the participant with the best reputation, among the Servers which don't exceed Bob's determined time of computation and cost per time. Let that participant be Alice (A). Even with \( c \) faulty witnesses, no less than \( c+1 \) witnesses will select A. A is added to the witness multicast group. One of the witnesses "the leader" multicasts the job description received from B.

Threshold Witnessing. This operation requires the \( 2c+1 \) witnesses to first export B's computation to A, then verify the accuracy of the computation performed by A, and based on the quality of the service performed, compute and sign the new rating of A. The essence of the service witnessing operation is the usage of "ringers".

Ringer Generation. Each witness (\( W_j \)) selects one random value \( x_z \) from the input set specified by B in the job description and computes a ringer \( r_j = H(f(x_z)) \). \( W_j \) generates a unique session identifier, sid. (to prevent replay attacks by introducing a freshness element). Then \( W_j \) sends \( S_W(H(Id(W_j); sid; r_j)) \), its identifier, sid, the ringer, together with the signed digest and \( W_j \)'s public key certificate to A. When A receives such a message, it verifies \( W_j \)'s signature. Note that even though A knows \( r_j \) and A may collude with a subset of the witnesses, none of them actually knows the \( x_z \) value generated by an honest witness. A waits to receive \( 2c+1 \) valid messages for the same session identifier, sid. If within a given time frame, starting with the receipt of the first ringer, A receives less than \( c+1 \) such messages, it ignores the job received. Otherwise, A sends a multicast message to all the witnesses that participated. The message contains a concatenation of all the signed ringers received. The witnesses that receive this message, inquire the remaining witnesses for their ringers.

Revealing the Ringers. A performs the computation and reveals the input values \( x_z \) hidden in the \( 2c+1 \) ringers. A creates a single message containing \( S_{w_j}(H(Id(W_j); sid; r_j)) \) and \( S_A(H(Id(A); sid; z)) \), for \( j = 1...2c+1 \). The message also contains the results of the computation,
f(x_1),...,f(x_A), along with its signed digest. Note that the first signed digest was sent by W_j, and is used to prove the value of the ringer r_j.

A then sends this message on the witness multicast channel. Each witness W_j verifies the correctness of only its own ringer, that is, r_j = H(f(x_j)). The multicast of A is meant to prevent a witness from falsely claiming that A did not send back a correct answer.

If any witness W_j discovers that A did not send back x_z or that r_j <> H(f(x_z)), W_j sends a multicast message to all the other witnesses revealing this fact. The other witnesses are able to verify the claim by computing the correct answer to W_j's ringer and compare it with the answer sent back by Alice (received during A's previous multicast) This acts as the proof that A did not perform the entire computation. A negative rating is then issued.

**Signature Generation.** Based on A's current rating, the returned results of the current computation, and the time elapsed since A received the job description, each witness W_j is able to compute A's new rating. In general, if A is caught cheating, either by not performing the entire computation or performing it slower than advertised, its rating will decrease, otherwise, it will increase. Each W_j then generates a verifiable signature share of A's new reputation.

Then W_j sends this value, along with its certified verification key VK_j and A's new rating in clear, to all the other witnesses, using the group's multicast channel. Each witness waits to receive c correct signature shares for the same new reputation of A as the one generated by itself. Any participant can verify the validity of a signature share.

Since c + 1 different and correct signature shares are enough to generate a valid signature, each witness is able to generate the signed new rating of A locally.

**Reputation Distribution.** The results of the computation are returned to B and the new reputation of A is distributed: The first witness is in charge of sending the new reputation of A on the broadcast channel to all the participants in the system. Note that a witness cannot simply send an incorrect reputation since it will be easily detected, as it would need a fresh timestamp and to be signed with the master key, that is, by at least c + 1 honest witnesses.

**Punishing Malicious Witnesses.** If a witness acts maliciously at any stage (e.g., as part of rating signing or distribution), the witnessing scheme allows for immediate counter-measures upon detection. One such measure could be simply flagging the identity of the particular witness and effectively removing it from the system. Also, the only actual effect of such malicious behavior is just a minor slow-down of the protocol but sure detection. So the likelihood of a witness committing such "suicidal" action can be considered minor. Thus any rational witness should act honest in the witnessing protocol, for its own survival benefit. As a result, e.g., the expected number of transmissions required for rating distribution is exactly 1.
2.2 Attack and Improvements

**Cheating and Laziness.** Is bad-mouthing (incorrect negative feedback) possible? A bad rating requires a secured proof of non-compliance by Alice, that all witnesses agree with. If Alice responded correctly, all messages are signed, ringers are non-invertible, and at most $c$ of the witnesses are malicious, this is not possible. Furthermore, a straight-forward ballot-stuffing (un-earned positive feedback) attack, where clients create and duplicate simple compute jobs in order to artificially increase the ratings of preferred servers, is thwarted through the indirection introduced by the witnessing layer. For each job requested by a client, a server is chosen by $2c+1$ witnesses, containing an honest majority.

Next, is "lazy behavior" (resulting in ballot-stuffing) possible and how likely? What is the likelihood of cheating by simply finding the ringers after doing less work than required? From analysis in [3] results that the more challenges are presented to Alice, the less its probability of getting away with less work.

Here is another important problem: if Alice knows the number of ringers, once she finds all of them she can simply stop working. Simple and effective solution can be to add "fake" ringers to the set of submitted ringers. Because she does not know which ones and how many of the challenges are fake, she is forced to execute all the queries to guarantee a correct answer.

**Sybil Attacks.** The attack consists in the creation of large numbers of fake identities by a single malicious entity (hence the name "Sybil attacks"). The fake identities could then be used to "help" the malicious entity in supporting its opinion, effectively implementing ballot stuffing or bad-mouthing attacks. Fortunately, since any fake participant can own at most a non-unique master secret key share (of the malicious entity that created it), it will be rendered unsuitable to act as an independent witness.

**Mobile Virus Attacks.** If participants are similarly configured, it is possible for attackers to repeatedly exploit discovered weaknesses and rapidly corrupt a number of hosts in excess of the bound $c$. An elegant solution for mobile viruses that can easily be applied in our setting, combines proactive re-computation of the verifiable secret shares of participants with the rebooting of infected hosts. Periodic share renewal provides also a simple solution to the key revocation problem. Participants that leave or are eliminated from the system, loose their ability of acting as witnesses at the next share renewal period.

3 Conclusions

Reputation Systems is one of ways to create trust between participants on Internet. It is similar to creation trust between people in real world. In automated systems in space of computers we must use cryptography knowledge, but the basis is the same. Thanks to it we can make Internet more secure and usable in more areas of our activities.
References

Securing reputation mechanism in computing markets

Server Selection (cost function + rating)

Witnesses

Witness Selection

Witnessing

Witnessing details

check: \( z' = z \)?

- yes
  - failure proof \( (z, rnd, r_j) \)
  - new (+) rating threshold signature share for \( W_j \)

- no
  - ringer: \( r_j = H(f(x_j)) \)

Alice (Server)

proof: \( z' \)

Bob (Client)

reputation(Alice)