Proposal of a new trust model for the Internet peers using a financial reputation score
財務面の評判を用いたインターネットピア向け信頼モデルの提案

by

Marat Vyshegorodtsev

Submitted to the Department of Electrical Engineering and Information Systems
in partial fulfillment of the requirements for the degree of

Master of Engineering

at the

THE UNIVERSITY OF TOKYO

March 2013

Author . . . . . .
Department of Electrical Engineering and Information Systems
February 6, 2013

Certified by . . . . . .
Yasushi Wakahara
Professor
Thesis Supervisor

Proposal of a new trust model for the Internet peers using a financial reputation score

財務面の評判を用いたインターネットピア向け信頼モデルの提案

by

Marat Vyshegorodtsev

Submitted to the Department of Electrical Engineering and Information Systems
on February 6, 2013, in partial fulfillment of the
requirements for the degree of
Master of Engineering

Abstract

The trust anchors defined in the domain name system (DNS) and secure socket layer (SSL) architecture make the governance of the Internet very rigidly aligned with the domain name registrars and the software vendors. The root certification authorities are defined ultimately and built into the software key stores. However, an end-user is not given any criteria to distinguish the built-in root CAs; hence he is unable to manage the default key store effectively.

In this thesis, a new approach of creating a reputation score for the trusted third parties (TTPs) is proposed. Each TTP is evaluated using its financial performance. The amount of transaction fees put on the certificate-signing transactions is counted and used as the reputation score instead of peer feedbacks. It becomes more expensive for a potential attacker to perform Sybil attack due to the increased cost of the vote. The verification experiments showed that even if the feedback-based reputation score is replaced with the financial reputation score, the peer distribution will still follow the power law distribution, and thus the trust decision algorithms based on power law are applicable.

Flatcoin, the practical implementation of the proposed reputation system is also described. It expands the Namecoin protocol with certificate payment transactions. The distributed timestamping approach used in Flatcoin provides domain name registration and paid certificate-signing requests in a completely distributed manner. Flatcoin effectively replaces DNS, DNSSEC and SSL Extended Validation without any single pretrusted certificate delivered with the client software.

Thesis Supervisor: Yasushi Wakahara
Title: Professor

Acknowledgments

The research study of this thesis was financially supported by the Monbukagakusho scholarship from the Japanese Ministry of Education, Culture, Sports, Science and Technology from April 2010 through March 2013.

The author is greatly indebted to the thesis advisor, Professor Yasushi Wakahara, for the continuous support of the master research study, for his instructions and guidance on how to become a good researcher, for his patience, kindness, enthusiasm, expertise and immense knowledge. He had always pushed me to think deep, and to look beyond the surface. Second, I would like to express my heartfelt gratitude to Assistant Professor Daisuke Miyamoto, for sparing his time to join the weekly discussion on my research progress and providing his valuable suggestions, comments and technical help. I also want to thank all the CNL lab members.

I would like to express gratitude to Rakuten and Bonakodo for providing necessary data sets and advises on security and data mining technologies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter identifies the problem, sets the research objectives and shows this thesis outline.

1.1 Problem formulation

The constant growth of the Internet led to trust issues on global scale. The importance of data and information transmitted over the wide has increased significantly, so that encrypted and authenticated channel is the basic deployment requirement for the new major services. In recent years, web giants, such as Gmail, Facebook, Twitter, switched to "full HTTPS" protecting not only the login page but all the delivered content. Recent projects, such as GitHub, make their services HTTPS-only and completely disable unencrypted and unauthenticated communication channels. The common trend for the next several years is reduction of untrusted data in all protocols that could pose a risk to the assets and business continuity of organizations in the Internet.

Secure Socket Layer (SSL) solves the integrity and secrecy of HTTP and authenticity of DNS as follows:

- Authenticity of the IP-address and DNS name pair
- Authenticity of the IP-address and entity name pair (only in high-grade Ex-

tended Validation SSL – “green address bar certificates”).

- Secrecy of transmitted data
- Integrity of transmitted data

While secrecy and integrity problems are widely researched by cryptographers, the root cause of the authenticity problem lies in the field of trusted relationships.

Current SSL implementation follows ITU-T Recommendation X.509 [4] standard which has significant disadvantages:

- Trade-off between number of certificate authorities (CAs) delivered with the software and price of the certificates, i.e., the fewer CAs the Internet has, the more expensive the certificates are. Compromise possibility is higher with bigger number of certificates.
- Trade-off between number of CAs and geopolitical coverage, i.e., there could be no trusted CA if the number of them is few
- Unlimited scope of authority of each CA

Similar problem occurs in the DNSSEC implementation, however the purpose and trust anchors’ selection algorithm of the systems are different. In DNSSEC, the trust tree is aligned with IANA, zone maintainers and registrars, while in SSL – with software manufacturers, root CAs and subordinate CAs. All peers in the Internet unconditionally trust all parties included in these trust trees. Such conditions lead to major security breaches: compromise of the CA signing key [5, 6], misuse of the CA privileges [7, 8].

The problem could be stated in one sentence: very small trusted authorities have a gigantic impact on the whole Internet. Currently, Mozilla Firefox includes around 120 root CA certificates. Counting all subordinate CAs the number of all trusted third parties (TTPs) exceeds 640 CA certificates or 320 different organizations. Such a big number of TTPs breaks the main principle of the notarization, that is both subject and verifier peers must explicitly trust the TTP. Most of the certificates are

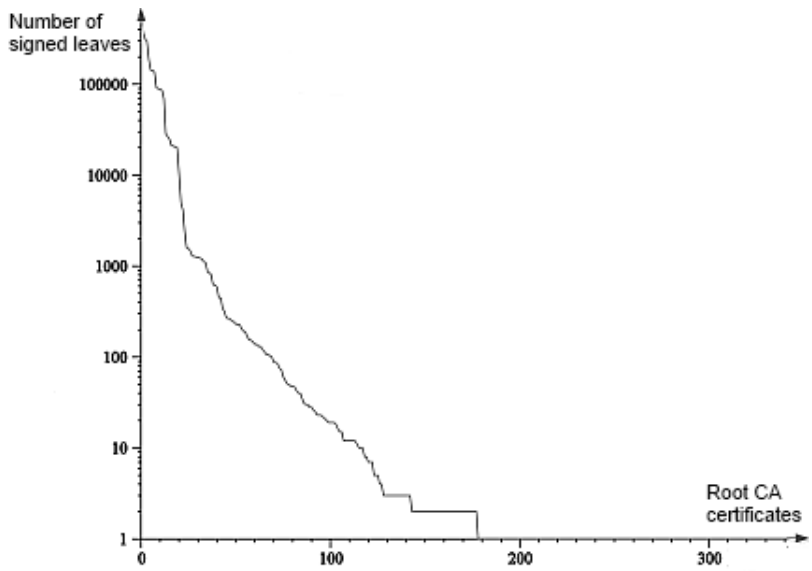

Figure 1-1: Distribution of SSL CAs by number of leaves according to EFF [1]

not known to anyone or not trusted at all, for example, the subordinate CA certificate issued to the Department of Homeland Security may not be trusted by default in the countries other than the United States. The results of the SSL Observatory by EFF [1] show that more than a half of all CA certificates did not have a single subject on the IPv4 block at port 443, nevertheless these certificates are automatically trusted by the network software. The distribution is shown in figure 1-1.

1.2 Objectives

The main goal of this research study is to:

- Identify the ways to localize the impact of TTPs in SSL and DNSSEC on the Internet,
- Develop a robust and secure way to pin certificates to IP-addresses and entity names,
- Reduce the number of CA certificates in the key store to the minimum required level defined by the end-user,

- Provide some independent dynamic metric to assist end-users in the decision whether to include a CA certificate into the key store or not.

The proposed method is supposed to provide the same or better level of security, while reducing the dependency of the Internet on the X.509 trust tree.

1.3 Thesis outline

The rest of the thesis is organized as follows: chapter 2 explains the background of our research and the technologies we have applied to our proposal. Chapter 3 introduces the new reputation system that we have developed and explanation of its implementation. In chapter 4, the proposal verification is performed. Conclusion, limitations of current work and future work are given in chapter 5.

Chapter 2

Background research

In this chapter the background technologies and research studies are described.

2.1 Trust models

In the early 90's, when the Internet was a tool for mainly computer science related people, the problems of trust started to arise. Some engineers at Netscape developed a protocol to secure HTTP communications called SSL (X.509). The protocol was designed in a way that considered only one trust anchor – Verisign. This decision lead to a monopoly making SSL certificates very expensive, and thus a lot of criticism was received by Netscape. From that moment the number of certificates has significantly increased reaching about 650 different organizations around the world ranging from Arabic telecoms and Chinese government to US Department of Homeland Security. Given the difficult geopolitical situation and general interpretation of trust in different countries, it may be concluded that the increased number made the trustability factor of SSL decrease. The problem is complex and gives us a trade-off between trustability and number of CA certificates in the trust chain. Several proposals approach this kind of problem; they are described in subsections below.

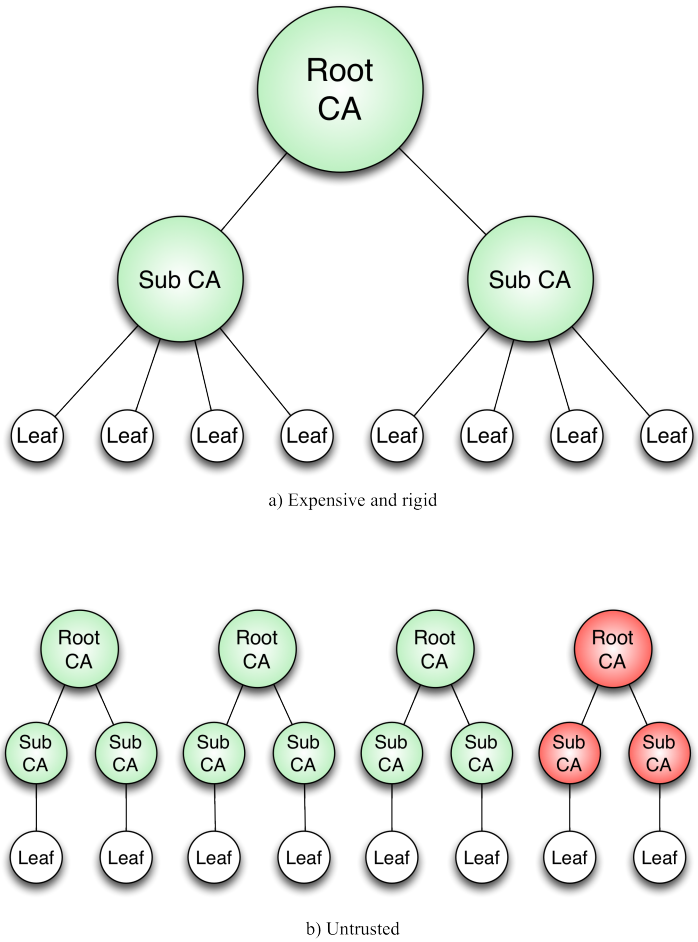

Figure 2-1: Structure of trust in SSL (a – in early times, b – now)

2.1.1 X.509v3 and cross-signed schemes

The X.509v3 certificate standard allows complex certification path building as described in IETF RFC 4158 [9]. The certification path structures such as "mesh" or "bridge" can solve the problem when a compromise or misbehavior of a single CA leads to the compromise of all the Internet SSL users. However, such a mechanism still implies a predefined built-in list of CAs delivered with the client software (web browsers, middleware, operating systems). Such lists of CAs can be effectively maintained in corporate environments through group policies, but in the personal computing world, a user's decision is left up to software vendors. Such conditions limit the ubiquity of SSL: the maximum number of CAs in a meshed path structure is limited to the number of CAs built into the client software. Moreover, the stan-

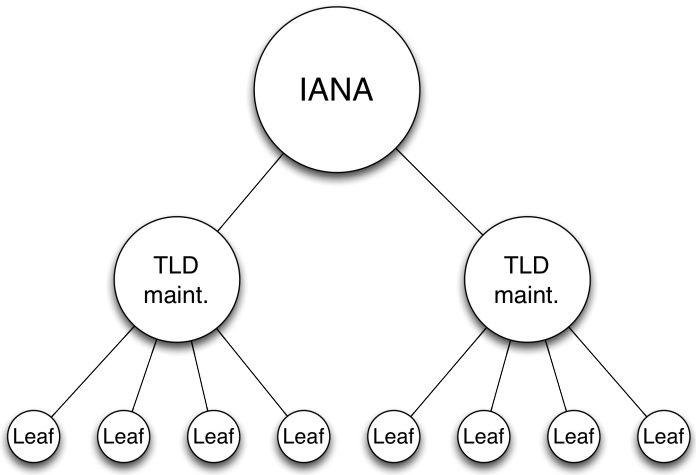

Figure 2-2: Structure of trust in DANE

dard does not clearly define the number of cross-links and the decision algorithms on whether CA path should be trusted or not. For example, if a certificate is signed by only one CA, the scheme converges to the regular SSL.

2.1.2 The DNS-Based Authentication of Named Entities (DANE)

The IETF RFC 6698 [10] standard suggests another approach of delivering trusted certificates to an end-user. Instead of signing the certificate with a trusted CA, the peer would put the certificate fingerprint into a DNS record. The authenticity of DNS record is guaranteed by DNSSEC, and thus the trust chain is established as in figure 2-2.

The DNSSEC architecture is strictly hierarchical and strongly resembles the first model of SSL: Verisign is replaced with IANA and subordinary CAs are replaced with zone maintainers and registrars (see figure 2-1, a). Additionally, the entities in the trust chain originally were selected as domain zone maintainers, not as trust anchors, i.e., companies managed with strict security guidelines such as WebTrust for SSL CAs.

2.1.3 TOFU, TACK, and Perspectives

The concept of trust-on-first-use (TOFU) is widely used in OpenSSH. Its security is based on the belief that an attack will very unlikely occur during the first communication with the server – so called "prayer condition".

TOFU can hardly be applied to SSL as is – peer certificates are renewed very often, and thus too many prayer conditions will occur. CA certificates are renewed less often and the number of them is fewer, and thus TOFU is better applicable for "pinning" CA certificates, i.e. remembering certificate public key or fingerprint for the given hostname.

Self-signed certificate pinning is proposed with the internet-draft for Trust Assertions for Certificate Keys (TACK) [11]. Once a client connects to a remote host, it tries to remember the TACK public key (self-signing CA) as in TOFU and reuse it in further connections. This proposal eliminates the necessity to "pray" every time the remote host's certificate is changed, and yet it is required to take a risk of untrusted connection, when the self-signing CA certificate needs to be renewed. The proposal mitigates the risk with the TACK rotation algorithm, but this algorithm would not work for clients not communicating with the host for a long time.

The Perspectives project [2] offers a solution similar to the IETF RFC 5055 "Server-based Certificate Validation Protocol" [12]. An improved caching concept of TOFU among distributed network of notaries is suggested. When an end-user requests a certificate from the remote peer, he does it through several representative channels (notaries), so that even if some channels are compromised, it will become clear to the end-user that the attack was being performed (see figure 2-3).

The research study shows that the cross-linked scheme with additional notaries significantly decreases the probability of compromise. The solution successfully eliminates the problem of one CA compromising the whole network, and yet it still requires some default list of notaries to be maintained. Users can add or remove their own notaries from the list, however such operation is not different from maintaining the key store for the regular SSL. In addition, the concept of remote path validation breaches

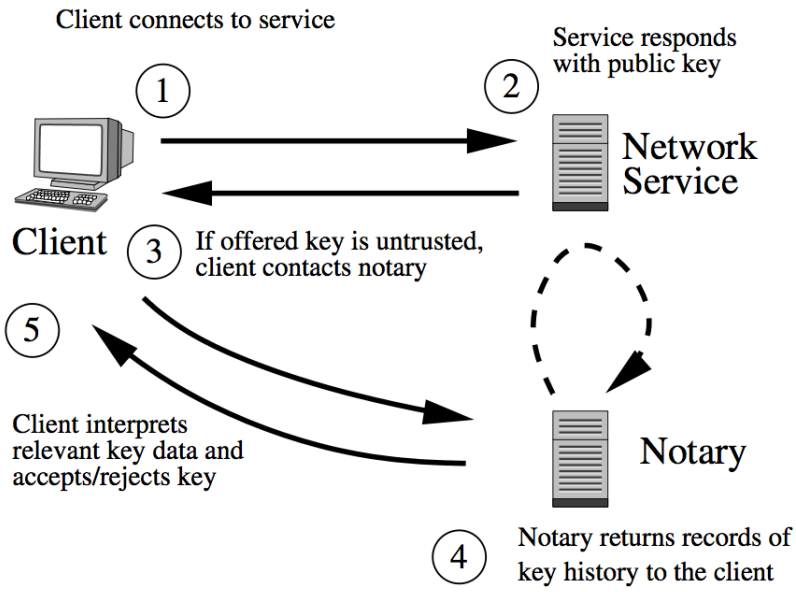

Figure 2-3: Overview of a client using Perspectives. In practice, several notaries would be contacted in parallel before making a key trust decision [2].

the end-user's privacy. The authors suggest adding of one more communication tier to protect privacy that will increase latency of the communication.

2.2 Proof-of-work

Proof-of-work (POW) is an asymmetric economic measure that makes repeated attacks disadvantageous to benign behavior. Repeated attacks include Sybil and Denial-of-Service attacks. In both cases, an attacker creates many false identities to abuse either reputation system or available resources. For example, to abuse a voting system, the attacker would register as many bogus identities as he could and vote for the desired candidate option [13]. POW adds extra cost to each vote, so that it becomes economically unreasonable to perform the attack.

POW functions are often used as cryptographic primitives.

2.2.1 Hashcash

Blocking the spam or denial-of-service attacks converges to the problem of peer authenticity: if a remote server could identify each sender of an e-mail or a TCP packet, then it would be easy to block spam bots or flooders. In fact, this problem converges to even narrower problem of identifying whether the remote user is a single peer or a set of Sybils of this peer performing an attack. Defense mechanisms generally utilize the differences between human peers and automated peers (e.g., CAPTCHA) or require some work on the remote user's side to increase the cost of the attack (e.g., Hashcash).

Hashcash utilizes a computational imbalance between benign and malicious behavior. An attack would require a lot of computational resources to generate a valid hash for each e-mail. A benign user would spend a feasible minimum of his resources to calculate only one hash. The algorithm of Hashcash is based on bruteforcing a nonce value, such that gives a SHA-1 hash of given format, e.g. starting with N leading zeros. Since the output of SHA-1 algorithm is unpredictable, the time needed for finding a solution nonce is high. The number of leading zeros expected in the output are correlated with bruteforcing time exponentially.

For example, let us consider a sender alice@cnl.t.u-tokyo.ac.jp willing to send a message to bob@cnl.t.u-tokyo.ac.jp. The mail server at cnl.t.u-tokyo.ac.jp sets the requirement of minimum 20 leading bits of Hashcash token to be zeroes. Alice creates a message with the following parameters:

```
Version=1
Bits=20
Date=130205
Resource=bob@cnl.t.u-tokyo.ac.jp
Extension field (optional) = [empty]
Random = Some random value to avoid stamps collision
Counter = Some value that is required to find a stamp with the desired
    number of preimage bits
```

The parameters above would produce the following string:

```
1:20:130205:bob@cnl.t.u-tokyo.ac.jp::[a-zA-Z0-9+/=]+:[a-zA-Z0-9+/=]+
```

After a second of bruteforcing the counter value, a hashcash token is created:

```
1:20:130205:bob@cnl.t.u-tokyo.ac.jp::/D3BrOa+OAgoHRBu:
    000000000000000000000000000000000000B3jK
```

Taking a SHA-1 sum of this token would produce the following string:

```
0000092b0d997c48fe033d7bb456eeb3e9c73ade
```

Note that 5 leading hex digit zeroes equal to 20 bits of binary zeroes. This technique reduces the sending rate of the attacker by the time required to bruteforce the counter. Also, it adds extra CPU and electricity cost.

The Bitcoin protocol uses Hashcash algorithm for transaction timestamping: the transaction block should end with the counter and the SHA-1 function should produce the same result with leading zeroes. The number of zeroes increases with time limiting the number of Bitcoins in the system – when the number is too high, it is almost impossible to find the necessary token value. The more detailed specifics of Bitcoin protocol will be described below.

2.2.2 Bitcoin

There are two main reasons why all known government-backed currencies are very centralized: the mint has to control emission of money and maintain integrity of the monetary funds. Creating a completely decentralized peer-to-peer currency implies that solution to these two problems was found without using a mint.

Bitcoin addresses these two problems in the following way (see figure 2-4):

1. All transactions are gathered in blocks, and blocks are connected in chains

2. Each block becomes valid when a hash value of given format was calculated. The hash calculation algorithm is the same as in Hashcash.

3. Each transaction is added to the block, duplicate transactions are ignored. The longest chain of the valid blocks is considered to be the official timestamped log of transactions.

Peers get rewarded with Bitcoins for calculating hashes, however the complexity of hash calculations (the number of leading bits) is increasing over time. Therefore, the uncontrolled emission problem is mitigated. Since all transactions are timestamped in a trusted distributed manner, the problem of integrity of monetary funds (i.e. double spending) is also tackled.

Transaction chaining as shown in figure 2-5 allows us to build different applications requiring trustworthy timestamping, such as DNS, auctions, homestead property allocations and so on. In this thesis, timestamping is used to mark domain name registrations (see section 2.2.3) and SSL extended validation transactions (see section 3.2).

2.2.3 Namecoin

The timestamping based on POW could be applied to different types of transactions, since it eliminates ownership ambiguity in systems where "first come, first served" principle works. For example, in domain name system (DNS) the owner is the first person, who registered an intent for the domain name. The whois database is used to keep the list of the registered domain names. This database is timestamped by the registrars. Applying the concept of distributed timestamping as in Bitcon, it is possible to drop the registrars from the domain registration process.

Namecoin [14, 15] is an example of a distributed timestamp DNS. Each domain name registration transaction must be verified in the similar way as transactions are verified in Bitcoin. To use Namecoin each peer needs to generate a pair of public and private keys. All further transactions are signed with this key pair, therefore DNSSEC is not necessary in Namecoin DNS.

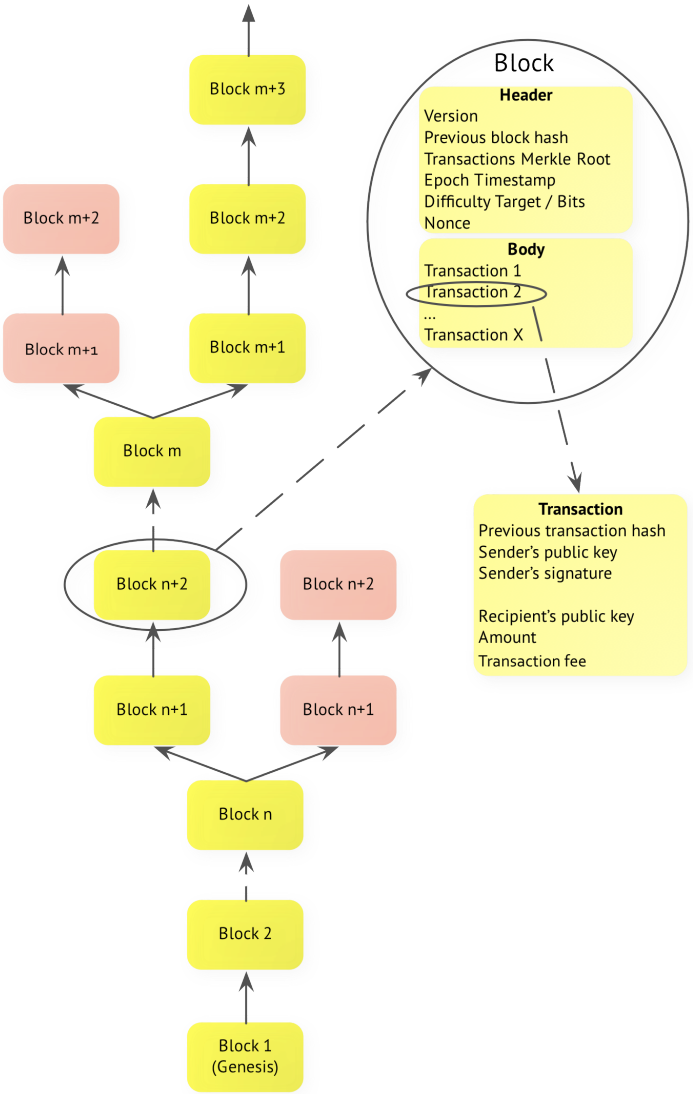

Figure 2-4: Bitcoin calculation algorithm

The important feature of such registration mechanism – the peers do not use any trust anchors, but instead they negotiate what data is trusted using the defined protocol for timestamping and transaction chaining as in Bitcoin. The key pinning occurs upon the domain name registration, hence the proposals described in section 2.1.3 are also not necessary. Therefore, Namecoin solves the problem of DNS governance and the problem of ownership authenticity. Such a solution comes with a trade-off between usability and cost – computing hashes is a very CPU-heavy and wasteful computing job.

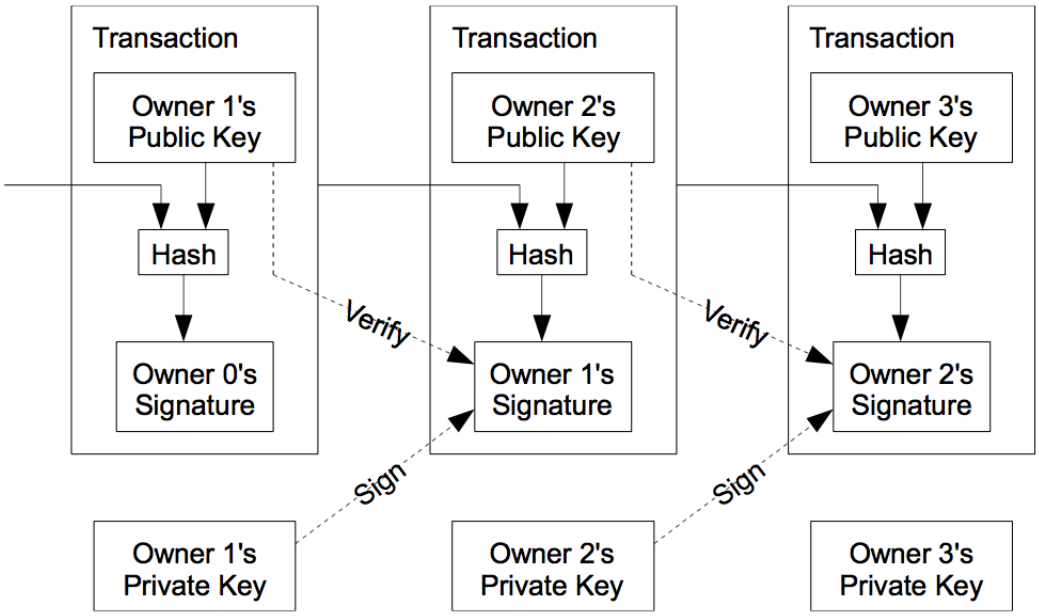

Figure 2-5: Bitcoin chaining [3]

Additionally, Namecoin does not solve the problem of entity authenticity, i.e., binding a domain name to a real person or organization as in SSL Extended Validation (EV). The SSL EV could be used, however, it would ruin the ubiquity of Namecoin and inherit all problems of rigid SSL structure described in section 2.1.1. The proposal described in the thesis mainly is focused on this problem.

2.3 Reputation systems

In web-of-trust (WOT) networks, loose connectivity of the peers can be solved by implementing reputation algorithms. In this research study, three approaches were analyzed: peer feedbacks, stochastic analysis and credit (quota) feedback. These approaches were found insufficient as a solution to global reputation score calculation.

2.3.1 Peer feedback

Peer feedback systems employ voting from each peer against every other peer upon their communication. For example, when one peer buys something from another peer

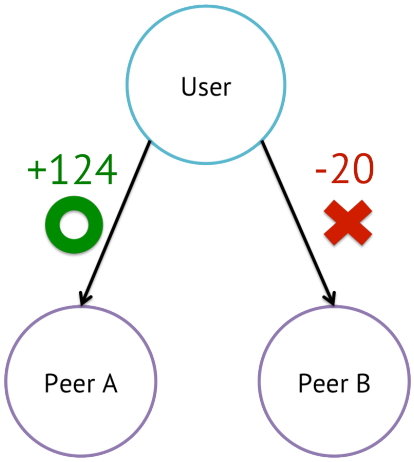

Figure 2-6: Feedback-based reputation systems

at an Internet auction, the buyer can rate the seller on quality of the product, speed of delivery, speed of reaction and so on. The feedback could be mutual, so that buyers will have some reputation score before they begin selling things.

Such voting systems could be binary, trinary or complex. Binary systems consist of two types of feedbacks: positive and negative (figure 2-6). In trinary systems neutral feedbacks are added to positive and negative, for example as in eBay. Complex systems, as in Amazon.com, use 5-star ratings over multiple parameters and more.

A significant disadvantage of such systems is the exposure to Sybil attacks. In order to mitigate false feedbacks provided by bots, it is required to use some algorithm to classify feedbacks or to weigh the peers. The popular solution for this is dependency matrix as in [16, 17]. However, the attacker can adapt to the algorithm easily by investing some benign reputation to build bridge connections inside of the matrix, so that the algorithm will consider them benign. Cost of the attack largely depends on the minimum cost of one feedback; usually such cost defined by the cheapest product sold by the seller.

2.3.2 Stochastic analysis

Machine learning methods are famous in fighting spam messages and denial-of-service attacks [18]. Main disadvantages of such algorithms:

- Learning data set should be present to form initial parameters of the model.
- Algorithm should react to the information changes, i.e. the learning should occur continuously and on-line.
- The learning data set is hard to form in environments where malicious behavior is hard to define. For example, in financial transactions, when double spending occurs, beneficiary peer would claim that the payer was benign, while the losing peer would react negatively.
- Machine learning algorithms are hardly applicable to the fields where human discretion is fuzzy. The attackers have time advantage required to adapt to the algorithm before it learns new parameters for malicious behavior.

The quality of learning data set should be guaranteed by some single trusted third party. Presence of such a party diminishes the independency component of a peer-to-peer system.

2.3.3 Credit-based feedback

Credit-based reputation systems, such as Reddit, employ a quota on peer feedbacks [19]. A peer gains more voting ability when he receives positive feedbacks from other peers.

Such systems are prone to Sybil attacks as well as regular feedback systems, but cost of the attack is higher, since it could be regulated through with the following mechanisms:

- Invitation-based registration
- Registration withdrawal for both inviter and invitee after certain limit of negative feedbacks
- High initial hurdle before peer is enabled to vote

Such systems work well in closed community environments (Reddit, Slashdot, Habrahabr etc.), but on the global scale they are hardly applicable – bootstrapping of such a system would require some seed, which will distribute invitation initially, and thus the independency property of the system would be diminished. In such case, the reputation system would be a simple improvement over traditional SSL.

2.4 Summary

The main problem of SSL is not completely solved by the research studies described in this chapter. Perspectives and Tack provide better pinning (when current SSL does not provide it at all), but do not eliminate the prayer condition and do not support extended validation. DANE provides pinning to DNS records, but it has hierarchical structure as well as the traditional SSL. Namecoin has distributed architecture and certificate pinning by design, but the extended validation is not provided. Feedback-based and stochastic reputation systems that could enhance trusted connections for extended validation in distributed environments (e.g. Namecoin) are prone to Sybil attacks and are very hard to maintain. Therefore, the new distributed approach is needed in order to provide extended validation (i.e. entity-bound trusted connections) in the future anchorless Internet.

Chapter 3

Financial reputation system and Flatcoin

In this chapter the new reputation system and trust model is proposed in section 3.1. The technical implementation of the proposed system is presented in section 3.2.

3.1 Financial reputation system

This section covers the premises of the proposed system and its theoritical background.

3.1.1 How many CA certificates are needed

According to the research of NetCraft in 2009 [20] and the Electronic Frontier Foundation in 2010 [1], the distribution of the main players in the SSL market is highly skewed and comprised mainly of three players (2009, 2010): VeriSign (47.5%, 39.6%), GoDaddy (23.4%, 21.8%) and Comodo (15.4%, 9.8%). The distribution of clients among CAs is very skewed; and from 70 to 85 per cent of the Internet sites are covered by only three CAs, when the total number of CA certificates reaches 650 and above. Unfortunately, the rest part of mainly unused CAs cannot be simply excluded

due to commercial or gepolitical reasons. The assumtion was made in this research study that each user should be given some criterion to decide whether a certificate of certain TTP should be included into the software key store or not. End-users may start with an empty key store, and then add the required CA certificates to it upon connecting to the sites. Therefore, in the beginning user is provided with *zero* CA certificates in the key store, i.e., browsers and middleware ship without certificates at all.

3.1.2 Gross income as a trustworthiness indicator

In order to rank the TTPs we sought some approach, which will comply with the following requirements:

1. There is no centralized authority to rank TTPs, hence no governance abuse and no targeted attacks
2. The ranking score must be verifiable and consistent among peers
3. The score must be resistant against the known attacks on P2P reputation systems: Sybil attack, false rumors, short-term abuse and denial-of-service [21].

The weighted voting approach with heuristic filtering could not fit the requirements due to the scale of the Internet – too many false identities could be generated to perform Sybil attacks. Verifying each identity would mean gigantic operational overhead in governance and lead to a tree-shaped structure of the trust model.

The basis of the reputation score should be some trustable and verifiable parameter. The gross income parameter satisfies all the requirements above. To achieve financial transparency, it is proposed to pay for certificates using a decentralized currency such as Bitcoin. In Bitcoin, all transactions are public and trustable among all peers. A centralized entity is not needed for transaction verification due to the nature of Bitcoin.

Since only the financial performance of peers is measured, it is pointless for the attacker to perform the Sybil attack: the result of the attack would be the same from

one peer or from many Sybil peers. However, if a peer behaves as a currency exchange node or repeatedly repays itself through a chain of coordinated attackers using the same money token, then its gross income will be very high, hence its reputation score will be undeservingly high as well. To mitigate such a condition the system calculates not the gross income directly, but the amount of transaction fees paid from each transaction. In the Bitcoin protocol the transaction fee is paid to the peer, who has calculated the correct proof-of-work for this transaction's block. When the network is big enough, the payee (TTP) will be unable to predict the destination of the transaction fee; hence the coordinated attack will not be possible and the currency exchange node will have very uncompetitive exchange rate due to high transaction fees.

The gross amount of transaction fees paid will serve as a reputation score metrics basis, but a bare sum of the transaction fees paid would not indicate the dynamics of the reputation score. To prevent short-term abuse the rapid growth of reputation score should be detected.

In the proposed financial reputation scoring system, only the positive feedbacks exist. Each act of purchase is counted as a positive feedback with the value of the transaction fee per this transaction. However, the feedback can be revoked together with the trust revokation to the TTP from the client. Adding negative feedbacks to the system would lead to attacks where malicious peers intentionally provide false negative feedbacks to benign peers, e.g., rivals.

3.1.3 Longevity of the reputation score

The proposed transaction type has an expiration date equal to the requested certificate expiration date. The reputation score of the TTP must be decreased by the amount of transaction fee upon the expiration of the certificate. Therefore, the longest issued certificate defines the maximum longevity of the reputation score. The certificate consumer may revoke the certificate actively by broadcasting the intent to the trust network; in this case the certificate is considered expired immediately and reputation score is decreased accordingly. The maximum certificate lifespan should

Table 3.1: A game of two TTP competitors

CA2 / CA1	Maximize income	Maximize fee
Maximize income	Repeat / Repeat	Lose / Win (in the next round)
Maximize fee	Win / Lose (in the next round)	**Repeat / Repeat**

be limited, but the exact period is decided on the client side. It is advised to set the maximum validity period to two years similarly to the current value used in the browsers. If the certificate expiration period exceeds the user policy, then the reputation gained from issuing such a certificate should not be considered.

3.1.4 Tradeoff between the reputation score and the income

Let us consider a simple game of two competing TTPs. The customers of these TTPs are the site owners who decide which TTP to use to validate their certificates. Let us assume that all buyers of certificates are partially blindfolded – they take into account only the reputation score and ignore rest of the factors, when making a certificate purchase decision. If the reputation score is equal, then they are divided equally between TTPs. If the reputation score of one TTP is higher, then all customers buy the certificate from the TTP with the higher reputation score.

The payoff function of each TTP is the gross income; hence they try to maximize the profit, while maintaining high reputation score. Each player has two available strategies: maximize profit or maximize the fee (reputation score). Table 3.1 demonstrates that fee maximization is the dominant strategy and "maximize fee-maximize fee" is the Nash equilibrium.

If both players decide to maximize the income, then the game repeats, because they will have the same reputation score, therefore they will get the same proportion of customers in the next round. However, if one player decides to maximize the income, another would rather prefer to maximize the fee to get all the customers in the next round due to the high reputation. Therefore, both players would prefer to maximize the fee first to eliminate the competitor.

Since the equilibrium point makes the payoff equal for both players, they must gain trust by some other means (e.g. providing better customer support or improve the quality of the service) or simply discount the service price. Such condition brings the market closer to the perfect competition condition, and thus the quality of services should increase. In comparison, the current SSL economic model is closer to the oligopoly, where barriers to entry in the SSL market exist.

In multiple players environment it could be considered that one player makes decisions against each opponent consequentially, therefore the competition can be split in smaller games of two players.

3.1.5 Reputation score dissemination

The transaction fee based score is discrete and symmetric, i.e., the score is the same for all peers globally. The data is distributed through the Namecoin network. Each new certificate-signing request transaction includes the following fields:

1. Public key of the destination peer (the ID of TTP)
2. Amount of money to be paid
3. Amount of the transaction fee to be paid
4. Link to the previous transaction
5. Sum of all transaction fees (calculated from the sum of the value of the same field of the previous transaction and the transaction fee)

When such a transaction is broadcasted to the network, all other peers must validate that it correctly refers to the previous transaction, and then they include it to the current block. An end-user seeking for the absolute value of the reputation score for some particular TTP will need to fetch only the last transaction. The algorithms that take into account the dynamics of the reputation score are able to restore the whole chain crawling the linked list of transactions. Certificate signing transactions are not different from any other type of transactions from the point of view of the

Bitcoin protocol, hence the Merkle-tree algorithm for storing such transactions can be applied too.

3.1.6 Pre-trusted third parties

The certificates of the pre-trusted third parties must be handled in a different way. For example, when a company or a university wants to distribute its own local self-signed CA certificate through all computers in the network, the reputation score becomes irrelevant for this certificate. Self-signed root CA certificates distributed locally should be kept in the separate key store and the behavior of the software must be different when handling such certificates. Such certificates are automatically granted the ultimate trust level.

Such an exception complies well with the ideology of the proposed system. The reputation score is needed only to assist an end-user with the selection of a barely known TTP among all TTPs. When a TTP is well known, then such assistance is not required.

The distribution of the pre-trusted certificates is another problem – the network administrator needs to deliver the certificate securely to each user on the network. In such case the two approaches can be combined: a local self-signed CA certificate can be cross-signed by some public TTPs, so an end-user can be confident that the certificate was not replaced during transmission.

The proposed system is fully backwards compatible with the regular SSL. Software manufactures may set up a predefined list of trusted root CAs and disable the network-querying component. However, when building such a list, a software manufacturer may rely on the reputation score.

It is very easy for a pre-trusted party to convert into a commercial notary. The new TTPs should enter commercial notary market by establishing a wide trust network as a pre-trusted entity.

3.2 Flatcoin

The proposed system Flatcoin is an extension to the Namecoin protocol together with a software interface concept. Flatcoin combines proof-of-work, cross-certification and dynamic TTP suggestion concepts. The economy of Namecoin domain registration process is extended with the entity validation processes and the economic reputation scoring system described in section 3.1.

Flatcoin uses the corresponding concept of virtual currency (FC), which peers "generate" or "mine" from proof-of-work calculations. The Bitcoin protocol limits the amount of currency available in the system, so currency inflation is impossible by design. In general, FC could be used only for the domain name registration and entity validation payments, however FC is not limited to such behavior: the peers in Flatcoin can freely exchange FCs. The market defines the exchange rate with the government-backed currencies.

In Flatcoin the user decides which root CAs to add to the keychain. The approach is similar to the web-of-trust and trust-on-first-use concepts with reputation score to assist in the trust decision. The core difference with the regular SSL is that the end-user is now given the reputation score parameter to make the trust decision.

The cross-certification allows the Flatcoin users to build the complicated trust relationship structures, so the web-of-trust becomes interconnected between entities through TPPs.

3.2.1 Processes in Flatcoin

Domain name registration

The proposed system Flatcoin derives the domain name registration process from Namecoin. It manages only gTLDs and ccTLDs so that the second-level domain names and below may use the traditional DNS hierarchical approach.

The DNS records database is independent and distributed. There are two counterparts in the process of registration: a payer (a peer paying for the domain name registration), and a payee (a peer collaborating on finding proof-of-work).

The way of adding entries to the database is the same as in Namecoin:

1. A peer broadcasts his intention to register a domain to the network
2. All other peers collect these requests into a block of transactions and start finding proof-of-work for this block
3. Once a peer finds a proof-of-work it broadcasts the calculated block to the network
4. All other peers accept the block if the proof-of-work is valid and the transactions in it are not conflicting
5. The derived proof-of-work is used for the next block, so the integrity and continuality are verified
6. In order to prevent the domain squatting by giving a lot of resources temporarily for an alternative unfair block branch, the domain name becomes effective only after N blocks are passed, where N is constant and derived a posteriori.

The domain name information is signed with the registrant public key used for the system. Updates to the network are peer-to-peer broadcasted and signed with the public key as well.

Entity validation

The entity validation process implies matching of the company or individual name to the domain name. When an end-user connects to arbitrary site, he needs to distinguish the domain names to be sure that the connection was made to the intended destination. If an end-user does not know the correct spelling of the domain name or mistypes it, he might suffer from the phishing attack. The purpose of the entity validation is to prove that, for example, internetsociety.org belongs to the Internet Society located at Reston, VA, but internetsociety.info is just some random site.

If the end-user already has the TTP in the key store that is both trusted by the site and him, then no action is required, and the site identity is easily checked through

the digital signature. If the end-user connects to a new site and there are no shared TTPs trusted by both parties, then the user is suggested to cache the certificate of the site or the certificate of the TTP, which has signed the site, or both. The user is advised by the browser software on the reputation score of TTP, its dynamics and the ranking among other TTPs.

To verify a domain the site owner needs to send a special type of transaction through the Flatcoin network – a "validation request" with transaction fee on it.

The algorithm of the certificate signing is as follows:

1. A site owner sends a certificate signing request to the desired TTP

2. The TTP sets the price for the validation process upon request. The price is comprised of the net profit and the transaction fee. The TTP half-signs the transaction and sends it to the site owner.

3. The site owner completes the transaction by signing it too. He also adds the link to the previous transaction and the sum of the amount of transaction fees paid by the TTP until now into the transaction details.

4. The site owner broadcasts the transaction to the Flatcoin network. The calculations of proof-of-work begin.

5. PoW is calculated. The peer that has calculated the correct PoW receives the transaction fee.

The process is depicted in figure 3-1. Each TTP can set up its own procedures for verification. The software vendors may encourage the TTPs with stricter procedures by giving a "green address bar" for the websites validated by such a TTP. The site owners are motivated to use two or more TTPs for their sites in order to seamlessly connect as many peers as possible. In addition to that we suggest abolishing the certificate chains in order to use all CA certificates in the same way as in the web-of-trust approach – all CAs are root. Thus, we can achieve a trust model closer to a natural one among humans.

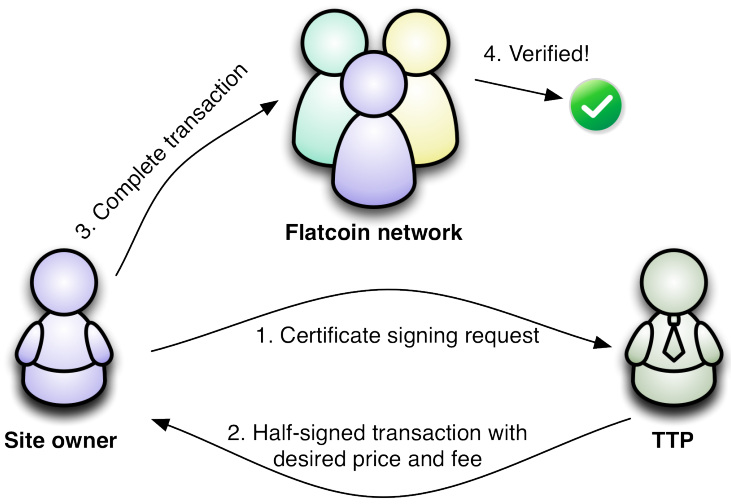

Figure 3-1: Entity validation process

Certificate revocation process

If the certificate of a site owner is compromised, he can revoke it. In regular SSL the site owner would put the certificate information on a Certificate Revocation List (CRL) or publish it via Online Certificate Status Protocol (OCSP). When a peer tries to communicate with the site owner, he queries the revocation status of the certificate using CRL or OCSP. Such a scheme is not effective because a lot of software ignores the case when the revocation status is not available.

The site owner in Flatcoin creates a broadcast message to all peers stating that the SSL certificate was breached, so that all the peers get this information immediately. The end-users can either cache this information or look it up in the Flatcoin's distributed database. The Flatcoin network is supposed to be always available to resolve DNS and SSL signing transactions, and thus the revocation information is always available too.

Removing trust relationship

If a site owner realizes that one of the selected TTPs is not reliable, he can revoke the TTP's signature (but not the whole cross-signed certificate) from the certificate. Such action is performed proactively through broadcasting the intent to the Flatcoin network. In this case, the reputation of the TTP that has issued this signature is

decreased exactly by the quantity of transaction fees paid for this signature.

System bootstrapping

When Flatcoin is brought into action, most of users will have empty webs-of-trust and TTPs will not have enough maturity level to provide users with trust decisions. The end-users are exposed to the risk of adding malicious TTP certificates to the keychain, because no trust relationships exist yet. At early stages it is hard to fill the TTPs with reputation scores. There could be several solutions to this problem:

- Convert the current CAs into Flatcoin TTPs and present them with scores per each certificate sold before. These scores will expire together with the certificate expiration date (in the similar way as they were gained through the regular Flatcoin transactions).

- Combine the current SSL model with the Flatcoin trust model and let an end-user decide on trust each time. Then withdraw the current SSL model from the browser after two-three years (normal SSL certificate lifespan).

Adding CA to the keychain at user side

When a user accesses an Internet host, a list of CAs that have signed the certificate are displayed. The reputation score defines the sorting of the CAs' list. User is suggested to add the most trusted CA for this host to the key store, so that the future connections to this host and other hosts using the same CA become automatically trusted. The user may apply different filters based on his policy (e.g., include only top 30 certificates of the whole system and reject all other).

3.2.2 Stakeholders' incentives in Flatcoin

This subsection describes the behavior of each participant in the Flatcoin system.

End-user

An end-user's behavior does not imply paying FC. A user might perform some proof-of-work calculations to get FCs to be able to create domains in future, but most users would just prefer a simple browsing. The users communicate with different sites, add the TTP certificates, and reach the condition when all their sites are trusted. In most cases the trust decisions on TTPs could be made in favor of the user by the software vendors.

Site owner

A site owner wants all his users to trust the information he provides in the certificate, therefore the site owner is motivated to add as many signatures to his certificate as he could. These signatures should also be from the most ubiquitous TTPs. However, it is up to the site owner which TTPs to use – if some widespread TTP shows unexpected behavior, the site owner can switch to a less popular TTP and thus raise its rating.

TTP

The strategy of a TTP is the most difficult in Flatcoin – a regular TTP wants to maximize its income, while maintaining a high reputation score. The main principle is that a TTP should not work on its reputation score, but it should work on the real world reputation and trust, so that it will have customers even when the Flatcoin reputation score is low or the fee is high. The economics behind TTP business is described in section 3.1.

Software vendor

A software vendor in Flatcoin does not need to make decisions on including root TTP certificates built into its product. However, there are several requirements for the software look, so that an end-user will be able to distinguish TTPs and cross-certification schemes. Also it is the responsibility of the software vendor to develop

some correct and secure web-of-trust traversal algorithm, so that the leaf certificates will be trusted through the long and complex certification paths such as mesh or bridge. It is also up to the software vendor to distinguish the TTPs based on their certification requirements. For example, the TTPs, which require a security audit to be performed and a passport to be shown, get a green address bar in the web browser, but the TTPs validating only the e-mail address get a blue bar.

Malicious user

A malicious user has two incentives in the Flatcoin network:

- Obtain the private key of the attack target
- Unfairly get a high reputation score and validate phishing sites

The first problem corresponds to the key management problem, which is not tackled by this proposal. The latter problem of self-promoting the reputation score is mitigated by the expense of buying or mining FCs, since maintaining a growing reputation score for a long time is expensive task.

National government

A national government wants all domains in the country to comply with its laws. If a domain is breaking the law, the government wants to suspend or remove the domain registration. In Flatcoin the proof-of-work concept is used only for top-level domain registration, so nothing prevents the government from registering a ccTLD and suggesting that all citizens register domain names under this ccTLD. The government would also want to set up a TTP and sign each domain, which it considers benign. The citizens can apply a setting to access the websites, which have the signature from the government. For example, such a whitelisted approach can be used in schools – a pupil using the school computer can only access the resources, which have a valid signature from, for example, the Ministry of Education. On the other hand, if the school is private and wants to use other resources as well it may

retain their decision. In this way the national government can only support smaller institutions' governance but cannot force users to use such settings.

3.3 Notarized social networks primer

In networks with only one trust anchor, such as Twitter, sometimes it is required to provide additional field verification of the profile. However, the capacity of the company does not allow verifying all users. Using the financial reputation system described above together with regular Bitcoin and a keyserver instead of Flatcoin it is possible to build the same trust model. Such a trust model will not be completely distributes as Flatcoin, but since it already has a trust anchor it is possible to align trust to this anchor.

When some user wants to be validated, he creates an OpenSSL key pair and generates an X.509 certificate signing request (CSR) with common name equal to the Twitter handling name, and all other fields correspondingly (real name, address, etc.). The user creates the Bitcoin transaction and includes its ID number into the certificate too.

Then the user sends this CSR to the chosen TTP and pays the Bitcoin transaction with the amount and transaction handling fee specified by the TTP. The user and the TTP exchange oAuth tokens to verify each other identities, then the TTP performs extended validation.

The received certificate is stored to the keyserver. It may be some common shared trusted server or a private server (specified in the "Homepage" profile field of the notary). However, the use of the service trust server is recommended (in case of Twitter, it should be managed by Twitter Inc., otherwise the number of fixed trust anchors increase). The fingerprint of the certificate is stored in the "Bio" field of the user.

The keyserver pre-calculates the reputation score and keeps all certificates indexed by the TTP ID. When a checking user accesses the keyserver, he might request the pre-calculated reputation score value per TTP, or the whole transaction log with this

TTP to calculate the reputation score manually. Using the reputation score the user makes the trust decision assisted with the client software. Given that the transaction log is verifiable through Bitcoin network, it is hard to mangle the reputation score on the trust server.

Chapter 4

Feasibility verification

For the validation of the assumption that richer sellers are more trustworthy an experiment was established. Since the sales data from the SSL CAs is not publicly available, an analogy was expanded to a regular sales with buyers feedbacks. Generally, the reputation calculation systems are evaluated against eCommerce web-sites such as eBay [17] or rating web-sites such as Epinions [19]. eBay is a good choice to compare the peer income and the reputation score, because the transaction data contains both the price of an item and the feedback value. Aside of eBay, the Rakuten Ichiba data set containing user reviews and price of items was obtained. The analysis these two data sets is described below.

4.1 eBay data set

The data was fetched using information from the pages as follows:

```
http://feedback.ebay.com/ws/eBayISAPI.dll?ViewFeedback2&ftab=AllFeedback
&userid=[USERNAME]&iid=-1&de=off&items=200&interval=365&which=all&
page=[PAGENUM]
```

The resulting data set parameters and their values are described in table 4.1.

Table 4.1: eBay data set parameters

Parameter	**Value**
Crawling period	Jan 7, 2013 – Jan 9, 2013
Feedback data period	12 months
No. of parallel crawlers	5
No. of users	2,394,736
No. of feedbacks	3,724,793
Size of the resulting DB	624 MiB (plain-text)
User with the highest reputation score	everydaysource

Table 4.2: The "peer" table structure

Field	**Description**
id	Unique peer id (primary key)
name	Name of the user used in eBay
ebay_reputation	The reputation of the user displayed at eBay
calc_reputation	Calculated reputation from the positive feedbacks (past 12 months)
capital	Amount of money received by this peer (past 12 months)
totalpages	Total number of web pages to crawl for this user
lastpage	Last page crawled for this user
worker	Process ID of the crawler program (to avoid race condition)
baduser	Marker for users with private feedbacks (excluded from the experiment)

4.1.1 Data fetching methodology

As described in [17] the eBay data set was acquired by crawling the feedback data from a very reputable user (reputation score is more than 10,000) as described in [22]. According to [23] the most reputable user on eBay in August 2008 was everydaysource. By the time of crawling the reputation of this user was 2,439,255, which is significantly above 10,000 points as recommended in [17]. Starting from the above user 351,298 feedbacks were analyzed giving the second wave of 309,399 additional users to crawl. The feedback data of these users was also analyzed giving another set of around 2M users and feedbacks. Due to the limitations of the crawler (only 5 concurrent threads from 5 different IP-addresses to avoid blocking) data gathering was stopped after the second wave was crawled completely. The fetched data was gathered to the centralized PostgreSQL database in the format shown in tables 4.2, 4.3, and 4.4.

Table 4.3: The “feedback” table structure

Field	Description
id	Unique feedback id (primary key)
from_id	Foreign key to peers table id (buyer)
to_id	Foreign key to peers table id (seller)
amount	Amount of money in the original currency
currency	Currency of the transaction
usdamount	Amount of money in US dollars
ebay_trid	eBay internal transaction ID to guarantee uniqueness
timestamp	Date and time of the transaction
item	Name of the purchased item
feedback_value	Positive feedback = 1, Neutral = 0, Negative = -1

Table 4.4: The “exchange” table structure

Field	Description
id	Unique currency id (primary key)
currency	Currency code
rate	Rate (received from Yahoo.Finance)
time	Date of the exchange rate fetch

4.1.2 Data set preparation methodology

The proposed system for reputation score calculation does not imply negative feedbacks, but it does imply the feedback expiration. The received feedback data were filtered by the following condition:

```
"amount">'$0.00' and "value"=1
```

An assumption was made that buyers with negative or neutral feedbacks would rather have a full refund, so that the transaction had never occurred. Since eBay transactions occurred in different currencies the conversion was performed using the data of Yahoo.Finance at January 9th, 2013. The currency rate fluctuations were considered insignificant for the sake of the experiment. In the proposed system the currency is consistent between transactions. After all transactions were filtered and converted to the US dollars, the fields *capital* and *calc_reputation* of table *peers* were populated with the SUM of the transaction USD amount and value for each peer:

```sql
UPDATE peers
  SET "capital"=(
    SELECT SUM("usdamount")
      FROM feedbacks
          WHERE
            "usdamount" > '$0.00'
          and
            "value"=1
          and
            "to_id"=peers.id
    )
  WHERE
    "lastpage">0
  and
    "totalpages"="lastpage"
```

When the peers data were populated the following parameters were analyzed:

1. Correlation between capital and calculated reputation
2. Correlation between capital rank and calculated reputation rank of a peer
3. Amount of capital at each capital rank
4. Amount of reputation at each reputation rank
5. Global growth of capital and reputation normalized by the max value.

4.1.3 Data set analysis

The correlation between capital and calculated reputation is shown in figure 4-1. The X-axis represents gross capital received during past 12 months in US dollars by each peer. The Y-axis represents the reputation score counted in votes. Both axes are in the logarithmic scale. Since the transactions were filtered by only positive feedbacks, it also represents the number of transactions used in this experiment.

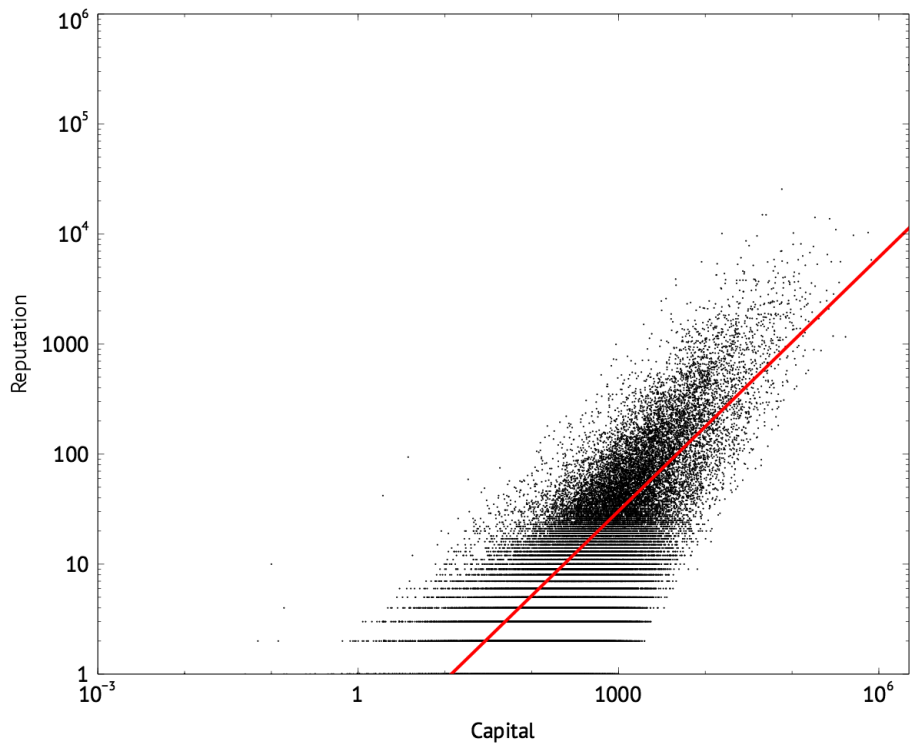

Figure 4-1: The correlation between peer capital and calculated reputation

The correlation is positive. The distribution converges as the reputation grows. The correlation coefficient calculated using the Pearson method is **0.775**. The top sellers gain similar reputation, while the peers with low reputation sell few random items with different prices. Peers have discrepancy in capital over 3 orders.

The correlation between capital rank indexes and calculated reputation rank indexes of peers is shown in figure 4-2. The X-axis represents the rank indexes of peers ordered by gross capital. Rank index 1 represents the richest peer. Dots located in the right part of the graphs represent low-income peers (the majority). The Y-axis represents the rank indexes of peers ordered by the calculated reputation value. Rank index 1 represents the most trustworthy peer. The dots in the bottom of both graphs represent more trustworthy peers according to their feedback voting value. The correlation is also positive, indicating that capital and reputation are connected. The Pearson correlation coefficient is **0.703**. The lower income and reputation peers have bigger discrepancy.

The correlation between capital rank index and the gross capital of a peer is shown

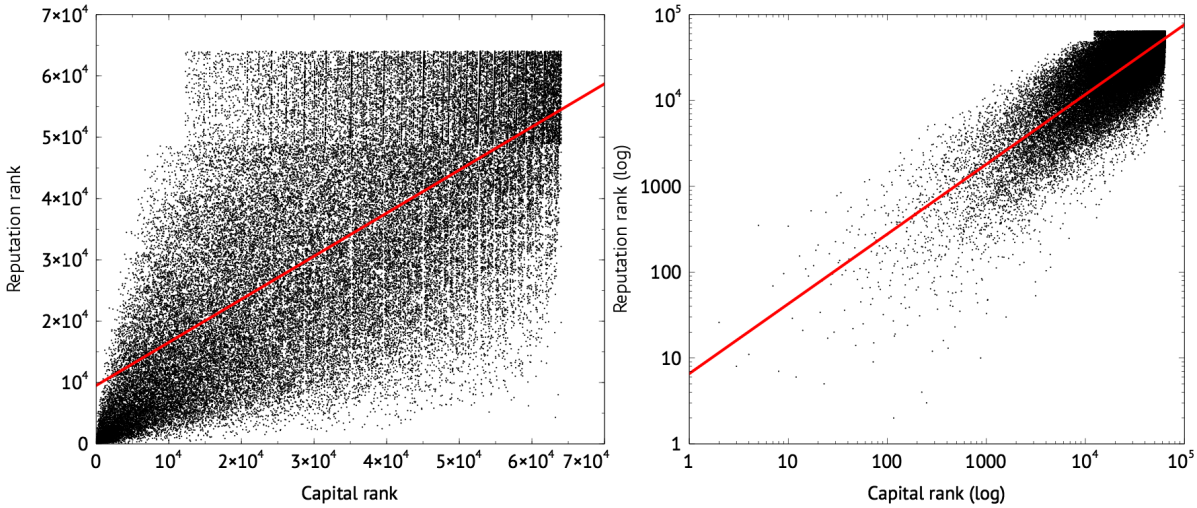

Figure 4-2: The correlation between capital rank index and calculated reputation rank index (left – log scale, right – regular scale)

Table 4.5: eBay capital distribution power law fitted parameters

Parameter	**Value**
X_{min}	839
α	1.920
δ	0.020

in figure 4-3. The X-axis represents the capital rank index, the lower the value – the richer the peer. The Y-axis represents the capital in US dollars received per each peer in log scale.

The distribution of capital between peers is skewed significantly to the left. The majority of users gained less than 1,000 USD during 12 months. It resembles figure 3(b) obtained in [17]. The fitted parameters of power-law distribution for these data are shown in table 4.5.

The correlation between reputation rank index and the calculated reputation of a peer is shown in figure 4-4. The X-axis represents the reputation rank index, the lower the value – the more trustworthy the peer. The Y-axis represents the calculated reputation received per each peer in the log scale.

The reputation score is more discrete than capital (the minimum unit is 1), and thus the tail of the distribution looks teared. The power-law distribution parameters for these data are shown in table 4.6.

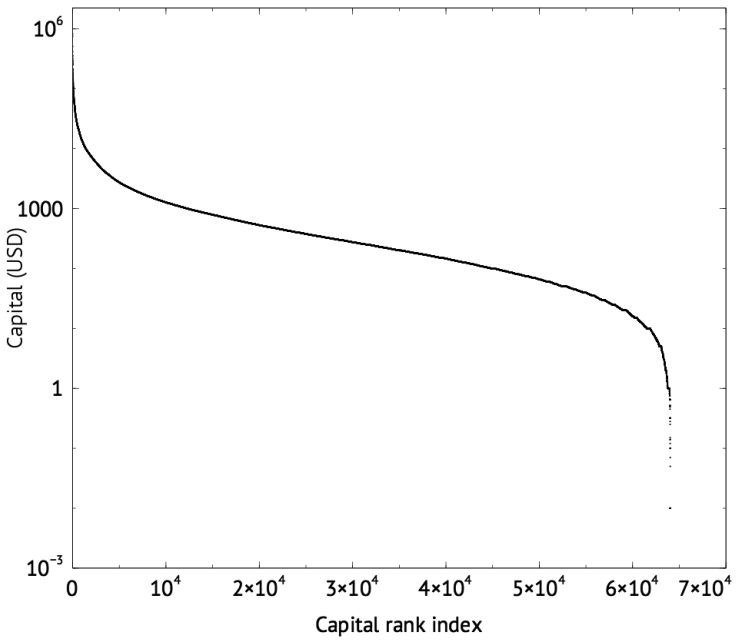

Figure 4-3: The correlation between capital rank index and capital value

Table 4.6: eBay reputation distribution power-law fitted parameters

Parameter	**Value**
X_{min}	866
α	2.450
δ	0.032

The graph shown in figure 4-5 demonstrates the dependency between capital and reputation score in one year time window. The X-axis represents time (hourly data points for 1 year), and the Y-axis represents the flow of reputation and capital normalized by the maximum value. The last data point represents 100% of capital and reputation obtained by all peers by the end of the 12 months. The difference between two lines is so small that both graphs almost merge into one, but a small discrepancy could be observed in the middle.

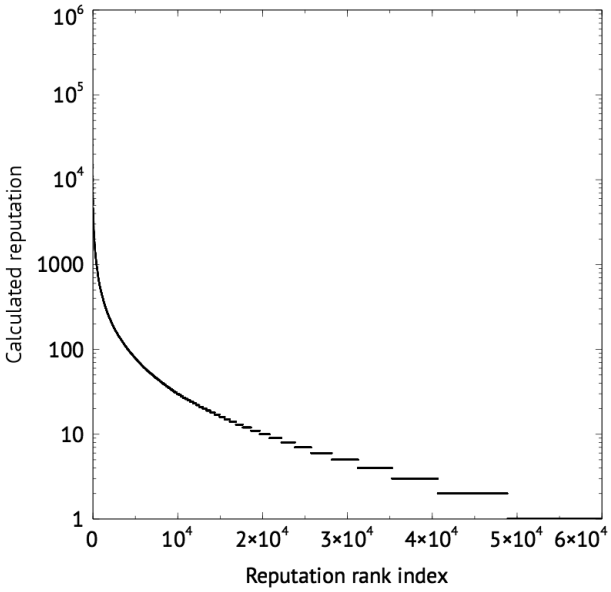

Figure 4-4: The correlation between reputation rank index and reputation value

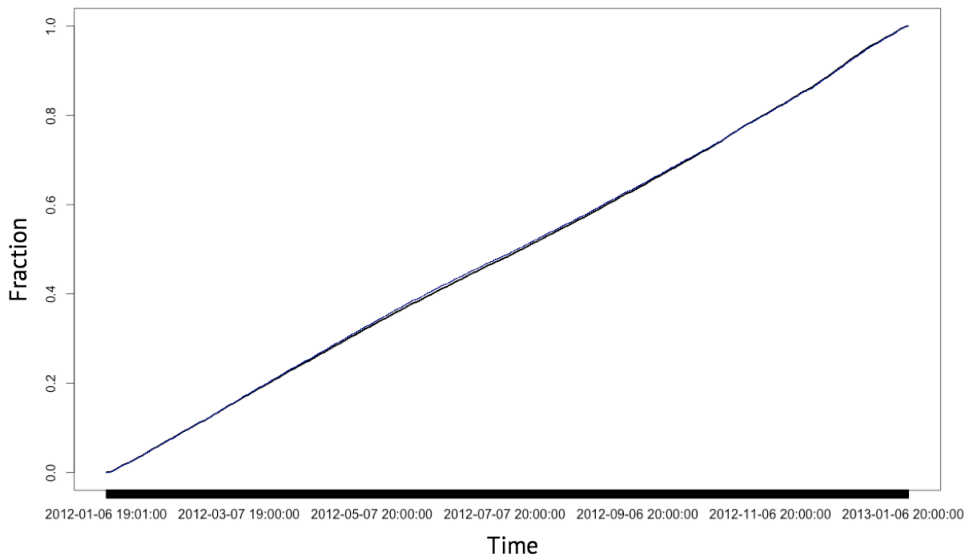

Figure 4-5: Flow of reputation and capital during the 12 months (all peers)

Table 4.7: Rakuten data set parameters

Parameter	**Value**
Feedback data period	2011/07/28
No. of users	218,232 (uncertain; anonymized data set)
No. of feedbacks	16,606,181
Size of the resulting DB	10 GiB
User with the highest reputation score	Rakuten Books

4.2 Rakuten data set

In order to validate the results of the eBay experiment, a data set from the biggest Japanese eCommerce was requested. Rakuten Ichiba is a marketplace for merchants selling different types of goods. Buyers rate the seller from 1 to 5 every time a purchase has been made. The data set was provided officially for academic institutions from Rakuten Institute of Technology, and therefore the problem of crawling quality was avoided in this experiment. The data set parameters are described in table 4.7.

4.2.1 Data set preparation methodology

The reputation scoring in Rakuten Ichiba is different from eBay: instead of positive, neutral and negative there are scores from 1 to 5. It was assumed that scores 1 and 2 match negative feedbacks, 3 is neutral, when 4 and 5 are positive. The filtering of the data set was performed: all transactions bigger than zero yen and reputation score more than 3 were included.

4.2.2 Data set analysis

Correlation between reputation value and capital is shown in figure 4-6. The X-axis represents gross capital received during past 12 months in Japanese yen by each peer. The Y-axis represents the reputation score – the sum of 4s and 5s to this peer. Both axes in the logarithmic scale. The absolute values were not comparable to eBay data since the reputation score calculation and the transaction currency were different.

The correlation coefficient was calculated using the Pearson method. Correlation

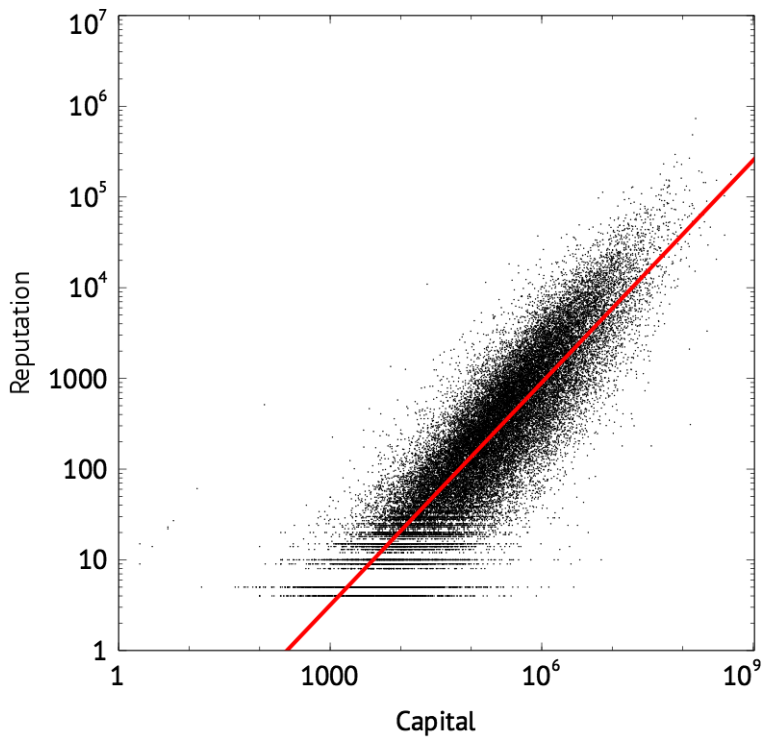

Figure 4-6: The correlation between peer capital and calculated reputation

coefficient between capital and reputation in the data set is **0.725**, which is close to the value obtained from the eBay data set – **0.775**.

The correlation between capital rank index and calculated reputation rank index of every peer is shown in figure 4-7. The X-axis represents the rank indexes of peers ordered by gross capital. Rank index 1 represents the richest peer. Dots located in the right part of the graphs represent low-income peers (the majority). The Y-axis represents the rank indexes of peers ordered by the calculated reputation value. Rank index 1 represents the most trustworthy peer. The dots in the bottom of both graphs represent more trustworthy peers according to their feedback voting value.

The above result indicated in figure 4-7 is similar to the eBay data: positive correlation, significant difference between number of low income sellers and number of high income sellers. The Pearson correlation coefficient is **0.894**.

Capital distribution graph has similar characteristics observed in the eBay experiment, however the inclination of the middle part is steeper (parameter α is higher). The fitted parameters of power-law are shown in table 4.8.

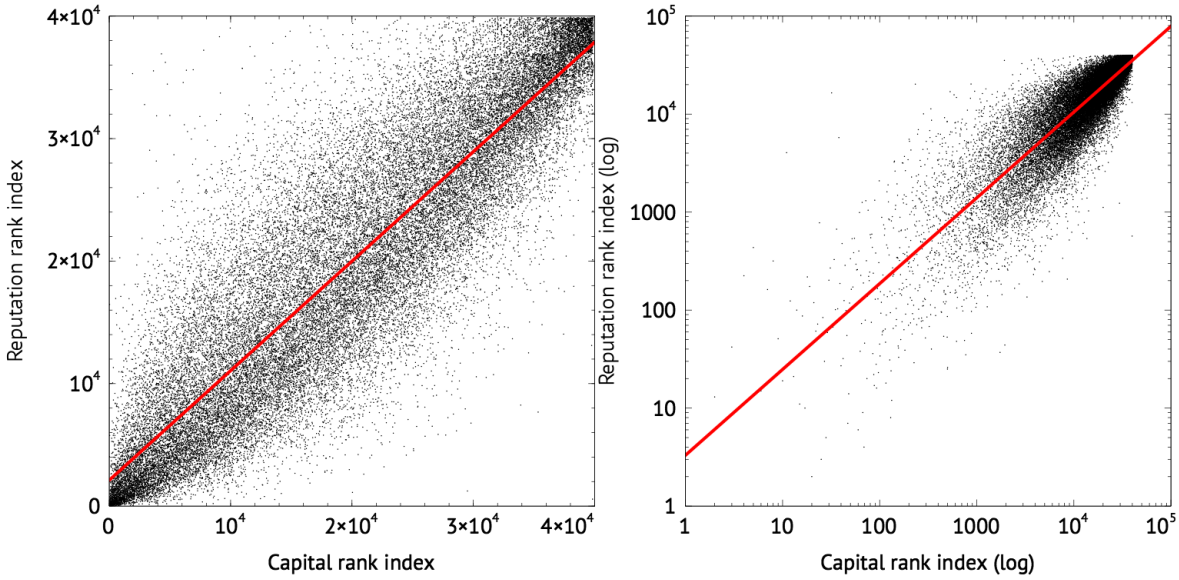

Figure 4-7: The correlation between capital rank index and calculated reputation rank index (left – log scale, right – regular scale)

Table 4.8: Rakuten capital distribution power-law fitted parameters

Parameter	**Value**
X_{min}	17,653,749
α	2.437
δ	0.023

The correlation between reputation rank index and the calculated reputation of every peer is shown in figure 4-9. The X-axis represents the reputation rank index, the lower the value – the more trustworthy the peer is. The Y-axis represents the calculated reputation received per each peer in the log scale.

The graph looks very similar to the one obtained in the eBay experiment. Hence, the crawling data in eBay experiment are valid. The power-law distribution parameters for these data are shown in table 4.9.

Table 4.9: Rakuten reputation distribution power-law fitted parameters

Parameter	**Value**
X_{min}	2,108
α	2.360
δ	0.023

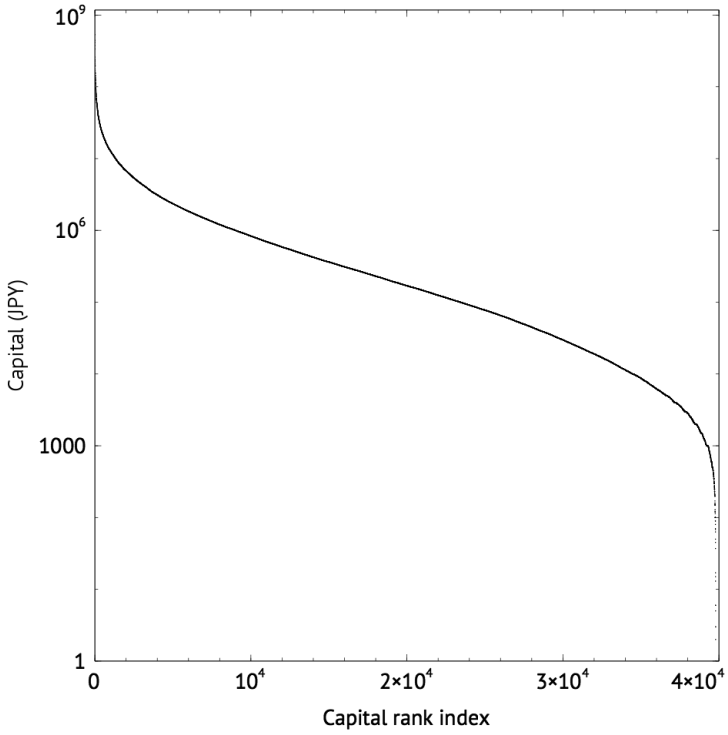

Figure 4-8: The correlation between capital rank index and capital value

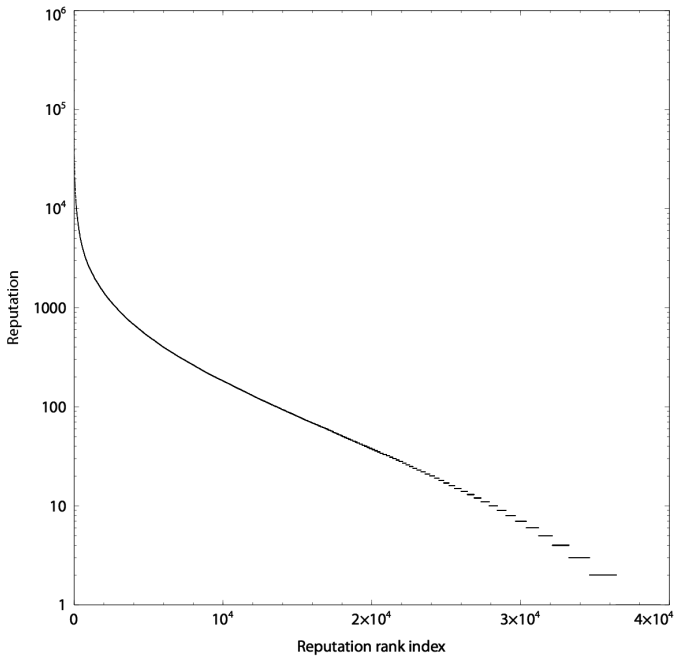

Figure 4-9: The correlation between reputation rank index and reputation value

4.2.3 Summary

Three main observations were found from the experiment:

1. Capital and reputation were related with a high correlation coefficients 0.72–0.77.

2. Capital and reputation adhere partial power-law distribution in both Rakuten and eBay data sets.

3. The discrepancy between capital and reputation ranks is lower for high income peers.

Our initial assumption to replace the feedback voting reputation parameter with just capital could be justified with the following:

- The power-law distribution shape remains

- It is more expensive to increase someone's reputation score, thus Sybil attackers are less financially motivated

- The top peers (as known as "power nodes" in [17]) are not affected by the change of the parameter.

Chapter 5

Conclusion, limitations, and future work

This chapter concludes the thesis. In section 5.1, the conclusion is made in connection to the objectives described in chapter 1. Limitations of current proposal are discussed in section 5.2. In section 5.3, the problems that should be tackled in the future are enumerated.

5.1 Conclusion

A new reputation system based on financial performance and its tecnical implementation Flatcoin were proposed.

Different aspects of Flatcoin were discussed on how to solve the rigidness problem with the current DNSSEC and SSL architecture. By making trust anchors optional and suggesting them dynamically based on their reputation score Flatcoin eliminates the rigid trust relationships. Every peer gets a key pair upon creation of a Flatcoin wallet ready to use for certification of other peers. Flatcoin wallet key pair can also be used to generate dependent certificates for web (X.509), mail (DKIM or S/MIME) and client authentication. In such a way, the penetration of secure communications over the Internet could be expanded, because it is available to each user by default.

The market of Flatcoin does not have barriers to entry in order to become a

TTP. TTP's revenue information is also open, so that Flatcoin market is closer to the perfect competition model and the oligopoly of CAs is diminished. An economy-based reputation scoring system and a TTP suggestion algorithm is introduced to help peers with an empty web-of-trust to fill it with the most trustable TTPs.

The objectives set in chapter 1 are discussed below.

Identify the ways to localize the impact of TTPs in SSL and DNSSEC on the Internet.

Distributed timestamping for domain name registration and zero default certificates configuration for SSL CAs reduce the number of pretrusted entities in the Internet to zero. The algorithm for adding certificates (see section 3.2.1) localizes the trust connections of the end-user to the scope of the network he generally uses. For example, if the end-user visits only Facebook and Twitter (like the author of this thesis), then he needs only Verisign's certificate in his key store. Note that the certificate only impacts the extended validation attributes, so that even with zero certificates end-users are safe if they do not make mistakes in domain names.

Develop a robust and secure way to pin certificates to IP-addresses and entity names.

Certificate pinning to the IP-address without prayer condition is described in section 3.2.1. The pinning occurs at the moment of domain name registration (not at the moment of the first access as in other proposals), so that there is no threat related to prayer condition. The certificates are stored in peer-to-peer manner, and thus the first objective is achieved.

Reduce the number of the built-in CA certificates to the minimum required level defined by the end-user.

Reducing the number of certificates is a hard problem, because there is no criteria on what certificate should be included in the default key store and what should not. In this thesis, the bottom-up approach was suggested – to include zero certificates and add them as deemed necessary. Since the proposal suggests that the cross-signing must be used, only the most trusted certificates are added. The determination of "the most trusted" is left up to the decision-making algorithms such as PowerTrust [17].

Since these algorithms cannot be applied without some reputation score, the financial reputation score was invented.

Provide some independent dynamic metric to assist end-users in the decision whether to include a CA certificate into the key store or not.

The proposed financial reputation score used in Flatcoin is more robust against Sybil attacks than peer feedbacks (voting), since it increases the cost of the attack. The score is implicit, i.e., does not require direct evaluation by the buyer, so that the maintenance burden is significantly lower than in feedback systems. The proposed metric is time-bound and expiring, therefore, the trust could be measured dynamically.

5.2 Limitations of the proposal

Flatcoin inherit different problems from Bitcoin. These problems are mainly related to operation costs and performance limitations. The proposed solutions do not define any specific approaches used in the underlying proof-of-work calculation. If the performance of Bitcoin (and accordingly Namecoin) could be improved, these problems would be solved.

5.2.1 Performance

If we have migrated all registered domains in .com and .net zones to Flatcoin's TLD zone, it would be up to 114 million domain names by now (stats of .com and .net combined as of March 2012 according to the domain name industry brief by Verisign [24]). There are several research studies on DHT-like domain name search in distributed P2P networks. One such study is Beehive [25], which allows searching with O(1) complexity in the best case and O(log N) in the worst. The database of Flatcoin can be distributed using Merkle-tree or DHT similarly to Beehive.

The reputation score calculation may be performed simply as calculating all transactions to the related peer. This might lead to the performance decrease with the big number of transactions. To reduce the burden of recalculating the sum of the trans-

action fees it is suggested to keep the absolute score in each transaction together with the link to the previous transaction. Such an approach would take only one lookup of the previous transaction. Since the previous transaction was verified upon the proof-of-work calculation, there is no need to trace back all the chain of certificate-request transactions.

5.2.2 Electricity consumption

Wasteful electricity consumption will increase significantly once the system is deployed widely. Proof-of-work calculations require a lot of energy to power CPUs and GPUs. The calculations produce additional heat as well, so that the cooling system will consume electricity too. The proof-of-work algorithm needs to be improved to provide some useful and efficient calculations, such as protein folding. Heat generated from the calculations can be used to warm the buildings in winter.

5.2.3 Domain names privacy

The privacy is a big problem in distributed databases [26]. The network becomes aware of the new top-level domain registration after a few moments of its announcement. Rivals can use this information to gain a market advantage. It is not recommended to register top-level domains if such a concern arises.

5.2.4 Cost of deployment

Flatcoin does not require any centralized servers to be deployed for its operation. The cost is evenly split among all stakeholders. However, for the legacy system support it might be needed to maintain several recursive DNS resolvers compatible with both legacy DNS and Flatcoin.

5.2.5 Content delivery networks

Currently content delivery networks take advantage of recursive domain name resolution and ANYCAST messages. Flatcoin uses the distributed DB and the in-

formation about the location of each domain's IP-address cannot be used effectively. Some new algorithm for content delivery networks must be developed. The problems described above are beyond the scope of this paper, however, it is required to consider them when building an implementation of Flatcoin.

5.3 Future work

Deeper analysis from the cognition, cooperation and coordination perspective is required to understand the dynamics of the model should it be adopted. Agent-based simulation of all stakeholders should be performed and analysed.

Full game theory model and pay-off function should be derived in order to understand that there is no potential exploitation vectors in the proposed models, which would lead to the collapse of the economic component of the system.

Appendix A

Source codes

A.1 Listing of power law fit algorithm

The source code below is generously provided by Laurent Dubroca.

```
library(VGAM)    # zeta function

plfit<-function(x=rpareto(1000,10,2.5),method="limit",value=c(),finite=
    FALSE,nowarn=FALSE,nosmall=FALSE){
   #init method value to NULL
    vec <- c() ; sampl <- c() ; limit <- c(); fixed <- c()

#  test and trap for bad input
#
    switch(method,
      range = vec <- value,
      sample = sampl <- value,
      limit = limit <- value,
      fixed = fixed <- value,
      argok <- 0)

    if(exists("argok")){stop("(plfit) Unrecognized method")}

    if( !is.null(vec) && (!is.vector(vec) || min(vec)<=1 || length(vec)
        <=1) ){
```

```
      print(paste("(plfit) Error: ''range'' argument must contain a vector > 1; using default."))
      vec <- c()
    }
    if( !is.null(sampl) && ( !(sampl==floor(sampl)) ||  length(sampl)>1 || sampl<2 ) ){
      print(paste("(plfit) Error: ''sample'' argument must be a positive integer > 2; using default."))
      sample <- c()
    }
    if( !is.null(limit) && (length(limit)>1 || limit<1) ){
      print(paste("(plfit) Error: ''limit'' argument must be a positive >=1; using default."))
      limit <- c()
    }
    if( !is.null(fixed) && (length(fixed)>1 || fixed<=0) ){
      print(paste("(plfit) Error: ''fixed'' argument must be a positive >0; using default."))
      fixed <- c()
    }

#  select method (discrete or continuous) for fitting and test if x is a vector
    fdattype<-"unknow"
    if( is.vector(x,"numeric") ){ fdattype<-"real" }
    if( all(x==floor(x)) && is.vector(x) ){ fdattype<-"integer" }
    if( all(x==floor(x)) && min(x) > 1000 && length(x) > 100 ){ fdattype <- "real" }
    if( fdattype=="unknow" ){ stop("(plfit) Error: x must contain only reals or only integers.") }

#
#  end test and trap for bad input


#  estimate xmin and alpha in the continuous case
```

```
#
    if( fdattype=="real" ){

     xmins <- sort(unique(x))
     xmins <- xmins[-length(xmins)]

     if( !is.null(limit) ){
       xmins <- xmins[xmins>=limit]
     }
     if( !is.null(fixed) ){
       xmins <- fixed
     }
     if( !is.null(sampl) ){
       xmins <- xmins[unique(round(seq(1,length(xmins),length.out=sampl))
           )]
     }

     dat <- rep(0,length(xmins))
     z   <- sort(x)

     for( xm in 1:length(xmins) ){
       xmin <- xmins[xm]
       z    <- z[z>=xmin]
       n    <- length(z)
       # estimate alpha using direct MLE
       a    <- n/sum(log(z/xmin))
       # truncate search if nosmall is selected
       if( nosmall ){
        if((a-1)/sqrt(n) > 0.1){
            dat <- dat[1:(xm-1)]
            print(paste("(plfit) Warning : xmin search truncated beyond",
                xmins[xm-1]))
            break
        }
       }
       # compute KS statistic
```

```
      cx    <- c(0:(n-1))/n
      cf    <- 1-(xmin/z)^a
      dat[xm] <- max(abs(cf-cx))
    }

    D     <- min(dat)
    xmin  <- xmins[min(which(dat<=D))]
    z     <- x[x>=xmin]
    n     <- length(z)
    alpha <- 1 + n/sum(log(z/xmin))

    if( finite ){
      alpha <- alpha*(n-1)/n+1/n # finite-size correction
    }
    if( n<50 && !finite && !nowarn){
      print("(plfit) Warning : finite-size bias may be present")
    }

   }
#
#  end continuous case

#  estimate xmin and alpha in the discrete case
#
   if( fdattype=="integer" ){

    if( is.null(vec) ){ vec<-seq(1.5,3.5,.01) } # covers range of most
        practical scaling parameters
    zvec <- zeta(vec)

    xmins <- sort(unique(x))
    xmins <- xmins[-length(xmins)]

    if( !is.null(limit) ){
      limit <- round(limit)
      xmins <- xmins[xmins>=limit]
```

```
  }

  if( !is.null(fixed) ){
    xmins <- fixed
  }

  if( !is.null(sampl) ){
    xmins <- xmins[unique(round(seq(1,length(xmins),length.out=sampl))
        )]
  }

  if( is.null(xmins) || length(xmins) < 2){
    stop("(plfit) error: x must contain at least two unique values.")
  }

  if(length(which(xmins==0) > 0)){
    stop("(plfit) error: x must not contain the value 0.")
  }

  xmax <- max(x)
   dat <- matrix(0,nrow=length(xmins),ncol=2)
     z <- x
  for( xm in 1:length(xmins) ){
    xmin <- xmins[xm]
    z    <- z[z>=xmin]
    n    <- length(z)
    # estimate alpha via direct maximization of likelihood function
    #  vectorized version of numerical calculation
    # matlab: zdiff = sum( repmat((1:xmin-1)',1,length(vec)).^-repmat(
        vec,xmin-1,1) ,1);
    if(xmin==1){
      zdiff <- rep(0,length(vec))
    }else{
     zdiff <- apply(rep(t(1:(xmin-1)),length(vec))^-t(kronecker(t(
         array(1,xmin-1)),vec)),2,sum)
    }
```

```
      # matlab: L = -vec.*sum(log(z)) - n.*log(zvec - zdiff);
      L <- -vec*sum(log(z)) - n*log(zvec - zdiff);
      I <- which.max(L)
      # compute KS statistic
      fit <- cumsum((((xmin:xmax)^-vec[I])) / (zvec[I] - sum((1:(xmin-1)
          )^-vec[I]))))
      cdi <- cumsum(hist(z,c(min(z)-1,(xmin+.5):xmax,max(z)+1),plot=
          FALSE)$counts/n)
      dat[xm,] <- c(max(abs( fit - cdi )),vec[I])
    }
    D     <- min(dat[,1])
    I     <- which.min(dat[,1])
    xmin  <- xmins[I]
    n     <- sum(x>=xmin)
    alpha <- dat[I,2]

    if( finite ){
      alpha <- alpha*(n-1)/n+1/n # finite-size correction
    }
    if( n<50 && !finite && !nowarn){
      print("(plfit) Warning : finite-size bias may be present")
    }

  }
#
#  end discrete case

#  return xmin, alpha and D in a list
  return(list(xmin=xmin,alpha=alpha,D=D))
}
```

A.2 Listing of ebay crawler

```
use DBI;
use LWP::UserAgent;
use feature 'say';
use strict;
use warnings;

my $ua = LWP::UserAgent->new;

my $dbh = DBI->connect("DBI:Pg:dbname=onebyone;host=tochka.cnl.t.u-tokyo
    .ac.jp", "postgres", "nightstreetlight", {'RaiseError' => 1}) or die
     "Coundn't connect to postgres! Aborting\n";
#my $updatepeer = $dbh->prepare("INSERT INTO peers(name, reputation,
    capital, totalpages, lastpage) values(?, ?, 0, ?, ?) RETURNING id");

# name, reputation, capital, total pages, last page
my $updatepeer = $dbh->prepare("SELECT new_peer(?,?,0,?,?)");
my $updatefeedback = $dbh->prepare("INSERT INTO feedbacks(from_id,to_id,
    amount,currency,ebay_trid,timestamp,item,value) values
    (?,?,?,?,?,?,?,?)");
my $updatepage = $dbh->prepare("UPDATE peers SET lastpage = ? WHERE id=?
    ");

# RegExps
# regThisPeer: current peer name & reputation
#my $regThisPeer = qr!<b class="g-hdn">Member id </b><span class="mbg-nw
    ">([^<]+)</span></a> <span class="mbg-l"> \( (\d+)<img src="!;
my $regThisPeer = qr!<b class="g-hdn">Member id </b><span class="mbg-nw"
    >([^<]+)</span></a> <span class="mbg-l"> \( (\d+)<img src="!;


my $regEmptyFeedback = qr!<b class="g-hdn">Member id </b><span class="
    mbg-nw">([^<]+)</span></a> <span class="mbg-l"></span></div></td></
    tr><tr><td><div><div id="v4-1Oly_Outer!;

```

```
# regPeer: $1 - value, $2 - username, $3 - reputation, $4 - timestamp,
    $5 - transaction id, $6 - item, $7 - currency, $8 - amount
my $regPeer = qr!<img [^>]+ alt="([^"]+)"></td><td>[^<]+</td><td nowrap=
    "nowrap" id="memberBadgeId">Buyer: <div class="mbg"><a title="Member
     id [^"]+" href="http://myworld.ebay.com/[^"]+"><b class="g-hdn">
    Member id </b><span class="mbg-nw">([^<]+?)</span></a> <span class="
    mbg-l"> \( <a class="mbg-fb" title="Feedback Score Of ([0-9]+)" href
    ="[^"]+"><b class="g-hdn">Feedback Score Of</b> \d+</a><img [^>]+>\)
     </span> <span class="mbg-l"></span></div></td><td nowrap="nowrap"
    >([^<]+)</td><td><a name="(\d+)"></a><a href="[^"]+"><img [^>]+></a
    ></td></tr><tr class="bot"><td> </td><td>([^<]+)</td><td>([A-Z
    ]+) .?([\d.]+)</td>!;
# regPages: $1 - number of pages
my $regPages = qr!<span id="PC_pagination1" class="pg-cpp">\d+</span><
    span>of (\d+)</span>!;

sub parseUser($$){
        my $user = shift;
        my $page = shift;

        my $response = $ua->get('http://feedback.ebay.com/ws/eBayISAPI.
            dll?ViewFeedback2&ftab=AllFeedback&userid='.$user.'&iid=-1&
            de=off&items=200&interval=365&which=all&page='.$page);
        my $content = $response->decoded_content;
        my $filename = $user;
        $filename=~s/[^A-Za-z0-9_-]//g;
        if($filename ne $user) {
                open Z, ">:utf8","cache/$filename-realname.txt";
                print Z $user;
                close Z;
        }

        open F, ">:utf8", "cache/${filename}.$page.html";
        print F $content;
        close F;
```

```
48          # Figuring out how many pages we have
            my($pagenum) = $content=~$regPages;
50          say "User $user has $pagenum pages";

52          # Getting this user params
            if($content=~$regThisPeer || $content=~$regEmptyFeedback){
54                  my $name = $1;
                    my $rep = $2;
56                  if(!defined($rep)){
                            say "[-] Reputation is not counted yet?";
58                          $rep = 0;
                    }
60
                    if($rep eq "private") {
62                          say "[-] Reputation is private.";
                            $rep = 0;
64                  }

66                  say "User $user has reputation $rep";

68                  if($name ne $user){
                            say "[-] This peer regexp failed with $name";
70                  } else {
                            $updatepeer->execute($user,$rep,$pagenum,$page);
72                          my $currentPeerId = $updatepeer->
                                fetchrow_arrayref;
                            $currentPeerId = $currentPeerId->[0];
74                          &parsePage($currentPeerId,$content);
                    }
76          } else {

78                  if($content=~m/has decided to make his\/her Feedback
                        comments private/){
                            say "[-] Bad user (private feedbacks)";
80                          my $sth = $dbh->prepare("UPDATE peers SET
                                baduser=true WHERE name=?");
```

```
                        $sth->execute($user);
                }

                say "[-] Regex didn't match. Resetting worker for peer $user";
                my $sth = $dbh->prepare("UPDATE peers SET worker=null WHERE name=?");
                $sth->execute($user);
                #die "[FATAL] Got captcha. Aborting for now\n";
                say "Sleeping for 10 seconds";
                sleep(10);
        }

}

sub parsePage($$){
        my($currentPeerId,$content) = @_;
        while($content=~/$regPeer/g){
                # $1 - value
                # $2 - username
                # $3 - reputation
                # $4 - timestamp
                # $5 - ebay_trid
                # $6 - item
                # $7 - currency
                # $8 - amount

                # name, reputation, capital, total pages, last page
                $updatepeer->execute($2,$3,0,0); # we haven't crawled this user yet
                my $buyerId = ($updatepeer->fetchrow_arrayref)->[0];
                #my $timestamp = $4;
                my $value;
                if(substr($1,0,1) eq "P") { $value = 1 }
                if(substr($1,0,3) eq "Neg") { $value = -1 }
                if(substr($1,0,3) eq "Neu") { $value = 0 }
```

```
                say "[+] $1 from $2(${buyerId})(rep: $3) -> ${currentPeerId}, $7$8 at $4, item $6, trid: $5";
                # from_id,to_id,amount,currency,ebay_trid,timestamp,item,value
                $updatefeedback->execute($buyerId,$currentPeerId,$8,$7,$5,$4,$6,$value);
        }
}
#parseUser('everydaysource',1);
my ($user,$lastpage);
do{
        undef $user;
        my $counter = $dbh->prepare("UPDATE peers SET worker=$$ WHERE id=(SELECT id FROM peers WHERE totalpages>lastpage and totalpages>0 and baduser is null and (worker is null or worker=$$) LIMIT 1) RETURNING name,lastpage");
        $counter->execute();

        ($user,$lastpage) = @{$counter->fetchrow_arrayref} if $counter->rows()>-1;
        if($user){
                say "[*] Continuing for $user at page ".($lastpage+1);
                parseUser($user,$lastpage+1);
        }
} while (defined $user);
say "[*] Done for now.";
```

Bibliography

[1] The Electronic Frontier Foundation. Ssl observatory. https://www.eff.org/observatory. Accessed: 01/04/2012.

[2] Dan Wendlandt, DG Andersen, and Adrian Perrig. Perspectives: Improving SSH-style host authentication with multi-path probing. *ATC*, 2008.

[3] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[4] International Telecommunication Union. Public-key and attribute certificate frameworks. http://www.itu.int/rec/T-REC-X.509. Accessed: 10/01/2013.

[5] VASCO Data Security International. Diginotar reports security incident. http://www.vasco.com/company/about_vasco/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx. Accessed: 01/09/2011.

[6] Comodo CA Ltd. Comodo report of incident - comodo detected and thwarted an intrusion on 26-mar-2011. https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html. Accessed: 23/03/2011.

[7] Trustwave Spider Labs. Clarifying the trustwave ca policy update. http://blog.spiderlabs.com/2012/02/clarifying-the-trustwave-ca-policy-update.html. Accessed: 05/02/2012.

[8] TURKTRUST Inc. Technical details. http://turktrust.com.tr/en/kamuoyu-aciklamasi-en.2.html. Accessed: 08/01/2013.

[9] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas. Internet X.509 Public Key Infrastructure: Certification Path Building. RFC 4158 (Informational), September 2005.

[10] P. Hoffman and J. Schlyter. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698 (Proposed Standard), August 2012.

[11] Moxie Marlinspike and Trevor Perrin. Trust Assertions for Certificate Keys. Internet-Draft draft-perrin-tls-tack-02, IETF Secretariat, January 2013.

[12] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk. Server-Based Certificate Validation Protocol (SCVP). RFC 5055 (Proposed Standard), December 2007.

[13] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems - P2PECON '05*, page 128, New York, New York, USA, 2005. ACM Press.

[14] Dusan Barok. Bitcoin: censorship-resistant currency and domain system for the people. Networked Media Piet Zwart Institute, 2011.

[15] Marat Vyshegorodtsev, Daisuke Miyamoto, and Yasushi Wakahara. BS-3-3 Governance-free secure domain naming system(BS-3. Management and Control Technologies for Innovative Networks). *Proceedings of the IEICE General Conference*, 2012(2):S–5"–"S–6, March 2012.

[16] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the twelfth international conference on World Wide Web - WWW '03*, page 640, New York, New York, USA, 2003. ACM Press.

[17] A.G.P. Rahbar and O. Yang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *Parallel and Distributed Systems, IEEE Transactions on*, 18(4):460 –473, april 2007.

[18] Daewon Kim, Byoungkoo Kim, Ikkyun Kim, Jeongnyeo Kim, and Hyunsook Cho. Endpoint mitigation of ddos attacks based on dynamic thresholding. In TatWing Chim and TszHon Yuen, editors, *Information and Communications Security*, volume 7618 of *Lecture Notes in Computer Science*, pages 381–391. Springer Berlin Heidelberg, 2012.

[19] Yuqing Mao, Haifeng Shen, and Chengzheng Sun. From credit and risk to trust: towards a credit flow based trust model for social networks. In *Proceedings of the 17th ACM international conference on Supporting group work - GROUP '12*, page 209, New York, New York, USA, 2012. ACM Press.

[20] Netcraft. The netcraft ssl server survey. https://ssl.netcraft.com/ssl-sample-report/. Accessed: 3/12/2011.

[21] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 42(1):1–31, December 2009.

[22] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, page 61, 2002.

[23] Novato Technology. Nortica ebay 500. http://www.nortica.com/UserArea/ebay500.asp. Accessed: 06/01/2013.

[24] VeriSign Inc. The domain name industry brief. http://www.verisigninc.com/assets/domain-name-brief-march2012.pdf. Accessed: 01/04/2012.

[25] V. Ramasubramanian and E.G. Sirer. Beehive: O (1) lookup performance for power-law query distributions in peer-to-peer overlays. In *Proc. First Symp. on Networked Systems Design and Implementation (NSDI'04), San Francisco, California*, pages 99–112, 2004.

[26] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing Social Networks. *2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, May 2009.

[27] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4):661–703, November 2009.

[28] Bin Yu, M.P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. *IEEE First Symposium onMulti-Agent Security and Survivability, 2004*, pages 1–10, 2004.

[29] Anonymous. The collateral damage of internet censorship by DNS injection. *ACM SIGCOMM Computer Communication Review*, 42(3):21, June 2012.

[30] Dusko Pavlovic. Dynamics, robustness and fragility of trust. *Formal Aspects in Security and Trust*, pages 97–113, 2009.

[31] Marco Remondino and Guido Boella. How users'participation affects reputation management systems: The case of P2P networks. *Simulation Modelling Practice and Theory*, 18(10):1493–1505, November 2010.

[32] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[33] John Brainard, Ari Juels, and RL Rivest. Fourth-factor authentication: somebody you know. *Proceedings of the 13th ...*, pages 168–178, 2006.

[34] Radia Perlman and Charlie Kaufman. User-centric PKI. In *Proceedings of the 7th symposium on Identity and trust on the Internet - IDtrust '08*, page 59, New York, New York, USA, 2008. ACM Press.

[35] Chithra Selvaraj and Sheila Anand. A survey on Security Issues of Reputation Management Systems for Peer-to-Peer Networks. *Computer Science Review*, 6(4):145–160, July 2012.

[36] Agnieszka Danek, Joana Urbano, A Rocha, and E Oliveira. Engaging the dynamics of trust in computational trust and reputation systems. *Agent and Multi-Agent Systems: ...*, 2010.

[37] Ari Juels and John Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proceedings of NDSS '99 (Networks and Distributed Security Systems)*, pages 151–165, 1999.

[38] Sini Ruohomaa, Lea Kutvonen, and Eleni Koutrouli. Reputation Management Survey. *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 103–111, 2007.

[39] Junsheng Chang, Huaimin Wang, and Gang Yin. A time-frame based trust model for p2p systems. In MinSurp Rhee and Byoungcheon Lee, editors, *Information Security and Cryptology – ICISC 2006*, volume 4296 of *Lecture Notes in Computer Science*, pages 155–165. Springer Berlin Heidelberg, 2006.

[40] Félix Gómez Mármol and Gregorio Martínez Pérez. Providing trust in wireless sensor networks using a bio-inspired technique. *Telecommunication Systems*, 46:163–180, 2011.