The Theory of Creating Trust with a Set of Mistrust-Parties

and its Exemplary Application for the S-Network

Johannes Viehmann
Fraunhofer Institut FOKUS (MOTION)
Kaiserin-Augusta-Allee 31
D-10589 Berlin, Germany
Johannes.Viehmann@Fokus.Fraunhofer.de

***Abstract*— This paper presents the idea for achieving trustworthiness by splitting responsibilities between different parties mutually mistrusting one another. These parties are called mistrust-parties because some kind of mistrust between these parties is actively created to prevent potentially manipulative cooperation.**

The birth of the S-Network, a universally applicable trustworthy repository, should enable users to make and access reliable publications and secure deposits. The S-Network combines secure long term data storage and preservation in a computer network with non-repudiation and legal validity. This paper describes how one can apply the concept of creating trust with the help of mistrust-parties for the S-Network so that the S-Network itself would be highly trustworthy.

Besides being a potential application, the S-Network could also be used as a tool for vital parts in the measures for creating trust with a set of mistrust-parties described in this paper.

Trust; mistrust; game theory; non-repudiation; trustworthy repository; S-Network

I. Introduction

Trustworthy and secure services in computer networks are required for many applications in varying market sectors, including eCommerce, eGovernment, and eHealth.

In this paper, several known concepts for achieving trustworthiness – including the usage of “trusted third parties” and the “web of trust” – are briefly discussed.

Attempting to overcome some of their weaknesses, a new concept for creating trust with a set of *mistrust-parties* is introduced and analyzed using game theory.

To show how the concept could be applied, a large scale trustworthy repository called the *S-Network* will be presented here. The *S-Network* must provide guarantees for the long term preservation and for the permanent secure non-repudiation accessibility of its content. The *S-Network* requires strong authentication and offers confidentiality. Depending on the chosen access modalities, data stored in the *S-Network* is either called a *reliable publication* or a *secure deposit*: The audience (i.e., users having read rights) and the validity period of *secure deposits* may be expanded, while *reliable publications* my never be changed at all.

Requiring all users to agree on a user contract, the *S-Network* guarantees legal validity for its publications and deposits, including verifiable metadata values (e. g. who published what and when) with standardized legal implications for all its participants.

The *S-Network* is intended to become a universal platform for applications that have most stringent requirements, e.g. fair contract signing. Indeed, it must be resistant to both manipulation attempts and censorship. Especially, no single party, institution or state should have control over the *S-Network.*

In the trustworthiness creation process introduced in this paper, it may be necessary to do non-repudiation registrations. For these registrations, the *S-Network* will be beneficial, given that it was designed to be a non-repudiation information system. Thus, the *S-Network* is both a utility for, and an application of the concept that will be presented here.

II. Problems and the State of the Art

As described in [10] trust is essential for many information systems including the *S-Network.* In computer science, when trust is required, the concept of a “trusted third party” [22] often comes to mind. The “trusted third party” is usually assumed to be perfectly secure, always behaving correctly, and party-neutral (i.e., unbiased). If an absolutely fair and flawless “trusted third party” were to exist, critical tasks like *authentication* or *key distribution* in large scale computer networks could easily be solved with the help of a “trusted third party.” However, one concern would still remain, i.e., the residual risk associated with the potential elimination of the “trusted third party.”

To realize such a perfect “trusted third party” and to ensure its absolutely fair and flawless behavior throughout its lifetime is extremely difficult, if not impossible. Perhaps, for a relatively limited purpose, trustworthiness of a single party could be achieved. Hence, that is why in [27], a “trust

Presented at PST 2012 in Paris, published in the conference proceedings (http://www.ieee.org), © Copyright IEEE

classification" is suggested to model trustworthiness in specific contexts.

However, in general, it is unrealistic to assume that one could find (or create) and control a perfect "trusted third party." For example, the SSL (Secure Sockets Layer) protocol and its successor, the TLS (Transport Layer Security) protocol, are widely applied depending upon "trusted third parties." They are used for all kinds of security critical Internet applications (e.g., online-payments). Modern web browsers, today, do not only rely on a single "trusted third party." By default, they process and store numerous digital certificates arising from multiple "trusted third parties" called CAs (certification authorities). If just a single "trusted third party" is corrupt, then the concept is broken. The recent incidents with *DigitalNotar* [23] have shown the high vulnerability and the potentially disastrous consequences [18] inherent in this approach.

For the *S-Network*, it is not a good idea to rely on a "trusted third party." A platform that would allow its users to make *reliable publications* and *secure deposits* is by far too universally applicable. Anybody could have massive interests to manipulate and/or censor such a system, including especially governments and other powerful institutions or moral authorities who might want to control the *S-Network*.

A decentralized alternative for creating trustworthiness is known as the "web of trust" [28]. However, with this approach, it is hard to achieve legal validity. Its handling is too difficult [26]. Besides, to presuppose that trust is in general transitive is flawed [5].

Another idea for creating trust is based on the assumption that the majority will always behave correctly. Majority based concepts can be seen as a special kind of threshold agreement protocols where it can for sure be detected if less than a certain threshold number of participants behave incorrect as long as at least threshold participants behave correct. That is known as Byzantine fault tolerance (referring to the Byzantine Generals' Problem [13]) and there are lots of publications describing practical applications in computer science. However, many of these like PBFT [3] and Q/U [1] require additional authentication, verifiable signatures and reliable public keys – they cannot create trust without relying on another trust concept like a "trusted third party" or a "web of trust" for distributing the public keys correctly.

An example for a fault tolerant, majority-based concept to create trust that does not have such additional dependencies is *Bitcoin* [14]. More precisely, the assumption made for *Bitcoin* is that the majority of the calculation power is used honestly and correctly, at any point in time.

But, the assumption about the correctness of the majority may be wrong. For example, a majority could collectively cheat the system, for their own personal gain, at the cost of a minority. Coalitions having a majority can do whatever they want, including all kinds of manipulations. Processes of confederation and collusion are major problems for all democratic trustworthiness creation concepts.

With concepts based on the calculation power like *Bitcoin*, there are additional problems. The calculation capacities are not equally distributed. In order to prevent manipulations, the calculation power used for *Bitcoin* must be significantly bigger than the world's most powerful super-computer, which could be controlled by a single entity. That might be very expensive and a sheer waste. Furthermore, with the help of malicious software like the *Trojan.Badminer* reported in [9], it is possible to utilize calculation power of strangers for manipulations, as well.

A. *Work related to the S-Network*

The S-Network is designed as an open archival information system according to the OAIS standard [4]. It must fulfill the criteria for trustworthy repositories defined in [19]. [16] gives an overview about digital long-term preservation concepts and solutions like [17]. Most solutions are designed for limited purposes, e.g. for digital libraries.

The "OceanStore" [12] is designed as a highly distributed global scale persistent storage system like the S-Network. But it does not support non-repudiation and it does not offer legal validity. Furthermore, it depends on a "responsible party".

There have been several proposals for creating somehow reliable permanent digital publication systems. "The Eternity Service" [2] or "Publica" [25] try to offer anonymity, but therefore they cannot offer confidentiality, non-repudiation and legal validity like the *S-Network*. In contrast, the *S-Network* cannot support anonymous publications. Otherwise it would not be possible to deal with illegal content as discussed in chapter V.

III. Creating Trustworthiness with a Set of Mistrust-Parties

The trustworthiness creation concept introduced in this chapter is also majority based. It requires a fixed set of parties that all share common properties and in addition, a legal basis. Each party may consist of multiple entities, each having equal rights to act in the name of the entire party. No individual may be a member of more than one single party. The minimum number of parties is three.

The members of the parties have a legally binding obligation to behave in a specified way. However, the individuals are not trusted. It is assumed that each individual may, but not necessarily will, follow the rules.

Responsibilities are split across these parties. If and only if the majority of the involved parties agree on something, the outcome will be regarded as valid. The exact number of parties that are at the very least required to agree is called the *threshold* Ψ.

The number of parties involved in the split of some responsibility must be $2*\Psi-1$. If members belonging to less than *threshold* Ψ parties behave incorrectly, then this can be detected because there are still at least Ψ parties whose members are all behaving correctly. Those behaving incorrectly will have to take responsibility for their actions.

A. *Threats*

If the members of Ψ or more parties behave incorrectly, there might eventually be no way to detect that. Especially, if a large enough coalition involving members of at least Ψ different parties arises, it would be possible to break the legally binding rules in a coordinated way. Such cooperation could result in successful manipulations. Even worse, members of those parties behaving correctly would probably be charged with cheating.

This concept is threatened by such conspiracy coalitions – no different from any other majority based trust concept. Preventing manipulative cooperation is absolutely required in order to effectively improve the overall trustworthiness by splitting responsibilities over a fixed set of parties in comparison to concepts using a single "trusted third party."

Of course, there are other threats to consider: Parties could be blackmailed. Vulnerabilities of technical systems a certain party is responsible for could be used by attackers to make that system behave incorrectly. The party might not even notice.

However, this paper focuses on the threat of conspiracy coalitions. For the concept presented here, it is essential to make collusion and any process of confederation among the parties that could result in successful manipulation as risky and as expensive as possible. Therefore, the following measures are provided:

B. Separation

Each party is associated with one or more states. No single state may have more than one party associated with it. To become a member of such a party, it is necessary to make an official registration under the jurisdiction of one of the states the party is associated with. There may be additional requirements like having the citizenship of the state in which the registration takes place.

The *trust-distance* of two parties is defined as a mathematical function that models the gap between the two parties by taking geographical, political, jurisdictional and cultural facts about the state(s) each party is associated with as input. For example, a criterion for the geographical distance could be whether the states are bordering or not. A criterion for political and jurisdictional distance could be whether the states are completely sovereign states, sovereign states within the same union, or just federal states within the same sovereign state. For instance, a cultural criterion could be the degree of relatedness between official languages among the states.

The parties should be constituted so that the sum of the *trust-distances* of all possible pairs of different parties is maximized for the total number of parties desired.

Even enemies having a high *trust-distance* between each other could take advantage of cooperating with one another – and history has shown that they do. One possible reason for such cooperation even between ideological and political opponents is that they have both the same third party as a common enemy. For example, Sir John Colville reports the following conversation with Winton Churchill during the Second World War about the question whether to ally with the Soviet Union or not:

"I said that for him, the arch anti-Communist, this was bowing down in the House of Rimmon. He replied that he had only one single purpose – the destruction of Hitler – and his life was much simpler thereby. If Hitler invaded Hell he would at least make a favourable reference to the Devil!" ([6], page 404).

High *trust-distances* might imply some "natural" reluctance against potentially manipulative cooperation between different parties. If the potential benefit of manipulative cooperation is high enough, such barriers seem to be rather low – though still better than nothing. Probably most important, the maximization of *trust-distances* in the process of finding the optimal set of parties should ensure that any two parties are under distinct and independent jurisdictions. This might help to make the following legal measures effective even against those who have great much influence on some single jurisdiction.

C. Prohibition, Prosecution, Penalties

Manipulative cooperation between different parties must be forbidden by law. Attempts to create manipulative collaboration must also be made illegal.

Should prohibited behaviors be observed, these must be handled in the court system, and the delinquent parties must be penalized upon conviction.

The possible penalties must be well-known and they must comply with the principle of proportionality (according to the potential benefit that could be taken from successful manipulation).

D. Monitoring

Monitoring can be a helpful measure to uncover illegal actions. Ongoing manipulative cooperation might cause inconsistent intermediate results or it might produce suspect data traffic, for example.

However, once there is a coalition, there may be a coordinated attack involving more than Ψ parties simultaneously, which might quickly produce irreversible results and which could probably never be detected even with most careful monitoring.

Unfortunately, it is very difficult to detect collusion before actual manipulations begin. If people really want to communicate and negotiate in secret, they will manage to do it. For example, the *S-Network* is going to provide everything that is required for secure secret communication between any two participants.

To be able to detect collusion and any processes of confederation at the very beginning by external observation would require extreme measures of surveillance for all participants, which is neither practicable nor desirable.

Only the members of the parties themselves can reveal the illegal cooperation offers they get from members of other parties. These insiders can stop and report any illegal process of confederation at the beginning. But why should they?

With the help of game theory, it is possible to analyze the opportunities in the case of an incoming request for manipulative cooperation. Such a situation can be modeled as a majority game [15].

The simplest possible process of confederation involves only two parties Alice and Bob. There may be another party Charlie, but Alice and Bob already have the majority required to conduct manipulations, if just these two cooperate with one another. The situation to be analyzed here in the first place begins right after Alice has made the initial step – she has already proposed some kind of illegal cooperation to Bob.

Alice and Bob both have the choice to report the process of confederation, take an active part in the illegal manipulation process, or do nothing.

Defining integer utility values for the different options of the active players Alice and Bob makes it possible to compare the different possibilities with each other. The utility value −100 indicates a maximal penalty for being caught and convicted for illegal behavior. The utility value 0 indicates no significant personal advantage or disadvantage, while the utility value 100 indicates the maximum possible benefit.

For this game, it is assumed that any report will lead to a maximal penalty for all misbehaving entities. In reality, sometimes there might not be enough evidence for a conviction. Similarly, it is assumed that attempting to manipulate will always be successful if that attempt is made by a large enough coalition – having two participating parties in this simplest scenario. In reality, attempts at manipulation might fail due to technical problems or they could be detected by monitoring measures.

Table I shows the utility values for Alice and Bob for all possible combinations of their options. The utility value for Alice is noted first, the utility value for Bob is noted second. Such a notation is used for example in [21].

TABLE I. UTILITY VALUES FOR ALICE, BOB WITH MEASURES 3.1–3.3

		Bob		
		Report	***Do nothing***	***Manipulate***
Alice	Report	−100, 0	−100, 0	−100, −100
	Do nothing	−100, 0	0, 0	X∗100, X∗100
	Manipulate	−100, 0	X∗100, max(X∗100, 0)	100, 100

In some cases it is possible that the resulting utility value may depend on the behavior of the third party Charlie: If cooperation between Alice and Bob does not work, then it might probably be possible for one of them to cooperate with Charlie. The Variable X is set to −1 if Charlie reports the offer made in that situation, it is set to 0 if he does nothing or it is set to 1 if Charlie decides to manipulate.

It is most challenging to assume that Alice and Bob both get the maximum utility value 100 if the manipulation takes place – regardless, whether Alice and Bob are actually actively taking part in the manipulation process or not:

For Bob, this makes the option to do nothing dominant over the report option. Doing nothing, Bob can win the maximal utility value 100, if Alice and Charlie do the manipulation. If the manipulation does not take place, it cannot affect Bob, even if Charlie reports Alice's request for cooperation because Bob did nothing wrong. So Bob's worst possible utility value for doing nothing is 0. If Bob reports, he can only get a neutral utility value of 0. An additional measure, the revelation duty makes doing nothing less attractive.

E. Revelation

Participants receiving a request for cooperation violating the rules are forced by law to report that following a strict protocol. This revelation protocol consists of two steps:

First, the incident has to be registered in a non-repudiation repository, which makes sure that the incident will be revealed and published in the future. Once registered in that way, the manipulation attempt will be detected for sure.

Second, evidence must be collected to make sure that the misbehaving individual can be convicted and penalized, too. This step might require acting as if there were a real interest in manipulating. External observers, for example the police, can help to collect evidence.

TABLE II. UTILITY VALUES FOR (ALICE, BOB) WITH MEASURES 3.1–3.4

		Bob		
		Report	***Do nothing***	***Manipulate***
Alice	Report	−100, 0	−100, −100	−100, −100
	Do nothing	−100, 0	0, 0	X∗100, X∗100
	Manipulate	−100, 0	X∗100, X∗100	100, 100

Given the revelation duty, it would be less attractive for Bob to do nothing than to get directly involved in the manipulation (table II).

For Bob there is a very low risk to cooperate with Alice and to manipulate since it is rather unlikely that Alice is going to report: Alice has already committed a crime by making the offer for illegal cooperation to Bob; therefore Alice would definitely harm herself if she decided to report now.

The measures presented so far are obviously not enough to motivate people to reveal any process of confederation at the very beginning.

F. Actively Creating Mistrust: Testing with Temptation

The idea is to create a "healthy" mistrust against any suggestion or request for manipulative cooperation between the parties.

The correct revelation behavior should be tested with fake cooperation proposals for some kind of manipulation. Testing may be initiated by any participant by choice. There may also be procedures for choosing and probably even forcing participants to initiate tests, for example, with a random distribution and with a certain frequency.

A fake request for testing purposes must look perfectly real to the test subject, but it must be made in a way that it cannot lead to any successful manipulations – even if the person being tested shows interest in the proposal instead of revealing it correctly.

Therefore, fake proposals have to be registered for investigation in a standardized way before they are made. The person being tested may not get a chance to detect such a registration before the test is finished. After the test is completed, the registration must automatically be published and kept accessible in a non-repudiation fashion.

If Alice initiates such a test, the resulting situation for her is quite different from the previously discussed game situation that results from making a real illegal cooperation proposal. Testing is in fact another game for Alice, in which she does not have to make any further decisions – she has no more options.

From Bob's point of view, a test looks just like a real illegal cooperation request. Eventually, Alice does not actually make a real proposal for illegal cooperation. She could just be testing Bob's revelation behavior. Bob might not be able to distinguish between real and fake proposals. For him, there are no two different games. He faces a single Bayesian game [8] with at least two potential types of players, i.e., Alice$_{\mathrm{T}}$ is testing him, or Alice$_{\mathrm{I}}$ makes a real illegal request. The decision whether Bob plays with Alice$_{\mathrm{T}}$ or Alice$_{\mathrm{I}}$ is made by "Nature" at the very beginning of the game.

For Alice$_{\mathrm{T}}$, the utility value V is independent from Bob's decision and it will never be negative. In contrast to Alice$_{\mathrm{I}}$, Alice$_{\mathrm{T}}$ can never harm herself. Alice$_{\mathrm{T}}$ does nothing wrong, she is just testing Bob's behavior. Bob cannot know Alice$_{\mathrm{T}}$'s utility value V because V depends not only on her potential personal interest in testing him, but also on externalities, which are not accessible for Bob: Alice$_{\mathrm{T}}$ may be chosen by some procedure, which asks or even forces her by law to initiate the test and she might also get monetary compensation for initiating the test. Bob actually faces a continuum of player types testing him, having all possible utility values $V \mid V \geq 0$.

TABLE III. UTILITY VALUES FOR (ALICE$_{\mathrm{T/I}}$, BOB) WITH ALL MEASURES

		Bob		
		Report	***Do nothing***	***Manipulate***
Alice$_T$	Test	V, 0	V, –100	V, –100
Alice$_I$	Report	–100, 0	–100, –100	–100, –100
	Do nothing	–100, 0	0, 0	X*100, X*100
	Manipulate	–100, 0	X*100, X*100	100, 100

No matter what Alice$_{\mathrm{I}}$ does, she can never prove that she is really interested in doing manipulating. Any affirmation could be made by Alice$_{\mathrm{T}}$ acting as a part of an already registered test. For Bob, any request could always be a trap.

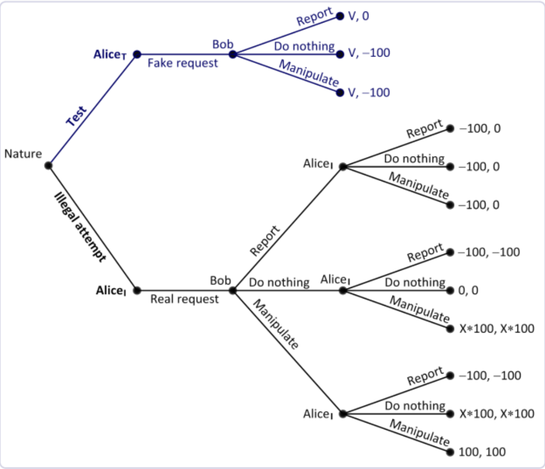

Figure 1. Extensive-form game tree

However, Bob can do some statistical analysis to estimate his risk of playing with Alice$_{\mathrm{T}}$: Let R be the number of test requests that have been revealed correctly. Let M be the number of real illegal manipulation requests that have been revealed correctly. These values are accessible for Bob because all reports have to be published. Bob can calculate the probability P_T that a reported request for cooperation is just a test:

$$P_T = \frac{R}{R+M} \tag{1}$$

P_T is a good approximation for the probability that Bob is playing with Alice$_{\mathrm{T}}$ though the unrevealed requests are ignored: The ratio between unrevealed test requests and the sum of unrevealed tests requests and unrevealed real illegal manipulation requests should be P_T as well, as long as those not revealing correctly cannot get better knowledge to answer the question whether they are tested than those who do reveal requests correctly.

Assuming that the likelihood that Alice$_{\mathrm{I}}$ behaves irrationally and chooses to either report or do nothing is close to zero, Bob can calculate an expected utility value $E_{\mathrm{Bob(Manipulate)}}$ for doing the manipulation weighted with the probability P_T:

$$E_{\mathrm{Bob(Manipulate)}} = P_T * (-100) + (1 - P_T) * 100 \tag{2}$$

Let N be the total number of tests. The probability P_R that a request will be revealed can be calculated by dividing the number of revealed tests R with N:

$$P_R = \frac{R}{N} \tag{3}$$

Alice$_{\mathrm{I}}$ can calculate her expected utility value $E_{\mathrm{Alice_I(Manipulate)}}$ to evaluate her chances if she makes a real illegal manipulation request and tries to manipulate:

$$E_{\mathrm{Alice_I(Manipulate)}} = P_R * (-100) + (1 - P_R) * 100 \tag{4}$$

1) *Multiple players.*

In practice, the threshold Ψ should be larger than two. The need to convince additional parties for manipulative cooperation makes it even more dangerous to try it.

With Ψ > 2, if Bob wants to cooperate in illegal activities, he does not only have to worry about the question whether he is actually playing with Alice$_{\mathrm{I}}$, but he will also need to be concerned about the decisions of the other Ψ–2 parties required in order to succeed in a manipulation. Each of them could reveal the manipulation request.

Those who register a request to serve as non-reputable evidence for later revelation should act for a while as if they were really interested in manipulating. That way, other parties can be asked to join the ongoing confederation process and their revelation behavior can be tested, too. Any request should be revealed Ψ–1 times.

Let i be a non-negative integer smaller than Ψ. Let R_i be the number of tests revealed by a player asked for illegal cooperation after i players have been asked before.

Let j be a non-negative integer smaller than Ψ. The player who is asked after j other players have been asked to join some illegal manipulation can calculate the probability P_j that the request is a test or will be revealed:

$$P_j = 1 - (1 - P_T) * \left(\prod_{i=0}^{\Psi-1} \left(1 - \frac{R_i}{N}\right) \right) \div \left(1 - \frac{R_j}{N}\right) \quad (5)$$

Note that the product in (5) has the probability that the request is not a test ($1-P_T$) and the probabilities that each of the other players will not reveal as factors. The product calculates the probability that requests to illegally cooperate are not revealed and could lead to a successful manipulation, which is $1-P_j$. It is possible to calculate this probability value in that way since the corresponding events are statistically independent and a successful manipulation requires all these events to occur.

Hence, the effective utility value can be calculated as in (2).

The concept of testing with temptation jointly with the revelation duty actively creates mistrust between the parties against any illegal cooperation proposals. This mistrust helps to prevent confederation processes at the very beginning. The more mistrust exists between the parties, the greater the trustworthiness in the entire system will be. Since actively creating mistrust is most characteristic for the parties in this concept, it is quite accurate to call them *mistrust-parties*.

2) Using the S-Network for the Revelation and Testing with Temptation Concepts.

The *S-Network* should become a highly trustworthy platform for *reliable publications* and *secure deposits*. A *reliable publication* may never be altered or deleted till the end of its validity period by anyone, including the publisher himself. It is guaranteed to be accessible throughout its validity period, but neither before nor after. For any of its strictly non-repudiation publications the *S-Network* has to create, validate and maintain metadata values indicating, for example, who stored what, when.

If the *S-Network* guarantees these properties with legal validity, then it is a good platform for the registration steps in the revelation protocol and in the testing process. The registration can be made with a *reliable publication* in the S-Network, which has an infinite validity period that starts in the near future. Until the validity period starts, nobody can access the *reliable publication*. During that period of time, it is possible to play act and to collect evidences. At the start of the validity period, the registration is guaranteed to become and to stay accessible for an unlimited period of time by the *S-Network*. The design of the *S-Network* also contains a concept of reliable links, which is appropriate to make sure that such a registration can and will be easily found as soon as it becomes accessible.

That way, the *S-Network* could be very useful as a technical platform for these procedures of the concept to create trustworthiness with the help of *mistrust-parties*. However, this only makes sense if the *S-Network* itself is highly trustworthy and secure. To make sure that the S-Network becomes as trustworthy as possible, the concept of splitting responsibilities over a set of *mistrust-parties* should be used:

IV. Creating Trust in the S-Network with Mistrust-Parties

A. Bit Preservation

The *S-Network* should offer long term bit sequence preservation for its content. An obvious strategy for preserving bit sequences is to create security copies and to distribute these copies. For example, this idea is used for LOCKSS [17] and it will be used for the *S-Network* as well.

However, if legal validity is to be achieved, instead of just making *several* copies and distributing them *somehow*, there must be sound regulations defining the difference between legally effective and legally void data.

Having a set of *mistrust-parties* with a *threshold* Ψ, it is possible to define reasonable rules for the number and the distribution of security copies:

Publications and deposits in the *S-Network* are exactly legally valid if they are confirmed by at least Ψ *mistrust-parties*. Ψ should be the majority of the potential confirmations. Even if up to Ψ−1 security copies are corrupted or destroyed, there should still be at least Ψ correct copies, which is the minimum number to achieve validity. Therefore, for any reliable publication or secure deposit in the *S-Network*, 2∗Ψ−1 security copies must be distributed over systems in 2∗Ψ−1 different *mistrust-parties*.

Reparation and reconstruction capability is crucial for long term preservation as there is no perfect long term storage medium known. Security copies have to be compared regularly with each other to detect faults. Dissenting versions must be replaced with the majority version confirmed by at least Ψ different *mistrust-parties*. These procedures must be Byzantine fault tolerant. The S-Network uses a Quorum based protocol.

As long as less than Ψ different *mistrust-parties* are incorrect, the data is still legally valid and deviating versions of security copies can certainly be repaired. A coalition involving Ψ or more *mistrust-parties* can perform all kinds of manipulations.

B. Metadata Generation and Validation

The 2∗Ψ−1 *mistrust-parties* that store a security copy have to generate and preserve some metadata values and they have to keep them accessible together with the security copy. An example for such a metadata value is the point of time when the publication or deposit in the *S-Network* takes place.

For any publication or deposit in the *S-Network*, these metadata values will be generated independently by the 2∗Ψ−1 systems in different mistrust-parties which have to store a security copy. The preservation of the metadata values works like any other bit preservation in the *S-Network* – with the exception that metadata values are already distributed when they are generated.

For legal validity, at least Ψ identical metadata values have to be received form at least Ψ different *mistrust-parties*. As long as only up to Ψ−1 parties behave incorrect, there are still valid and correct metadata values. A coalition involving Ψ or more *mistrust-parties* can once again perform all kinds of manipulations.

C. Access Control

Read rights for a *reliable publication* or *secure deposit* within the *S-Network* may be restricted. Entire *mistrust-parties* storing security copies possibly do not have the right to read the content, so it is not sufficient to distribute plaintext security copies.

Provable perfectly secure secret sharing technologies as shown in [20] can be used to split some data D into a set of $n \in \mathbb{N}$ shares with the property that at least *threshold* $t \in \mathbb{N}$ of these shares are required in order to be able to reconstruct D from that subset. Any subset with less than *threshold* t pieces does not reveal any information about the content of D.

In the *S-Network*, for *reliable publications* or *secure deposits* with restricted read rights, instead of plaintext security copies, n shares of a secret sharing split are distributed over the set of mistrust parties so that at least *threshold* Ψ *mistrust-parties* have to cooperate to be able to get *threshold* t shares and to reconstruct the plaintext.

Note that there are now two different *thresholds*: Ψ is the number of *mistrust-parties* that have to cooperate to be able to reconstruct the secret, whereas t is the number of shares that is required to reconstruct the secret. Obviously, $t \geq \Psi$ must always be satisfied.

Each party is obligated to grant read access for their share only to those who have official read rights. No party is allowed to reconstruct a secret from shares without having official read rights. As long as there are only up to Ψ−1 parties behaving incorrectly, access controls are operating normally, and secrecy remains perfect. To successfully manipulate would require at least Ψ misbehaving *mistrust-parties*.

1) Combining Access Control with Bit Preservation.

For long term preservation, fault detection and reconstruction needs to be possible for shares, too. However, the secrecy should not be affected by that procedure. A possible solution is to make 2*Ψ−1 security copies for each share and to distribute them to systems in 2*Ψ−1 different *mistrust-parties*. The distribution must be made in such a manner that at least Ψ *mistrust-parties* would still have to cooperate to get the t shares that are required to reconstruct the secret.

Each copy of some share S may be send to those *mistrust-parties* that have to store a copy of the same share S, obviously without affecting the secrecy because these parties should already have stored exactly a copy of share S. This makes fault detection and reparation possible for each individual share just as if that share was some plaintext. There is no need to reconstruct the secret and the other shares are not involved in the bit sequence preservation process for share S at all. If a security copy of share S deviates from a majority of at least Ψ copies of S received from at least Ψ different *mistrust-parties*, then this copy is invalid and it can be repaired.

This solution is very generic. It works with any secret sharing technology, including the simplest, but therefore least computationally intensive secret sharing technologies, which only supports a *threshold* t equal to the total number n of shares. However, it requires more than 2*Ψ−1 different *mistrust-parties*. Ψ can no longer be the majority of the total number of mistrust parties. With this solution, the *threshold* Ψ is only the majority of all copies for some share S.

If the total number of *mistrust-parties* is at least Ψ*(2*Ψ−1), the number n of shares and the secret sharing *threshold* t may both be equal to Ψ because no *mistrust-party* will need to store more than a single copy of a single share.

If the total number of *mistrust-parties* is smaller than Ψ*(2*Ψ−1), more shares are required, threshold t must be bigger than threshold Ψ and the shares have to be distributed with caution to make sure that no less than Ψ mistrust parties are able to reconstruct the secret. The optimization problem of finding the best possible distribution is somehow similar to the set coverage problem, which is NP-hard [11]. While finding perfect solutions is probably very difficult, it is definitely possible to find correct and good solutions with greedy algorithms developed for the *S-Network*.

2) Taking Trust-Distances into Account.

Combining two different responsibilities (bit preservation, access control) it makes sense to remind ourselves that *trust-distances* between different pairs of *mistrust-parties* may vary. In particular, there may be *unions* of *mistrust-parties* who have low *trust-distances*. Confederations among these parties in such a *union* may be more likely than coalitions with other more distanced parties.

Copies and shares should be distributed so that no single such *union* can manipulate a bit sequence and that no single such *union* can reconstruct a secret without involving other mistrust parties not belonging to the same *union*.

The distribution shown in Figure 2. makes certain that the above property is satisfied for *union* Beta, but not for *union* Alpha. Having a total number of 15 *mistrust-parties* and using n = t = 3 shares, there is actually no distribution ensuring this property for *union* Alpha, *union* Alpha is just too big. Either *union* Alpha will have the majority of Ψ or more copies of at least one share or it will control at least one copy of each share.

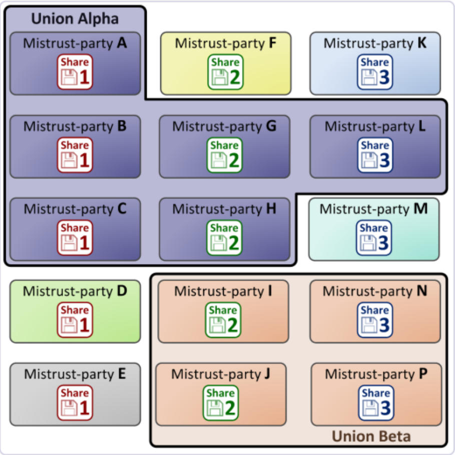

Figure 2. Distribution of shares with Ψ=t=3 and 15 different *mistrust-parties*

For 15 *mistrust-parties* with the threshold Ψ=3, only 20% of the *mistrust-parties* are required to manipulate or access the data without having the required read rights.

Figure 3. shows an alternative configuration and distribution with fewer *mistrust-parties*. The 10 *mistrust-parties* having the least trust distance with each other were pairwise merged. Having only 10 *mistrust-parties* and a threshold Ψ=3, at least 30% of all *mistrust-parties* have to agree to make something valid.

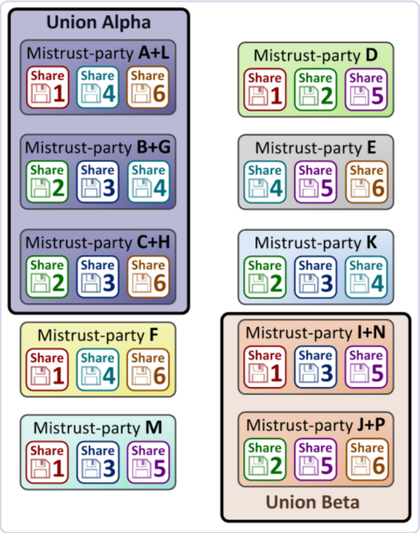

Figure 3. Distribution of shares with Ψ=3, t=6 and 10 different *mistrust-parties*

Using *threshold* t = 6 shares requires two times more memory. But having 33% less *mistrust-parties* may reduce administrative efforts in return.

Notice that *union* Alpha still consists of 3 different parties, which is equal to the threshold Ψ=3. However, the way the shares are distributed, *union* Alpha does not have all the shares because it does not have share 5. And for any share S, *union* Alpha is not responsible for more than two copies of the same share S. Hence, *union* Alpha can neither access the content nor delete or manipulate one of the shares.

Union Beta contains in the alternative configuration less than *threshold* Ψ parties, so union Beta needs the cooperation of other *mistrust-parties* not belonging to *union* Beta to cheat successfully, regardless.

D. Secure Communication

"Complexity is the worst enemy of security" ([7], page 17). Security critical systems should be as simple as possible. Limiting the number of features and using the same solid base technologies and concepts wherever possible reduces the number of potential vulnerabilities.

The *S-Network* distributes data over systems in different *mistrust-parties*. The users need to be authenticated for both access control and metadata validation. The *S-Network* therefore requires an efficient, reliable and secure communication between the systems and between the systems and the users, which ensures secrecy, authenticity, integrity and the correct order of exchanged messages.

Secure communication is possible with a concept using secret sharing and *mistrust-parties*, too. No "trusted party" is required and there are no dependencies on assumptions of complexity theory: Independent from further technical development, this concept can be guaranteed to be as secure as the *S-Network* in general is – successful manipulations must involve at least Ψ incorrect *mistrust-parties*. See [24] for details.

V. Required Negotiation – a Challenge

The concept of creating trust by splitting responsibilities over a set of *mistrust-parties* is threatened by conspiratorial cooperation among the parties. It is essential to put great much effort on preventing negotiation among the parties, which could lead to successful manipulations.

However, there may be situations requiring some kind of correction, which might be impossible without negotiation resulting in certain special forms of cooperation between the parties.

It can be challenging to define some legal possibilities for negotiation and cooperation without opening up possibilities for other illegal manipulations using the legal negotiation as a starting point.

A. Non-repudiation and Illegal Content

As a drastic example, consider the case that some user of the *S-Network* would make a *reliable publication* with an infinite validity period containing child pornography. In general, the *S-Network* is a non-repudiation platform and it guarantees to keep any *reliable publication* accessible for the entire audience throughout the whole validity period – which would mean forever in this example. In the case of child pornography, such exposure would be a second crime against the victims.

Of course, the person making a *reliable publication* in the *S-Network* can be identified for sure. Publishing child pornography in the *S-Network* would therefore be a kind of self-destructive behavior. But the knowledge that the consequence will most likely be a prison sentence does not mean that no one will try. Particularly, for someone who believes that he has nothing to lose, it could appear to be highly attractive to raise some kind of monument with a publication that has to be kept open accessible forever.

Clearly enough there must be some way to prevent such misuse of the *S-Network*.

Only manipulative actions involving at least Ψ different *mistrust-parties* can affect the permanent accessibility of a publication or deposit within the *S-Network*. The concept of creating trustworthiness with *mistrust-parties* was developed to prevent precisely such manipulations. Using this concept ensures the *S-Network* is robust against censorship, for example. But this concept also makes it difficult to react on illegal content that cannot be tolerated.

To make sure that content like child pornography does not have to remain openly accessible in the *S-Network* forever, a certain form of negotiation and cooperation between the *mistrust-parties* is required and must therefore be allowed.

First of all, there must be a set of regulations defining exactly which content is illegal for *reliable publications* and *secure deposits* in the *S-Network*. These rules must be applicable for the entire *S-Network*, for all *mistrust-parties*.

If some content is seriously suspected to be illegal, a jury involving representatives from all *mistrust-parties* must investigate the case and arrive at a verdict deciding whether the content is legal or illegal. The necessary negotiation must be documented in a standardized way, for example with the help of *reliable publications* in the *S-Network*.

Such a trail does not allow the participants to discuss or to plan any manipulation of the *S-Network's* content. Talk about deleting content is not allowed at any point in time, even if the jury decides that some content is illegal.

Otherwise, such a trail would be nothing but a negotiation about manipulations between different *mistrust-parties*. The entire concept of preventing manipulative cooperation using measures like prohibition, revelation duty and actively testing the correct revelation behavior with temptation would be completely useless.

But what should be the consequence of a verdict that some content is not legal according to the rules of the *S-Network* – if not some kind of manipulation affecting that content?

B. Non-repudiation Access and the Right of Review

If a jury comprised of representatives from all *mistrust-parties* judges that some content is illegal according to the rules of the *S-Network*, then the consequence should be limited to a slight modification in the access procedure for that content:

Upon a regular read request by a user, the response will be nothing more than a notification that the content requested violates the rules of the *S-Network*. The notification should contain a link to the documentation about the trail in which the content was judged to be illegal. The notification should also warn about the possible legal consequences if the user just insists on his request.

If the user still wants to get access, then he has to confirm that he understood the warning and that he takes responsibility for the possible consequences in a non-repudiation fashion by making a *reliable publication* in the *S-Network*. This publication should contain a good reason and justification why the user should have access despite the fact that the content is said to be illegal. Only after the publication takes place will access be granted.

There are several special situations, in which access even to content like child pornography may be justified:

First of all, any decision that some content is illegal according to the rules of the *S-Network* must be reviewable. The notification telling a user that some content he tries to access was judged to be illegal must also inform the user about his right to initiate another review of that content. It hints to instructions how to initiate further investigations. Such an investigation will require a new jury containing representatives from all *mistrust-parties*. During the new trail, the content whose legality is questioned may have to be accessed by the jury again. However, this has to take place in a controlled environment and with appropriate non-repudiation documentation.

Another situation, in which access to some content that is illegal according to the rules of the *S-Network* may be justified, is for example a trail against the person who committed a crime by producing and publishing that content.

In contrast to deletion, changes to access conditions would not produce irreversible permanent losses. Still, access can be enforced by the entire audience. Content blocked in that way can be revalidated at any point in time and it can be rehabilitated.

On the other hand, if someone enforces his private access to illegal content like child pornography for obviously nothing but personal pleasure, then this person can be identified and prosecuted because enforcing access is only possible in a non-repudiation fashion.

VI. Conclusion, Current State and Future Work

Creating trustworthiness with the help of *mistrust-parties* does not have a risky dependency on a single trusted party. In contrast to other majority based trust concepts, there is no assumption that a majority will behave correctly. Potentially manipulative cooperation is prevented with passive and active measures including the testing of the correct revelation behavior. Game theoretical analysis proves the effectiveness of these measures.

The *S-Network* should allow its participants to make non-repudiation publications and deposits, which could be used for the revelation of cooperating offers between *mistrust-parties* and for the registration of tests for the correct revelation behavior.

Besides the potential application of the *S-Network* as a tool for creating trust with *mistrust-parties*, the *S-Network* itself could be made trustworthy with the concept of creating trust with *mistrust-parties*, too.

If the *S-Network* relies on *mistrust-parties*, then it will require some kind of negotiation and manual cooperation between the *mistrust-parties* to handle illegal content, for example. This paper shows how to do that without making manipulations easier.

An *S-Network* prototype was developed to illustrate how the described ideas could be applied for a concrete technical system.

Further research on the trust creation concept with *mistrust-parties* should try to analyze the behavior of real human beings participating in it. Especially, for measuring the effectiveness of the behavior testing, a psychological experiment would probably be most appropriate. However, to get realistic results, it is necessary to make the participants of such an experiment at least believe that incorrect behavior may have seriously bad consequences for them – which could probably be difficult.

The development of the best possible *trust-distance* function for a specific application is still an open challenge. *Trust-distances* between *mistrust-parties* could change over time. Eventually, a *mistrust-party* may no longer be interested in participating. For a long term system like the *S-Network*, such dynamic changes could be a big problem. Research should try to find best practices for dynamic changes in the set of *mistrust-parties*.

REFERENCES

[1] Michael Abd-El-Malek, Gregory R. Ganger, Garth R. Goodsony, Michael K. Reiter, Jay J. Wylie: Fault-scalable Byzantine fault-tolerant services, Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05 pp. 59-74, ACM New York 2005, ISBN: 1-59593-079-5; doi>10.1145/1095810.1095817

[2] Ross J. Anderson: The Eternity Service, Proceedings of Pragocrypt 1996, http://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/725johnson.pdf (2012-03-12)

[3] Miguel Castro, Barbara Liskov: Practical Byzantine fault tolerance; in proceedings of the third symposium on Operating systems design and implementation, USENIX Association Berkeley, CA, USA 1999, http://people.cs.umass.edu/~arun/cs677/reading/PBFT1.pdf (2012-03-12)

[4] Consultative Committee for Space Data Systems: Reference Model for an Open Archival Information System Blue Book, ISO-Standard 14721:2003; CCSDS 2002, http://public.ccsds.org/publications/archive/650x0b1.pdf (2011-03-28)

[5] Bruce Christianson, William S. Harbison: Why isn't trust transitive?, Lecture Notes in Computer Science, 1997, Volume 1189/1997 pp. 171-176, Springer Verlag Berlin Heidelberg 1996, DOI: 10.1007/3-540-62494-5_16

[6] John Rupert Colville: The Fringes of Power, Downing Street Diaries 1939-1955, Hodder and Stoughton London 1985, ISBN: 0-340-38296-1

[7] Niels Ferguson, Bruce Schneier, Tadayoshi Kohno: Cryptography Engineering, Wiley Publishing, Inc. Indianapolis 2010, ISBN: 978‑0‑470‑47424‑2

[8] John C. Harsanyi: Games with Incomplete Information Played by “Bayesian” Players, I-III, Part I. The Basic Model, Management Science, November 1967, Volume 14, Number 3 pp. pp. 159-182, Institute of Management Sciences 1967,

[9] Poul Jensen: Bitcoin Mining with Trojan.Badminer, Symantec 2011, http://www.symantec.com/connect/blogs/bitcoin-mining-trojanbadminer (2012-01-26)

[10] Audun Jøsang: The right type of trust for distributed systems, Published in: NSPW '96 Proceedings of the 1996 workshop on New security paradigms , ACM New York 1996, http://dl.acm.org/citation.cfm?doid=304851.304877 (2011-11-04) ISBN:0-89791-944-0; doi: 10.1145/304851.304877

[11] Richard M. Karp: Reducibility Among Combinatorial Problems, University of California at Berkeley 1972, http://www.cs.berkeley.edu/~luca/cs172/karp.pdf (2011-07-07)

[12] John Kubiatowicz et al.: OceanStore: an architecture for global-scale persistent storage, ASPLOS-IX Proceedings of the ninth international conference on Architectural support for programming languages and operating systems pp. 190-201, ACM New York 2000, ISBN: 1-58113-317-0; doi>10.1145/378993.379239

[13] Leslie Lamport, Robert Shostak, Marshall Pease: The Byzantine Generals Problem, ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 4 Issue 3, July 1982 pp. 382-401, ACM New York 1982, doi>10.1145/357172.357176

[14] Satoshi Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System, Bitcoin Project 2008, http://bitcoin.org/bitcoin.pdf (2012-01-26)

[15] John von Neumann, Oskar Morgenstern: Theory of games and economic behavior (third edition), Princeton University Press 1953

[16] Heike Neuroth, Achim Oßwald: nestor Handbuch: Eine kleine Enzyklopädie der digitalen Langzeitarchivierung Version 2.0, Verlag Werner Hülsbusch, Boizenburg 2009, ISBN: 987‑3‑940317‑48‑3

[17] Vicky Reich, David S. H. Rosenthal: LOCKSS: A Permanent Web Publishing and Access System, D‑Lib Magazine June 2001 (Volume 7 Number 6); Corporation for National Research Initiatives (CNRI) 2001, http://www.dlib.org/dlib/june01/reich/06reich.html (2010-05-30) DOI: 10.1045/june2001-reich

[18] Uli Ries: Neuer SSL-Gau: Falsches Google-Zertifikat blieb fünf Wochen unentdeckt, Heise Security, Heise Zeitschriften Verlag Hannover 2011, http://heise.de/-1333070 (2011-09-05)

[19] Research Libraries Group, OCLC: Trusted Digital Repositories: Attributes and Responsibilities, RLG Mountain View, CA 2002, http://www.oclc.org/research/activities/past/rlg/trustedrep/repositories.pdf (2011-03-27)

[20] Adi Shamir: How to share a secret, Communications of the ACM Volume 22 Issue 11 pp. 612-613, ACM New York 1979,

[21] Gernot Sieg: Spieltheorie (3. Auflage), Oldenbourg Verlag München 2010, ISBN: 978-3-486-59657-1

[22] P. J. Skevington, T. P. Hart: Trusted third parties in electronic commerce, BT Technology Journal, Vol 15, No. 2, April 1997 pp. 39-44, Artikel: SpringerLink 2004; Print: BT Laboratories / Springer Netherlands 1997, ISSN 1358-3948 (Print), 1573-1995 (Online Journal); Artikel: DOI 10.1023/A:1018628522847

[23] VASCO: DigiNotar reports security incident, VASCO Data Security International, Inc. Oakbrook Terrace, Illinois and Zürich 2011, http://www.vasco.com/company/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx (2011-09-05)

[24] Johannes Viehmann: Secure communication with secret sharing in static computer networks with partition in mistrust parties, 2011 Ninth Annual International Conference on Privacy, Security and Trust (PST) Montreal, Quebec, Canada, July 19-21 pp. 205-212, IEEE Computer Society 2011, Print-ISBN: 978-1-4577-0582-3; Digital Object Identifier: 10.1109/PST.2011.5971985

[25] Marc Waldman, Aviel D. Rubin, Lorrie Faith Cranor: The architecture of robust publishing systems, ACM Transactions on Internet Technology (TOIT), Volume 1 Issue 2, November 2001 pp. 199-230, ACM New York 2001, ISSN: 1533-5399 EISSN: 1557-6051 doi>10.1145/502152.502154

[26] Alma Whitten, J. D. Tygar: Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0; Published in: Proceedings of the 8th USENIX Security Symposium, August 23-26 1999, Washington D. C., USENIX Association, Berkeley 1999, http://www.eecs.berkeley.edu/~tygar/papers/Why_Johnny_Cant_Encrypt/USENIX.pdf (2011-11-04)

[27] R. Yahalom, B. Klein, Th. Beth: Trust Relationships in Secure Systems - A Distributed Authentication Perspective, Symposium on Research in Security and Privacy, 1993 Proceedings pp. 150-164, IEEE Computer Society 1993, Print ISBN: 0-8186-3370-0, DOI: 10.1109/RISP.1993.287635

[28] Philip Zimmermann: The Official PGP User's Guide, The MIT Press 1995, ISBN: 0-262-74017-6