Advances in Networks
2013; 1(2) : 17-24
Published online March 20, 2013 (http://www.sciencepublishinggroup.com/j/net)
doi: 10.11648/j.net.20130102.11

Secure communication with secret sharing for the S-Network using a large set of mistrust-parties

Johannes Viehmann

Fraunhofer Institut FOKUS (MOTION), Berlin, Germany

Email address:
Johannes.Viehmann@Fokus.Fraunhofer.de (J. Viehmann)

To cite this article:
Johannes Viehmann. Secure Communication with Secret Sharing for the S-Network Using a Large set of Mistrust-Parties. *Advances in Networks.* Vol. 1, No. 2, 2013, pp. 17-24, doi: 10.11648/j.net.20130102.11

Abstract: This article presents a solution to ensure secure communication in computer networks by using secret sharing and multiple parties mistrusting each other instead of relying on a "trusted party" or a "web of trust". In contrast to other solutions requiring asymmetric encryption, this concept can provide security even after any possible advance in cryptanalysis and even if unlimited calculation power was available to attack it. But this solution requires the computer network to have special properties. The S-Network, a trustworthy repository, is presented as a potential application. A multi-partition-routing protocol is introduced to optimize the concept for secure communication with secret sharing in the context of the S-Network.

Keywords: Secret Sharing; Trust; Mistrust; Secure Communication; PSMT; S-Network

1. Introduction

Secure communication is required for network applications in various market sectors (e.g. eCommerce, eHealth).

The concept of secure communication with secret sharing in static computer networks with partition in mistrust-parties has been introduced in [15]. This article is an updated version of that publication, extended by describing in detail how the concept could be optimally applied for the S-Network [16], a large scale trustworthy repository for non-repudiation long term preservation.

2. The Problem

2.1. Computer Networks with High Security Requirements

A secure channel between Alice and Bob is a communication channel which allows them to exchange messages in a finite time so that the secrecy, integrity and authenticity of the messages can be ensured and that the temporal order in which the delivered messages were sent by Alice can be reconstructed by Bob. There are provable secure solutions to keep perfect secrecy [12], but integrity, authenticity and the order can only be ensured with arbitrary high probability: Bit sequences passing tests for these could be guessed.

To enable secure communication in a computer network, any two participants should be able to establish a secure channel with each other. The needed level of security varies from application to application. Strong long term security means, that the cryptographic concept should be secure and practically useable for the future – independent from any possible further technical development. For the following, strong long term security requirements are assumed.

2.2. Providing Secure Channels in Big Networks is Hard

It is possible to build an arbitrarily secure channel between any two participants Alice and Bob having strong long term security properties, but that requires that Alice and Bob share an exclusive secret in advance. Alice and Bob have to check their identities and exchange the secret manually. In a small network with only a few participants, it is possible to do such a manual procedure for all possible pairs of participants. But the effort grows quadratically with the number of participants. With several thousand or with several million participants, this is not manageable.

3. State of the Art

3.1. Secure Communication with "Trusted Party"

"Trusted parties", sometimes called "trusted third parties", can be used to provide secure communication between any two participants in computer networks. The idea is that all participants identify themselves only to the "trusted party". For the further usage of the "trusted party" there are different concepts:

Inline usage of a "trusted party": Each participant shares an exclusive key with the "trusted party" so that a secure channel can be built between each participant and the "trusted party". Messages between two simple participants Alice and Bob are first send from Alice to the "trusted party" over a secure channel and then the "trusted party" forwards them to Bob over another secure channel. All messages have to pass the "trusted party", which makes it likely that the central "rusted party" becomes a bottleneck.

Online usage of a "trusted party" as key server: The "trusted party" generates a session key for Alice and Bob so that they can build a direct secure channel between them. See [9] for a solution with this approach. With keys of constant length, this approach reduces the workload of the "trusted party".

At least somehow offline usage of a "trusted party" or of a hierarchy of several "trusted parties" as certification authority (CA): See [7] for a description of public key infrastructures (PKI) and a discussion of the advantages in comparison with a key server. Most important with regard to strong long term security requirement is, that asymmetric encryption (e.g. [10]) is required for PKI solutions. Eventually in the future any asymmetric algorithms can be broken in a relevant short time. For algorithms whose security depends on the assumed difficulty of calculating discrete logarithms or to do prime factorization for large numbers, a theoretical solution for breaking them with quantum computers in polynomial time has already been shown in [13]. For the prime factorization of small numbers, it has been shown that the Shor algorithm really works [8].

Potentially insecure functions used for creating signatures on certificates are another potential point to attack a PKI. See [14] for an attack that takes advantage of the MD5 cryptographic hash function that has been widely used on certificates, but which is not collision resistant. Public keys are typically used to encrypt and exchange symmetric session keys so that messages can be encrypted with more efficient symmetric cyphers like AES [5](i.e. hybrid encryption). However, if the security of the symmetric cypher used for hybrid encryption might eventually be broken, this is an additional vulnerability. Recent advances in cryptanalysis [3] show that this threat should be taken serious.

So in a typical PKI, there are at least three different potentially insecure algorithms that can be attacked independently. It is enough to break just one of these potential weaknesses to break the entire system's security.

No matter how "trusted parties" are used – the security of the communication depends on the fair and always correct behavior of the "trusted party". Why should participants trust the "trusted party"? To control institutions that have so much power is difficult and maybe it is utopian or naive to believe that universal neutrality can at all be enforced in a big network that really matters.

3.2. Secure Communication with "web of trust"

To avoid the need to trust in some single party, the "web of trust" offers a decentralized alternative concept [4]. However, with this approach, it is not possible to achieve legal validity and it requires asymmetric encryption, too. Furthermore, the demands for the users are high as they have to decide whom to trust.

In general, it has also to be questioned whether trust is transitive at all.

3.3. Secure Communication with Secret Sharing

Secret sharing is a technology to split a secret x into a set of n pieces with the property that you need at least t pieces of the set to be able to reconstruct x from that subset. Any subset with less than the threshold t pieces does not reveal any information about x at all. There are several perfectly secure secret sharing systems known, e.g. [11].

Secret sharing can be used to avoid the need to trust a single party by dividing the responsibility for trust related things between several parties. A typical application of secret sharing is to store a secret, for example a secret key.

It is also possible to use secret sharing for "perfectly secure message transmission" (PSMT) over disjoint paths as shown in [6]. These solutions require a set of separated communication channels (called "wires") between sender and addressee. But how these disjunct "wires" could be realized is not mentioned, neither how the identities could be checked nor how authentication could work. In [1] a method to find separate wires is presented, but it provides only paths with disjunct edges, not with disjoint nodes. Hence, it is not a solution for PSMT.

4. Basic Notation and Requirements

Let x and y be bit sequences. The concatenation of x and y prefixed with their identifiers and lengths is noted as $x \circ y$. The symmetric encryption of a bit sequence x with key K is notated as $E_K(x)$. The corresponding decryption is written as $D_K(E_K(x))$. Let $P(x)$ be a function calculating a message authentication code (MAC) of a bit sequence x. K, $E_K(x)$, $D_K(E_K(x))$, $P(x)$ and $x \circ y$ are bit sequences. Messages are bit sequences, too.

The following notation will be used for the set of pieces of a secret sharing split:

$$\mathrm{Zn}_t(x) = \mathrm{Tn}_{t,0}(x), \ldots, \mathrm{Tn}_{t,n-1}(x) \qquad (1)$$

The inverse operation will be noted as:

$$x = \mathrm{Zn}_t^{-1}(M) \mid M \subseteq Zn_t(x) \ \wedge \#M \geq t \qquad (2)$$

The concept introduced in this article makes use of security technologies like secret sharing that do have a threshold. To describe a unique security level for the entire system, a constant threshold Ψ is defined. Ψ is a natural number and it must be greater than two.

4.1. S-Nodes with Partition in Mistrust-Parties

The concept for secure communication presented in this article requires an applicable legal framework and it requires the computer network in which the secure communication

takes place to have the following properties:

The logical addresses of the logical systems within the network must be everlasting, absolute and unique. In the following, such a uniquely addressable logical system will be called an S-Node. S-Nodes added to the network have to be kept accessible by their logical addresses. If an S-Node is not accessible because of some failure, it has to be repaired and restored within a finite time. Such a network may be called a static network.

For each S-Node there must be exactly one natural or juristic person responsible for it in a legal sense: the S-Operator. Let X be the set of all S-Operators in a static network. A partition of such a static network is the split of X into not empty disjoint subsets so that the union of all subsets is X. The subsets of a partition of a static network are called parties.

The solution presented in this paper requires a special partition of the static network with at least Ψ parties so that any two S-Operators belonging to two different parties mistrust each other in a way that they will not cooperate for illegal and therefore potentially dangerous manipulations. Such a partition is called a partition into mistrust-parties.

This mistrust between the parties can be established by a strict geographical, cultural and legal separation, by laws that prohibit certain forms of cooperation explicitly and by active measures to test the correct behavior of the S-Operators in the sense of these laws. Such a test can include fake proposals for building manipulative coalitions, for example. S-Operators have the duty to report illegal offers they get in a standardized fashion. Because any illegal offer could just be a fake for testing the correct reaction, not reporting them might be very risky. Details about the concept of creating trust with a set of mistrust-parties and its application for the S-Network can be found in [16].

MP is used as abbreviation for mistrust-party in general. A certain MP is identified with an index i and noted as MP_i. If an S-Operator belongs to MP_i, all the S-Nodes he is responsible for belong to MP_i, too. Let $\#MP_i$ be the total number of S-Nodes belonging to mistrust-party MP_i.

5. Solution with Secret Sharing and MPs

5.1. Acquaintances, Partisan Forwarding

Two S-Nodes are called acquaintances, if messages can be exchanged between them over an arbitrary secure channel. Therefore the S-Operators of the acquaintances have to check the identities of each other's S-Node's owner and they have to exchange the necessary communication data (including an exclusive secret key). This security critical manual operation is a high effort.

The S-Operators do also have to make sure that data can actually be transmitted between acquaintances in a finite time. Therefore, S-Operators of two S-Nodes that are acquaintances have to negotiate manually appropriate physical channels and they have to provide them to the S-Nodes. For example, one channel could be a direct microwave transmission and the Internet could be used as another single channel between the acquaintances.

Acquaintances do have high responsibility for each other. In order to split responsibilities between Ψ MPs, an arbitrary S-Node S_x must get for each mistrust-party MP_i at least one S-Node belonging to MP_i as acquaintance. This ensures that the identity of the owner of S_x has to be verified for each MP_i at least by one S-Operator belonging to that MP_i whose S-Node becomes an acquaintance.

Because of the high manual effort, an S-Node cannot have more than just a few acquaintances to be practicable.

Only acquaintances may communicate directly with each other. If two S-Nodes are not acquaintances, a message m can be exchanged between them if there is a series of pairwise acquaintances among them and if m can be forwarded from one acquaintance to another along that series. Such an indirect connection is called a forwarding-connection. The forwarding S-Nodes between the sender and the addressee are called forwarders. For the solution presented here, any two S-Nodes must be acquaintances or there must be a forwarding-connection between them.

In contrast to the direct communication with an acquaintance, the forwarding communication cannot take place over a secure channel because a sender and an addressee who are not acquaintances do not have an exclusive shared secret key – they do not even know whether their pretended communication partner exists at all.

To make the communication between S-Nodes which are not acquaintances secure and reliable, there are additional requirements. For any two S-Nodes S_A and S_B belonging to the same MP_i, there must be a connection without any S-Node of all the other MPs involved. This means, that if S_A and S_B are not acquaintances, there must be a forwarding-connection between them so that all the forwarders belong to MP_i. Such a connection within a single MP is called partisan forwarding.

If the network structure within MP_i is like a single ring so that each S-Node belonging to MP_i has exactly two acquaintances in MP_i, there is always a partisan forwarding between any two S-Nodes belonging to MP_i.

5.2. Partition-Routing

The following protocol for partition-routing enables secure communication between any two S-Nodes S_A and S_B that are not acquaintances:

1. Preparation: Let x be the bit sequence to be transmitted. S_A creates a bit sequence x_p containing a random one-time key K_R, the encryption $E_{K_R}(x)$ and a message authentication code $P(K_R\, x)$. So x_p is $K_R \circ E_{K_R}(x) \circ P(K_R \circ x)$.

Let n be $n \in \mathbb{N} \mid n \geq \Psi$. S_A splits x_p with secret sharing:

$$Zn_\Psi(x_p) = \{Tn_{\Psi,0}(x_p), \ldots, Tn_{\Psi,n-1}(x_p)\}$$

Let A_B be the address of the addressee S_B. Let H be additional required header data (e.g. some message number and the current time). S_A generates n split messages τ_i:

$$\tau_i = A_B \circ H \circ Tn_{\Psi,i}(x_p)$$

2. Separation: S_A sends each τ_i over a secure channel to a different acquaintance of S_A not belonging to any of the MPs S_A or S_B belongs to. S_A may not send more than one piece of $Zn_\Psi(x_p)$ into any MP.

3. Check and forwarding: Each forwarding S-Node S_f decrypts and checks messages m arriving over secure channels from its acquaintances.

If m is from an acquaintance not belonging to the same MP as S_f, this acquaintance is the sender S_A. S_f generates an identity confirmation I_{Ai} containing the address of S_A, and additional identity data that was manually exchanged and verified when S_A and S_f became acquaintances. S_f adds I_{Ai} as proof of authenticity to m (i.e. m becomes $\tau_i \circ I_{Ai}$).

Else if m is from an acquaintance belonging to the same MP as S_f, m must already contain an I_{Ai}.

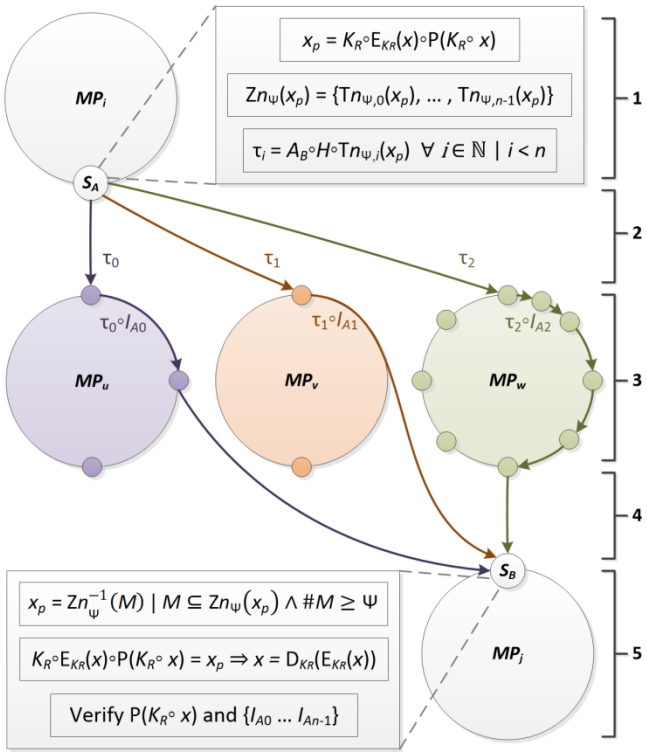

***Figure 1** .Partition-routing with Ψ=n=3. MP_i may stand for the same MP as MP_j, but MP_u, MP_v and MP_w have to be distinct mistrust-parties.*

S_f must forward correct messages m for an addressee S_B according to these rules:

3.a: If S_B is not an acquaintance of S_f, S_f forwards m over a secure channel to the next forwarder, who must be one of the acquaintances of S_f belonging to the same MP as S_f. The forwarder must be chosen so that m gets closer to an acquaintance of S_B.

Continue with step 3 for the next forwarder.

3.b: Else if S_B is an acquaintance of S_f, S_f forwards m over a secure channel to S_B. Continue with step 4.

4. Check and collection: The addressee S_B decrypts and checks messages arriving over secure channels from its acquaintances and extracts $Tn_{\Psi,i}(x_p)$ from τ_i if possible. Correct arriving parts $Tn_{\Psi,i}(x_p)$ and the according identity confirmations I_{Ai} are collected and stored together with the information from which MP they actually were forwarded.

5. Reconstruction and final check: As soon as at least Ψ parts $Tn_{\Psi,i}(x_p)$ of the set $Zn_\Psi(x_p)$ arrived correct at the addressee S_B together with the confirming I_{Ai} from Ψ different MPs, S_B can try to reconstruct x_p from that subset of $Zn_\Psi(x_p)$.

The original data x can be decrypted with K_R:

$$x = D_{KR}(E_{KR}(x)).$$

The integrity can be checked with K_R, x and $P(K_R \circ x)$.

Discussion: Only the sender S_A and the addressee S_B do get more than one piece of $Zn_\Psi(x_p)$ if this protocol is followed properly: In step 2, all the parts $Tn_{\Psi i}(x_p)$ are distributed over secure channels to different MPs. The forwarding of the loop in step 3.a between an acquaintance of S_A and an acquaintance of S_B is a strictly partisan forwarding over secure channels. This means that all the parts $Tn_{\Psi i}(x_p)$ stay in exactly the MP they were sent to at step 2 until they reach an acquaintance of S_B. Only then, at step 3.b, all the parts are send to the same MP, but they are directly send over secure channels to the addressee S_B.

To reconstruct x_p from a subset of $Zn_\Psi(x_p)$, at least threshold Ψ parts of $Zn_\Psi(x_p)$ are required. Any attack to get x_p and therefore any manipulation being more sophisticated than just guessing an entire valid bit sequence must affect at least Ψ forwarders in Ψ different MPs.

The identity confirmation I_{Ai} as proof of authenticity has to be identical from at least Ψ different MPs, too. To cheat requires again that at least Ψ S-Nodes in Ψ different MPs behave incorrect.

5.3. Optimization

With the simple ring like network structure and each S-Node having exactly two acquaintances belonging to the same MP, this is not yet a practicable solution:

The efficiency is unusably low. Partisan forwarding would need great many S-Nodes as forwarders. In the worst case, a message has to be forwarded by 50% of the S-Nodes that belong to MP_i. With thousands or millions of S-Nodes, this would be terribly slow. Messages are not just forwarded – they have to be decrypted, checked and encrypted with another key. On average each S-Node would have to process about 25% of all the messages exchanged by forwarding through its MP.

The total system robustness would be low. If only two S-Nodes belonging to the same MP are temporary not reachable for their acquaintances, the entire ring like network structure would break into two separate segments R and Q so that any partisan forwarding between an S-Node in R and another S-Node in Q would fail.

Robustness against failures in a communication network can be increased by mashing up the network tighter with additional redundant connection possibilities so that alternative routes can be chosen in case of failures [2].By increasing the number of acquaintances within the same MP per S-Node, alternative routes for the partisan forwarding can be created. But that implies also a higher manual effort.

With a few more carefully chosen acquaintances for each S-Node and with a fitting routing concept, a good robustness can be achieved. By doing so, the length of the most efficient partisan forwarding between any two S-Nodes belonging to the same MP can be reduced to a practical value, too. In [15], a decentralized optimization creating well distributed acquaintances in an iterative procedure is shown in detail. Only results are summarized here.

The optimization requires the static address of an S-Node to consist of two components – one identifying the mistrust-party MP_i the S-Node belongs to and the other identifying the S-Node within MP_i. The last is called the Intra-MP-Address. The Intra-MP-Address must be a natural number and it must be unique within its mistrust-party.

For the optimization, for each S-Node S_x belonging to MP_i, two acquaintances belonging to the same MP_i are chosen according to the following rules:

1. The S-Node with the biggest Intra-MP-Address in MP_i smaller than the Intra-MP-Address of S_x becomes an acquaintance of S_x, if such an S-Node exists.
2. The S-Node with the smallest Intra-MP-Address in MP_i bigger than the Intra-MP-Address of S_x becomes an acquaintance of S_x, if such an S-Node exists.
3. Additionally, the S-Node belonging to MP_i with the smallest Intra-MP-Address in MP_i and the S-Node belonging to MP_i with the biggest Intra-MP-Address in MP_i become acquaintances.

The result is again a ring like network structure per MP, but the S-Nodes on that ring are now sorted by their Intra-MP-Address. The ring-distance R(S_A, S_B) between two S-Nodes S_A and S_B belonging to the same MP_i is the number of S-Nodes on the sorted ring that are between S_A and S_B in the shorter direction.

4. Let d be a natural number bigger than one. Each S-Node S_x belonging to MP_i should additionally have those S-Nodes as acquaintances which have a ring-distance of $(d^f - 1)$ with $f \in \mathbb{N} \wedge f < \lceil \log_d(\#MP_i) \rceil$ to S_x. Because the ring-distances might change whenever a new S-Node is inserted into MP_i and making new acquaintances has a high manual effort, for this optimization an approximation to the perfect distribution with enduring well-chosen acquaintances is the best solution.

If each S-Node has these acquaintances then in the optimized partisan forwarding process of a message between two arbitrary S-Nodes S_A and S_B belonging to the same MP_i, at each forwarding step from S_{old} to S_{new} the ring-distance to S_B is reduced according to this formula:

$$R(S_{new}, S_B) \leq R(S_{old}, S_B) - \frac{R(S_{old}, S_B)}{d} \quad (3)$$

Let F_i be the number of S-Nodes required as forwarders between S_A and S_B in an optimized partisan forwarding in MP_i. F_i would then be logarithmic with the number of S-Nodes belonging to MP_i:

$$F_i \leq (d-1) * \lceil \log_d(\#MP_i) \rceil \quad (4)$$

Let A_i be the number of acquaintances each S-Node S_x needs in his own MP to provide such an efficient partisan forwarding. The upper bound of A_i for this optimization is:

$$A_i \leq 2 * \lceil \log_d(\#MP_i) \rceil \quad (5)$$

For $d = 2$, F_i becomes minimal, but A_i becomes maximal, so the most acquaintances per S-Node will be required. Because making many acquaintances means a high manual effort, it probably makes sense to choose a higher d and to accept slightly longer routes in the partisan forwarding.

5.3.1. Foresighted Partisan Forwarding

In the process of partisan forwarding each forwarder S-Node being not an acquaintance of the addressee S_B has to identify the acquaintance that would be the next optimal forwarder. In a network constructed the way shown before, that is the acquaintance with the Intra-MP-Address having the lowest address-distance to the Intra-MP-Address of S_B.

If some S-Node S_F would be the next optimal forwarder, but the current forwarder S_E cannot reach S_F, alternative routes may be tried until S_F is restored and reachable again. Alternative routes are not necessarily less efficient. To find the best alternative route is however more difficult: the address-distances have to be checked further ahead.

Let X be a set of S-Nodes that belong to MP_i. Let B(X) be the set of those S-Nodes belonging to the same MP_i which have at least one acquaintance in set X. For optimal routing, S_E has to choose the S-Node in B(B($\{S_E\}$)\$S_F$)\$S_E$ as next forwarder that has the Intra-MP-Address with the minimal address-distance to the Intra-MP-Address of S_B.

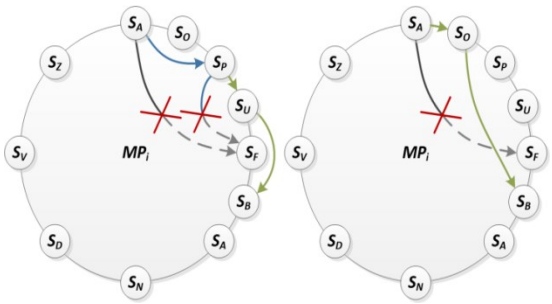

Figure 2. *If some S-Node S_F is not reachable, foresighted partisan forwarding (right side) involving S_O is more efficient than forwarding to the acquaintance S_P with the lowest ring-distance to the addressee S_B (left side).*

Let S_U be an acquaintance of the addressee S_B. If the addressee S_B is not reachable for S_U, other acquaintances of S_B may be tried instead. Each S-Node in B($\{S_B\}$)\$S_U$ that has not yet been tried can be chosen as a preliminary target for the partisan forwarding route. The number of alternative acquaintances for each S-Node in MP_i is between $\lceil \log_d(\#MP_i) \rceil - 1$ and $\lceil \log_d(\#MP_i) \rceil * 2 - 1$.

To avoid endless circling messages all alternative forwarders have to be logged in the message's header.

5.4. Acquaintances in Foreign MPs

In the partition-routing protocol between the first and the last forwarder only partisan forwarding is used to deliver each split message from the sender S_A to the addressee S_B.

For an efficient routing, it is essential to find an acquaintance of S_B belonging to the MP in which the entire partisan forwarding takes place so that it can be used as preliminary target in the optimized partisan forwarding process.

Therefore, those S-Nodes in different MPs that have the same Intra-MP-Address should become pairwise acquaintances. If in any MP_i there is an S-Node $S_{i\chi}$ with the Intra-MP-Address χ, but in another MP_j there is not yet an S-Node having the same Intra-MP-Address χ, $S_{i\chi}$ must get some suboptimal preliminary acquaintance in MP_j if $S_{i\chi}$ needs at least one foreign acquaintance in MP_j.

Let $S_{j\varphi}$ be the S-Node in MP_j having the greatest Intra-MP-Address smaller than χ. Then $S_{j\varphi}$ becomes the suboptimal preliminary acquaintance of $S_{i\chi}$ in MP_j. If later an S-Node $S_{j\chi}$ is added to MP_j, then $S_{j\chi}$ becomes the optimal acquaintance of $S_{i\chi}$ in MP_j. The suboptimal preliminary acquaintance $S_{j\varphi}$ becomes superfluous for $S_{i\chi}$.

5.4.1. Protocol for Optimized Partition-Routing

Let $S_{i\alpha}$ be the sender belonging to MP_i. Let $S_{j\beta}$ be the addressee belonging to MP_j having the Intra-MP-Address β.

The following protocol has to be repeated for each split message of the partition-routing protocol. It delivers such a message m from $S_{i\alpha}$ to $S_{j\beta}$. All the forwarders must belong to the same MP. Let MP_v be that MP. Let S_{v*} be a variable for an S-Node belonging to MP_v.

1. Check for common acquaintance: $S_{i\alpha}$ sends m to an acquaintance S_{v*} belonging to MP_v. If S_{v*} is also an acquaintance of $S_{j\beta}$ continue with step 4.

2. Route to optimal acquaintance: With the foresighted partisan forwarding, the S-Nodes in MP_v try to deliver m to an S-Node $S_{v\beta}$ belonging to MP_v and having the same Intra-MP-Address β as $S_{j\beta}$. If the S-Node $S_{v\beta}$ exists and can be reached continue with step 4.

3. Go to start point for alternative search loop: S_{v*} is set to the S-Node having the biggest Intra-MP-Address smaller than β in MP_v. If the foresighted partisan forwarding did not end at S_{v*}, but at $S_{v\chi}$, m must be send now to S_{v*}. This should always be possible in a single forwarding step because S_{v*} and $S_{v\chi}$ are at least acquaintances.

4. Try to reach addressee: If S_{v*} is an acquaintance of $S_{j\beta}$, m is forwarded to $S_{j\beta}$. End of the protocol.

5. Check if search failed: If S_{v*} has the first Intra-MP-Address (null), there is no acquaintance of $S_{j\beta}$ in MP_v. End of the protocol.

6. Forward to next possible acquaintance: For any S-Node $S_{x\chi}$ having the Intra-MP-Address χ let $\Phi(S_{x\chi})$ be the smallest natural number bigger null for that the equation χ modulo $d^{z-\Phi(S_{x\chi})} = 0$ holds.

Let $L(S_{v*})$ be a subset of $B(S_{v*})$ containing only those acquaintances $S_{v\chi}$ that have $\Phi(S_{v\chi}) \leq \Phi(S_{v*})$.

S_{v*} forwards m to the S-Node of $L(S_{v*})$ having the biggest Intra-MP-Address that is smaller than the Intra-MP-Address of S_{v*}. That S-Node becomes the new S_{v*}.

Continue with step 4.

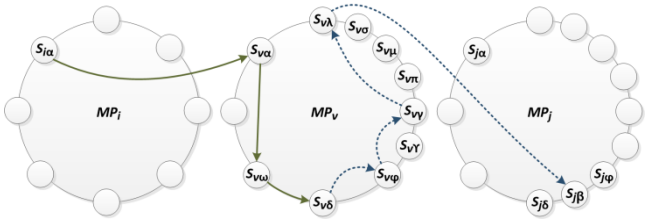

Figure 3. *Searching a suboptimal preliminary acquaintance of $S_{j\beta}$ in MP_v: The S-Nodes $S_{v\Upsilon}$, $S_{v\pi}$, $S_{v\mu}$ and $S_{v\sigma}$ have been added later in the iterative process of making acquaintances [15] – after checking $S_{v\varphi}$ and $S_{v\gamma}$, it is clear that they cannot be acquaintances of $S_{j\beta}$ so they may be skipped.*

Let F be the maximum number of forwarders required for partition-routing of a message split over n mistrust-parties. With the optimization, an upper bound for F is:

$$F \leq \sum_{i=0}^{n-1}(2 * F_i + 3) \tag{6}$$

Note: F_i is the number of S-Nodes required for partisan forwarding (4) within a single MP.

If the number of MPs is bigger than the security threshold Ψ, in case of any disturbance in some MP_k it would be possible to avoid MP_k completely and choose another MP instead to deliver a split message. Or if for the Zn_Ψ n is chosen bigger than Ψ, in up to $n - \Psi$ different MPs there may be failures and the communication still works. Therefore, additional redundancy is superfluous for acquaintances in foreign MPs.

Let G be the number of MPs. Any S-Node will not need more than $G - 1$acquaintances in all foreign MPs together. Per S-Node belonging to MP_i, this leads to a total number TA_i of required optimal acquaintances according to the following formula:

$$TA_i \leq A_i + G - 1 = 2 * \lceil \log_d(\#MP_i) \rceil + G - 1 \tag{7}$$

6. Secure Communication for Users

The solution presented so far is applicable only for secure communication between S-Nodes which should always be online. Human beings and their client systems that are typically often offline should be able to communicate in a secure and reliable way with any S-Node, too.

Users having their own S-Node and who trust in its reliability could use their S-Node as a proxy server. The proxy S-Node could simply forward messages addressing other S-Nodes – using the partition-routing protocol for those other S-Nodes that are not acquaintances.

Of course, users could also manually exchange the required information for building a secure channel to additional S-Nodes belonging to other MPs with the S-Operators of these S-Nodes. Being able to build secure channels to at least Ψ S-Nodes in Ψ different MPs, the users could themselves start the partition-routing protocol. Hence, dependency on a single proxy S-Node being a potential point of failure could be avoided. The disadvantage would be the higher manual effort.

7. Adaption for the S-Network

The S-Network is a trustworthy repository currently developed at Fraunhofer FOKUS. The S-Network combines secure long term data storage and preservation in a computer network with non-repudiation and legal validity. For the future, the S-Network must guarantee to be secure even after any possible technical advance.

The S-Network uses MPs to store and maintain $2 * \Psi - 1$ backup copies in a distributed way so that up to $\Psi - 1$ erroneous copies will not cause losses and that they can automatically be fixed. Secure communication between the systems storing the backup copies is essential. With the concept presented here, the MPs used for data preservation can be used again for the required message exchange and the same level of security and trustworthiness can be guaranteed for both: Only if Ψ or more MPs are incorrect there might be some unwanted incident that cannot be repaired.

Besides bit preservation, the S-Network also has to protect its content. For that access control, the same concepts including provable secure technologies like secret sharing and of course the same MPs can be used. Shares of a secret sharing split are distributed over systems in different MPs so that at least threshold Ψ of them have to cooperate to be able to reconstruct the plaintext. Combining the concepts for bit preservation and access control with MPs will require a significantly higher number of MPs than just $2 * \Psi - 1$ because it must be possible to restore each share without making it easier to reconstruct the plaintext [16].

Secure communication with secret sharing between any two systems (i.e. S-Nodes) S_A and S_B in the S-Network as shown before requires that either S_A and S_B are acquaintances or that partition-routing between them must be possible. Hence there must be at least ΨMPs in which both S_A and S_B have acquaintances. Since making acquaintances is a high manual effort, there should not be more acquaintances than absolutely necessary.

Agreeing on just $\Psi + 1$ MPs in which any partisan forwarding should take place, each S-Node would need at most $\Psi + 1$ foreign MP acquaintances – one in each of these MPs. But the result would be unbalanced: S-Nodes belonging to these "privileged" MPs would have a much higher workload and more responsibility than the others.

7.1. Multi-Partition-Routing

Let G be the total number of MPs. Choosing the foreign MPs in which an S-Node should have acquaintances carefully and using an adapted routing protocol, secure communication between any two S-Nodes S_A and S_B is possible in a completely decentralized way with just $2 * \Psi$ foreign MP acquaintances per S-Node for any $G > 2 * \Psi$.

Let the MPs be indexed with natural numbers starting at zero. Each S-Node S_X belonging to MP_y should have acquaintances in those MPs having an index number $k = (y + d)$ modulo G for any $d \in \mathbb{Z}\backslash 0$ with $|d| \leq \Psi$.

S_X needs also acquaintances in its own MP MP_y for optimized partisan forwarding in MP_y as described above. If each S-Node has these acquaintances, the following multi-partition-routing protocol offers secure communication between any S-Nodes S_A belonging to MP_a and S_B belonging to MP_b that are not acquaintances:

I. Preparation: As described in section 5.2. step 1.

II. Separation: S_A sends each τ_i over a secure channel to a different foreign acquaintance S_f not belonging to MP_b. S_A may not send more than one τ_i into any MP_k.

III. Check and forwarding: Each forwarding S-Node S_f decrypts and checks messages m arriving over secure channels from its foreign acquaintances. If m does not contain an identity confirmation I_{Ai} (i.e. $m = \tau_i$) then S_f adds I_{Ai} as proof of authenticity to m (i.e. m becomes $\tau_i \circ I_{Ai}$).

Let k be the index of the MP S_f belongs to. Let p be the index of the MP from which m was send to S_f.

III.i: If S_f has an acquaintance in MP_b then continue for m with the partition-routing protocol (section 5.2) starting with step 3.b if S_B is an acquaintance of S_f or with step 3.a otherwise. End of this protocol.

III.ii: Calculate the direction d: If $\big((0 < k - p \leq \Psi) \vee (0 < G + k - p \leq \Psi)\big)$ then $d = 1$, else $d = -1$.

III.iii: Calculate short forwards only width w and border index s: Let w be the smallest natural number for which $(a - d * w)$ modulo $G = b$ holds. Let h be the biggest natural number with $(h * \Psi) < G - \Psi$. Then the border index s is: $s = (a + d * h * \Psi)$ modulo G.

III.iv: Calculate index q of the next forwarder's MP: If an integer t with $|t| < \Psi$ and $s = (k + t)$ modulo G exists then $\Delta = w$ else $\Delta = \Psi$.

$q = (k + d * \Delta)$ modulo G.

III.v: S_f chooses an acquaintance belonging to MP_q as next forwarder and sends m over a secure channel to that S-Node. Do step III. for the next forwarder.

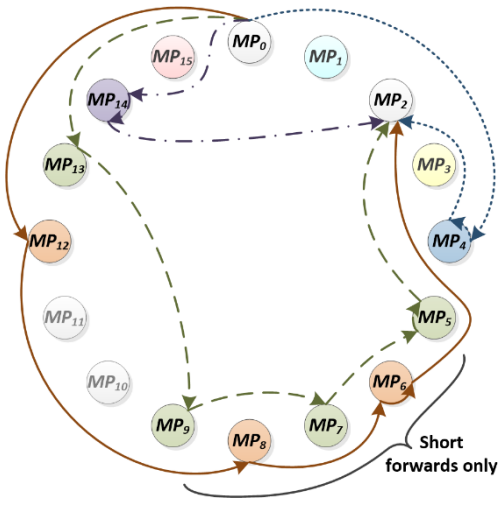

Figure 4. *Multi-partition-routing with G=16 and Ψ=4.*

The multi-partition-routing protocol makes sure that no single MP except MP_b receives more than one piece of $Zn_\Psi(x_p)$. The sender S_A initially separates the τ_i in step II.

As soon as τ_i arrives in an MP having an acquaintance in MP_b (step III.i), only partisan forwarding and a direct forwarding to S_B are performed for τ_i till the protocol ends.

In step III.v τ_i is forwarded to some S-Node belonging to a foreign MP that is not MP_b – and since the protocol continues with step III, this might happen multiple times. Each τ_i goes either only increasing modulo G through the MP indices $(d = 1)$or only in the other direction $(d = -1)$.In each direction there are at most Ψ different shares. Since all the shares routed in the same direction are in different initial MPs after step II and since the distance modulo G between any two indices of these initial MPs is lower than Ψ, shares forwarded in the same direction stay separated as long as the index is decremented or incremented by $\Delta = \Psi$.

It can be shown that $\Delta > \Psi$ is never applied. $\Delta < \Psi$ might be applied if S_A has acquaintances in MP_b to ensure that no share is forwarded in step III.v to an MP in which S_A also has acquaintances. The border s is the last index for which applying $\Delta = \Psi$ is safe. When passing s and if there are no acquaintances in MP_b, only short forwards with $\Delta = w$ are applied. This can happen only for up to$w$ shares since at most the S-Nodes in w initial MPs have no acquaintances in MP_b. As those w initial MPs have indices that are consecutive modulo G it is straightforward to show that separation is preserved.

Multi-partition-routing requires fewer acquaintances in foreign MPs. But there may be up to $G/\Psi + \Psi$ additional forwarders required for each τ_i. An upper bound F_m for the total number of forwarders in multi-partition-routing using n shares can be defined in a similar manner as in (6).

$$F_m \leq \sum_{i=0}^{n-1} \left(2 * F_i + 3 + \left(\frac{G}{\Psi} + \Psi\right)\right) \qquad (8)$$

8. Conclusion

This article presents a concept for secure communication in large computer networks without a “trusted party” or a “web of trust” and without relying on assumptions of complexity theory. Unlike in previous PSMT proposals, a realistic concept to actually create communication paths with disjunct sets of nodes is provided. Depending on the choice of algorithms used to build secure channels between acquaintances, even unlimited calculating power cannot help to break the security of the concept. Hence the solution is applicable where strong long term security is required.

Perfect security is not guaranteed: Attackers could randomly generate a message passing integrity tests. The likelihood that this happens can be reduced by expanding the MAC. The security also depends on the choice of Ψ: A manipulation involving at least Ψ S-Nodes in Ψ different MPs can break the concept. Increasing Ψ and the number of MPs probably only makes sense up to a certain degree. It is crucial to prevent manipulative cooperation among the MPs.

For a trustworthy repository that has to guarantee secure long term serviceability like the S-Network, the solution presented here seems to be a good choice. Using the same concepts and the same MPs to provide bit preservation, secrecy and access control, the S-Network will need more different MPs than the number of MPs actually required for secure communication. Multi-partition-routing introduced in this article keeps the required number of acquaintances in foreign MPs on a low level even in such scenarios, producing only moderately increased forwarding effort and slightly longer message transmission routes.

References

[1] A. Bagchie et al.: Constructing Disjoint Paths for Secure Communication; Lecture Notes in Computer Science, 2003, Volume 2848/2003 pp. 181-195; Springer 2003.

[2] P. Baran: On Distributed Communications Networks; RAND Corporation Santa Monica 1962; http://www.rand.org/pubs/papers/P2626 (2010-01-25).

[3] A. Biryukov, D. Khovratovich: Related-key Cryptanalysis of the Full AES-192 and AES-256; Cryptology ePrint Archive 2009; http://eprint.iacr.org/2009/317.

[4] J. Callas et al.: OpenPGP Message Format; The Internet Society 2007; http://tools.ietf.org/html/rfc4880 (2011-11-11).

[5] J. Daemen, Vincent Rijmen: The design of Rijndael: AES – the advanced encryption standard; Springer 2002.

[6] D. Dolev, C. Dwork, O. Waarts, M. Yung: Perfectly secure message transmission; 31st Annual Symposium on Foundations of Computer Science (FOCS 1990) 1990.

[7] N. Ferguson, Bruce Schneier, Tadayoshi Kohno: Cryptography Engineering; Wiley Publishing, Indianapolis 2010.

[8] IBM Research Division; IBM's Test-Tube Quantum Computer Makes History; First Demonstration Of Shor's Historic Factoring Algorithm; Science Daily 2001.

[9] J. Kohl, C. Neuman: The Kerberos Network Authentication Service; Massachusetts Institute of Technology 1993; www.ietf.org/rfc/rfc1510.txt(2011-02-02)

[10] RSA Laboratories: PKCS #1 v2.1: RSA Cryptography Standard; RSA Security Inc. 2002; ftp://ftp.rsasecurity.com /pub/pkcs/pkcs-1/pkcs-1v2-1.pdf (2011-02-02)

[11] Adi Shamir: How to share a secret; Communications of the ACM v.22 issue 11 pp. 612-613; ACM New York 1979.

[12] C. E. Shannon: Communication theory of secrecy systems; Bell System Technical Journal 28 pp. 656 - 715; Bell Labs 1949.

[13] P. W. Shor: Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer; Bell Labs 1994.

[14] A. Sotirov et al.: MD5 considered harmful today: Creating a rogue CA certificate; 25th Annual Chaos Communication Congress in Berlin 2008.

[15] J. Viehmann: Secure communication with secret sharing in static computer networks with partition in mistrust-parties, proc. of the Ninth Annual Conference on Privacy, Security and Trust (PST) Montreal, pp. 205-212, IEEE 2011.

[16] J. Viehmann: The Theory of Creating Trust with a Set of Mistrust-Parties and its Exemplary Application for the S-Network, proc. of the Tenth Annual Conference on Privacy, Security and Trust (PST) Paris, pp. 185-194, IEEE 2012