# On the Practical Feasibility of Secure Multipath Communication

Stefan Rass and Benjamin Rainer
Technical Faculty, Universität Klagenfurt
Klagenfurt, Austria

Stefan Schauer
Austrian Institute of Technology (AIT)
Safety & Security Department
9020 Klagenfurt, Austria

***Abstract*—Secure multipath transmission (MPT) uses network path redundancy to achieve privacy in the absence of public-key encryption or any shared secrets for symmetric encryption. Since this form of secret communication works without secret keys, the risk of human failure in key management naturally vanishes, leaving security to rest only on the network management. Consequently, MPT allows for secure communication even under hacker attacks, on condition that at least some parts of the network remain intact (unconquered) at all times. This feature is, however, bought at the price of high network connectivity (densely meshed structures) that is hardly found in real life networks. Based on a game-theoretic treatment of multipath transmission, we present theoretical results for judging the networks suitability for secure communication. In particular, as MPT uses non-intersecting and reliable paths, we present algorithms to compute these in a way that is especially suited for subsequent secure and reliable communication. Our treatment will use MPT as a motivating and illustrating example, however, the results obtained are not limited to any particular application of multipath transmission or security.**

***Keywords*—communication system security; multipath channels; privacy; risk analysis; security by design**

## I. Introduction

Private communication is traditionally achieved by means of encryption based on pre-shared secrets or public-key cryptography. The latter is known to never ultimately resist cryptanalysis because of its intractability based fundament, and any symmetric scheme is perfectly secure if and only if it is somehow isomorphic to the one-time pad. For this reason, secure communication services usually require the user to properly manage certificates and cryptographic keys, which is an intricate and error prone process. Multipath transmission (MPT) offers an elegant yet somewhat expensive alternative, by exploiting network path redundancy to achieve security, besides increased reliability. In particular, MPT does not rely on shared secrets, but assumes the network to be sufficiently meshed to prevent an attacker from sniffing on the entirety of a transmission (for the same reason, failure of network components may not cause a complete breakdown, thus increasing reliability of the system at the same time). Under this hypothesis and suitable channel coding schemes, the portion of the information that escapes the adversary's eyes acts in the very same way as a secret key protects information through encryption. Throughout this work, we consider end-to-end communication between two (fixed) nodes in the network by means of MPT.

Communication by MPT, whenever applicable, offers some neat advantages: first, its security can be shown and retained under the assumption that whole parts of the network are fully under the attacker's control, including knowledge of all cryptographic keys and identity credentials. This threat model in particular covers situations in which software vulnerability exploits (e.g., buffer overflows, SQL-injections, etc.) give remote administrative permissions to an external attacker. Exploiting such vulnerabilities (possibly even zero-day attacks) in a whole set of components in the system is covered by the attacker model used in the sequel.

Second, MPT does not rest on any unproven mathematical conjectures or empirical indications, such as public-key and symmetric cryptography do. While both are considered highly trustworthy, insecurity due to human failure in the operation of the system remains a non-negligible threat. MPT naturally achieves risk diversion by removing duties of key-management, and thus somewhat limiting vulnerabilities by human error.

Network reference architectures (topologies) often have some redundancy for robustness against node failures, whose potential for secure communication, however, often remains hidden. Most theoretical treatments of MPT are explicitly devoted to *perfectly secure transmission*, which leads to very strong criteria on the network connectivity (cf. [1]). Whereas perfect privacy demands zero probability for an attacker to learn any of the communicated bits, the slightly weaker notion of *arbitrarily secure transmission* (introduced in III.E) asks for a way to communicate such that the attacker's chance to learn something from the transmission is bounded by some fixed (acceptably low) value $\varepsilon > 0$.

Besides MPTs suitability for risk management and to gain some security against social engineering, there are also good reasons (cf. [2]) to theoretically study MPT, such as fault-tolerant distributed computing, verifiable secret sharing, secure multiparty computation (SMC) or simply the interest in information-theoretic security (like in quantum cryptography [3]). All of these areas at some stage rely on perfectly secure channels, which MPT can create. The need for high-security communication primitives is also motivated by the advent of new computing models like quantum- or DNA-computing. The whole field of post-quantum cryptography [4] accounts for such future security demands, and MPT is another theoretical yet hardly practical alternative.

This work shall be a step towards making MPT more practical. To this end, we derive theoretical results on how MPT can be carried out over networks to get secure and reliable pairwise communication channels. We validate our results using a prototype implementation of the described methods, which works on hierarchically structured networks. That is, we consider communication not only within an enterprise network, but also across different administrative domains. The resulting network models are graphs that model wide area networks, connecting "black-box nodes" that are themselves local area networks (LANs). A security analysis towards secure communication across such a hierarchically structured infrastructure can be based on conventional graph-theoretic algorithms (shortest path and max-flow), which will be at the core of this work.

## II. Related Work

The authors of [5] and [6] discovered MPT as a necessity for perfectly confidential communication, and the work of [7] and [8] complemented this structural result with sharp lower bounds on the necessary communication overhead for such a transmission. Common to all these references are their strong hypotheses on the underlying network graph topology, which gave rise to the game-theoretic treatment in [9] attempting to apply these ideas in general rather than only strongly connected networks. Ever since then, the picture has been extended in various ways, such as by looking for lower bounds on the graph connectivity [8], [10]-[14] implications of synchrony and asynchrony in the transmission [15], [16] impossibility results [17], [18] or applications of MPT in ad hoc and wireless networks [19]-[23].

Multipath transmission is currently under standardization in the course of the Transmission Control Protocol (TCP), see [24]. Experimental simulations have been done towards resource pooling and multipath transmission with a focus on other protocols such as the Stream Control Transmission Protocol (SCTP) and the Multipath RTP (MPTP) with concurrent multipath transmission; see [25] and [26], respectively. Furthermore, [27] introduced an improvement to multipath TCP (MPTCP) where the idea is to introduce fountain code to MPTCP to reduce the impact of paths with lower transmission quality on the overall throughput. Especially in the light of these latest developments, looking at the theoretical possibility of MPT under a more practical environment seems more demanding than ever.

## III. Preliminaries

Let a network be modeled as an undirected graph with node set $V$ comprising all network devices (computers, routers, switches, etc.), and the edge set $E \subseteq V \times V$ giving the (physical or logical) connections between them. Let the nodes be weighted by a security measure $\rho: V \to \mathbb{R}^+$ defined as $\rho(v) := \Pr(\text{node } v \text{ can be compromised})$. The value $\rho$ can be set to either exclude a node from any attack (say, by organisational assumptions, non-cryptographic protections or contractual regulations if $v$ models the subnet of a transmission service provider), or to express the (pessimistic) assumption that zero-day exploits or other intrusions on the device $v$ may be expected. In that case, we may put $\rho = 1$, which can also be done if the „probabilistic security" of a node is difficult or impossible to obtain reliably. In other cases, assigning appropriate resilience to each node is left to probabilistic security models or general statistical approaches (e.g., beta-reputation [28], [29]). Note that it is not necessary to weight edges, as an edge $u - v$ with weight $\rho$ can be replaced by two unweighted edges to an artificial node $w$ with weight $\rho$ in between: $u - w - v$.

We write $V(G)$, $E(G)$ to denote the vertex- and edge-sets of a graph $G$. Moreover, given subsets $V' \subseteq V, E' \subseteq E$, we write $G(V', E')$ for the induced subgraph. The symbol $\mathcal{P}(V)$ denotes the power-set of $V$.

The *degree* of a node $v$ is the number of edges that $v$ is part of. For two distinct nodes $s, r \in V$, hereafter representing the *sender* and *receiver* of a transmission, an $s$-$r$-*path* or *wire* in $G$ is a subgraph $\pi(V', E')$ of nodes and edges, where $\{s, r\} \subseteq V$ and the degree of all nodes $v \neq s, r$ is two, and only $s$ and $r$ have degree one. That is, a path is a subgraph that forms a connection from $s$ to $r$ as a sequence of nodes and edges. The set of all $s$-$r$-paths is called the set of *wires*, and denoted by $W_G(s, r)$. This set again constitutes a subgraph of $G$. Two $s$-$r$-paths $\pi_1, \pi_2$ are called *(node-)disjoint*, if $V(\pi_1) \cap V(\pi_2) = \{s, r\}$.

Random variables are as well denoted by uppercase letters, as those will exclusively be set-valued (thus justifying the overloaded notation here). We write $X \sim \mu$ whenever the distribution of $X$ is $\mu$. The symbol $X \xleftarrow{\mu} \Omega$ denotes a random draw $X$ from a set $\Omega$, according to the probability distribution $\mu$ (supported on $\Omega$).

### A. Adversary Model

Assuming that nodes in a network have common or similar security properties, say by running on the same firmware or residing in the same physical location, our attacker model is a family of subsets of $V$ that share common vulnerabilities. This models situations in which exploits on several machines create a path through the network towards the valuable data (an *attack path*). Formally, we model the attacker as a subset $\mathcal{A} \subset \mathcal{P}(V)$, where a set $A \in \mathcal{A}$ describes an attack scenario in which the adversary has gained full access and control over all nodes in $A \subseteq V$ (elsewhere called an *adversary structure* [30], [8]. As the attacker's behavior is unknown, let $Y$ be a random variable supported on $\mathcal{A}$, whose realization corresponds to the mounting of an attack. We will need this later for our formalization in section III.C.

### B. Abstract MPT and its Prerequisites

We write $\Pi(m, X)$ to denote a general MPT protocol $\Pi$ that transmits a message $m$ over a network, taking random coins $X$ to make internal decisions. In particular, the random variable $X$ is assumed to steer the choices of transmission paths, besides other protocol-specific actions that use randomness. As our upcoming treatment of security will heavily rely on what paths are chosen for transmission, and what nodes have been attacked successfully (random variable $Y$), let us write $X \in W_G(s, r)$ for the random variable that selects transmission paths. A particular transmission of a secret message $m$ from a sender $s$ to a receiver $r$ then works by selecting transmission

paths by sampling from $X$ and running the MPT protocol $\Pi(m, X)$ over the chosen set of paths. This captures most of the theoretical work on MPT cited in section II, where the set of paths is always assumed to be fixed prior to the transmission, when some additional assumptions are adopted, which commonly appear implicitly throughout the MPT literature (e.g., [6]-[8] and others):

*1) The network topology is reliably known, so that paths can be selected. Here, we can allow only for partial knowledge of the topology, treating all parts of the network with unknown topologies as black boxes (and taking advantage of the hierarchical graph modeling mentioned above and detailed later).*

*2) Packets can be routed over fixed chosen paths. Although such source routing is an existing yet mostly disabled feature of the internet protocol (IP), such routing can be over virtual LANs resembling the paths (network layer 2), or using port routing on layer 3 (transport).*

*3) The routing is reliable in the sense that a packet does not deviate from its designated transportation route. Although we explicitly assume this here, one can relax this assumption to a limited extent, while still retaining the possibility of secure communication [31]. We do not explore this any further here.*

*4) An exhaustive set of scenarios can be identified under which an adversary can attack. This is usually the result of topological vulnerability analysis (searching for attack paths and attack graphs), the results of which make up the abstract family $\mathcal{A} \subseteq \mathcal{P}(V)$ of component sets that are vulnerable to a specific attack. In section IV.B, we show how to derive an approximation of $\mathcal{A}$ from the anyway required computation of node-disjoint paths.*

We stress that these assumptions exclude adversaries being able to mimic a certain number of virtual nodes (*Sybil attacks*), which would mean that the network topology information is itself unreliable. It is subject of future work, yet outside the scope of this article, to consider adversaries with such power.

### C. Simple MPT – An Example

To motivate the general treatment and show how secure MPT may work, we use an inefficient yet simple example protocol. Suppose that the network $G(V, E)$ permits $n$ node-disjoint $s$ - $r$ -paths $\pi_1, \dots, \pi_n$, where $s, r \in V(G)$. Let the message be a bitstring $m$, which the sender writes as $m = s_1 \oplus s_2 \oplus \cdots \oplus s_n$, where $\oplus$ is the bitwise XOR. This representation is immediately found by choosing $n-1$ random strings $s_1, \dots, s_{n-1}$, and putting $s_n := m \oplus s_1 \cdots \oplus s_{n-1}$. We call each $s_i$ a *share* to $m$. From a cryptographic viewpoint, this is an $n$-out-of-$n$-sharing, as no subset of less than $n$ of the shares reveals any information on $m$. This is easy to see, as any unknown share, say $s_k$, acts as a one-time pad encryption of $m$.

For the same reason, an attacker is required to get all $n$ shares in order to correctly recover $m$. So, if each share $s_i$ travels over a distinct path $\pi_i$, then no set $A \in \mathcal{A}$ with cardinality $< n$ will suffice to disclose $m$. Consequently, any attack on less than $n$ nodes will necessarily fail, and only those attack scenarios $A \in \mathcal{A}$ will be successful (for the adversary), in which all $n$ paths are intercepted.[1].

Recall that $X, Y$ were random variables describing the (random) path choices and (unknown) compromised node sets. Let us introduce an (efficiently decidable) predicate $\phi(X, Y)$ that equals 1 if and only if attack $Y$ fails under transmission scenario $X$. Then $\phi(X, Y)$ is also a binary random variable, which measures the success rate of the (generic) protocol $\Pi(m, X)$, where $X$ is under the sender's control, and $Y$ is coming from the adversary and thus unknown. The next section will define security in terms of the predicate $\phi$, more precisely, its expectation.

### D. Security Measures

It is common in cryptography to capture attack scenarios in abstract "games". Security is then defined in terms of the likelihood for the attacker to win the game.

---

$Game_{MPT}^{\mu,\nu}(G, s, r, \Pi)$: Let $m$ be the message to be sent over $G$ from $s$ to $r$.

1) The (honest) sender chooses $X \overset{\mu}{\leftarrow} W_G(s, r)$.

2) The adversary conquers a node-set $Y \overset{\nu}{\leftarrow} \mathcal{A}$.

3) The protocol $\Pi(m, X)$ is executed, resulting in either success ($\phi(X, Y) = 1$) or failure ($\phi(X, Y) = 0$).

4) Output $1 - \phi(X, Y)$ as the game's outcome.

---

The security of an MPT transmission is the attacker's *advantage* in winning the above game,

$$Adv_{MPT}^{\mu,\nu}(s, r) := \Pr_{\mu,\nu}\left[Game_{MPT}^{\mu,\nu}(G, s, r, \Pi) = 1\right].$$

A widely unexploited feature of MPT is the degree of freedom to choose the paths (in particular, all prior research seems to keep the path choices $\mu$ fixed a priori in an attempt to guard against every scenario described by $\mathcal{A}$. We take a more general direction here, by using game-theory to optimize the honest party's behavior ($\mu$) and the attacker's behavior ($\nu$) simultaneously. This leads to the computation of a (Nash-) equilibrium $\mu^*, \nu^*$ for $Game_{MPT}^{\mu,\nu}(G, s, r, \Pi)$, which satisifes

$$Adv_{MPT}^{\mu,\nu^*}(s, r) \le Adv_{MPT}^{\mu^*,\nu^*}(s, r) \le Adv_{MPT}^{\mu^*,\nu}(s, r) \qquad (1)$$

for any distributions $\mu, \nu$.

The appeal of imposing a zero-sum hypothesis on the competition between the sender and the attacker lies in the validity of the right of the above inequalities under any real behavior of the attacker. Put differently, if the advantage is computed as the Nash-equilibrium solution $Adv_{MPT}^{\mu^*,\nu^*}(s, r)$, then this value lower-bounds the success-rate of the MPT-protocol $\Pi$ *regardless* of what the attacker actually does (see [9] and [32] for formal proofs), conditional on the only hypothesis that no attack outside $\mathcal{A}$ is mounted (in which case, however, any security analysis would fail).

[1] More general approaches (e.g. [6], [8]) replace this simple encoding by a more robust and flexible polynomial secret sharing, or equivalently, a Reed-Solomon encoding. This enjoys both, robustness against path failure, and perfect secrecy against eavesdropping on a certain limited number of wires.

The converse probability, called the *vulnerability*,

$$\rho(s,t) = 1 - Adv_{MPT}^{\mu^*,\nu^*}(s,r). \quad (2)$$

measures how many messages are discovered by the attacker, relative to the entire lot of transmitted information. This *upper bounds* the likelihood for an attack, which is why we can consistently use the same symbol as for the node-weights. The important difference here is that (2) refers to a whole transmission from $s$ to $r$, rather than a single node.

### E. Definitions

Since our adversary can control his advantage via clever choices about the compromised nodes, we extend the usual model of an adversary structure towards a *probability distribution supported on an adversary structure*.

*Definition 1.* Given a network $G(V,E)$, an *adversary* is described by a probability distribution $\nu$ supported on a family $\mathcal{A} = \{Y_1, Y_2, \dots, Y_k\} \subseteq \mathcal{P}(V)$ of possibly compromised nodes. Concerning semantics, we define $\nu_i := \Pr(\text{nodes in } Y_i \text{ get compromised})$. The attacker is computationally unconstrained regarding the processing of information in his possession.

Imposing no limit on the attacker's power is actually not unrealistic under our adversary model: by assumption, once the attacker has conquered a set $Y$ of nodes, we assume full control over all nodes in $Y$, including full knowledge about the data residing in these nodes. Hence, the attacker can compute anything that the honest parties could compute too, thus precluding (and invalidating) all intractability assumptions that would otherwise establish security of conventional public-key and symmetric cryptography.

Nevertheless, to keep things practical, we need to impose bounds on the computational power of the honest parties (no transmission scheme can feasibly handle inputs of exponential size), and on the size of the adversary structure $\mathcal{A}$ (to keep the running time of our algorithms within reasonable bounds).

*Definition 2.* [5] A transmission is called $\varepsilon$-*private* (for $\varepsilon > 0$), if for any two plain text messages, the corresponding random ciphertexts have distributions that are statistically indistinguishable (distant in the 1-norm) up to a difference of $2\varepsilon$. A transmission is called $\delta$-*reliable* for $\delta > 0$, if with probability at least $1-\delta$, the delivery is correct. A transmission is $(\varepsilon,\delta)$-*secure*, if it is both, $\varepsilon$-private and $\delta$-reliable, and it is called *perfectly secure* if $\varepsilon = \delta = 0$. The transmission is called *efficient*, if its bit- and round-complexity is polynomial in the size of the network and the message, as well as $\log 1/\varepsilon$ and $\log 1/\delta$, wherever $\varepsilon > 0$ and/or $\delta > 0$.

The vulnerability definition (2) is naturally linked to the above security concepts by the following fact:

**Theorem 1.** [9] Assume a Nash-equilibrium behavior $\mu = \mu^*$ for the honest parties in $Game_{MPT}^{\mu,\nu}(G,s,r,\Pi)$, and let $\rho = \rho(s,r)$ be computed as in (2) for a predicate $\phi$. If $\phi = 1$ indicates a successful confidential (not necessarily correct) transmission using $\Pi(m,X)$, then $\Pi$ is $2\rho$-private. Alternatively, if $\phi = 1$ indicates a successful correct (not necessarily confidential) transmission using $\Pi$, then $\Pi$ is $\rho$-reliable.

This is a major difference to the treatment common in cryptography. As opposed to the abstract games serving for complexity-theoretic reduction arguments towards security proofs, for multipath transmission we explicitly attempt to execute the game in reality. The optimal way of doing this is determined by techniques of game-theory, whose details are not relevant here (see [9] for a full treatment). Theorem 1 is the permission to use the following as our security definition:

*Definition 3.* A protocol is called $\varepsilon$-*secure*, if there is some $\mu_\varepsilon$ such that $Adv_{MPT}^{\mu_\varepsilon,\nu}(s,r) \leq \varepsilon$ for every distribution $\nu$ over $\mathcal{A}$. The protocol is said to achieve *arbitrarily secure message transmission (ASMT)*, if it is $\varepsilon$-secure for every $\varepsilon > 0$. A 0-secure protocol is said to achieve *perfectly secure message transmission (PSMT)*.

Notice that ASMT can achieve the same level of secrecy as any conventional encryption, if we set $\varepsilon$ to be the likelihood of guessing the key (e.g., $\varepsilon = 2^{-128}$ for a 128 Bit AES key). However, and more generally than PSMT, the Nash-equilibrium based analysis of security is extensible towards multiple interdependent security goals in a consistent way [33]. Other concepts like Definition 2 are much more difficult to handle or extend.

Obviously, PSMT implies ASMT. The converse is not true, since ASMT allows for a strictly positive residual chance of disclosing the message, which PSMT explicitly rules out. The advantage of ASMT over PSMT, however, is that the former may be possible in cases where PSMT is ruled out by insufficient graph connectivity. The remainder of this work is dedicated to a discussion on how to set up the transmission game $Game_{MPT}^{\mu,\nu}(G,s,r,\Pi)$ so that either ASMT is possible, or neither PSMT nor ASMT are achievable (provably).

Going through the literature on MPT (and also section III.C), one finds the idea of "bypassing" the attacker by virtue of using multiple paths to be a common denominator among most (if not all) MPT protocols. The next definition captures this more explicitly.

*Definition 4.* Let a (directed) graph $G(V,E)$ and a subset $Y \subseteq V$ be given. For two distinct nodes $s,r$, we define the *(directed) residual $s$-$r$-capacity of $G$ w.r.t. $X$*, denoted as $\kappa_Y(s,r)$, as the number of $s$-$r$-paths that circumvent $Y$, i.e., the number of paths that do not go through any node in $Y$.

The residual capacity is important as it characterizes the possibility or impossibility of secure transmission based on whether a person-in-the-middle attack between $s$ and $r$ is possible (or the attacker can be circumvented).

**Proposition 1.** ASMT from $s$ to $r$ is possible against an active adversary $\mathcal{A}$, if and only if the residual capacity $\kappa_Y(s,r) \geq 1$ for all $Y \in \mathcal{A}$.

*Proof.* For the necessity, suppose that PSMT is possible then the likelihood of a message to circumvent any $Y \in \mathcal{A}$ is 1. Then for every $Y \in \mathcal{A}$ there exists at least one path $X$ that avoids $Y$, hence the residual capacity is $\geq 1$. Conversely, if the residual capacity is $\geq 1$, then the following protocol can do

arbitrarily secure message transmission: put $n := \min_{Y \in \mathcal{A}} \kappa_Y(s,r)$ and observe that $n \geq 1$ by hypothesis. Now (as described in section II.B), let us divide the message $m$ by a $t$-out-of-$t$ -sharing as $m = s_1 \oplus s_2 \oplus \cdots \oplus s_t$ , and transmit $s_i$ over another distinct path (exhausting the set of available paths). Let $p$ denote the probability of $X_i$ to bypass all compromised nodes. Furthermore, $n \geq 1$ implies that $p > 0$ (note that the distributions $\mu, \nu$ that control the choice of paths and compromised nodes can be omitted, since $p > 0$ for any $\mu, \nu$ ). Since recovery of the message requires all shares $s_1, \ldots, s_t$ (the predicate $\phi$ would thus be defined as 1 in this case only), the likelihood for all these getting caught is $(1-\rho)^t = Adv_{MPT}^{\mu,\nu} \to 0$ as $t \to \infty$. So $Adv_{MPT}^{\mu,\nu} < \varepsilon$ for any given $\varepsilon > 0$, if $t$ is chosen sufficiently large. □

The rather simple transmission protocol used in the proof of Proposition 1 is clearly suboptimal in terms of communication overhead (yet its overhead is polynomial in $1/\varepsilon$ , thus nevertheless being efficient in the sense of Definition 2). Its meaning for our investigation, is merely to prove that ASMT is possible based on a certain graph connectivity. Nevertheless, the security of multipath transmission is in any case determined by the likelihood to circumvent compromised nodes. Consequently, a larger number of paths to choose from will eventually maximize the chances of bypassing the adversary. Our algorithms for path enumeration given in section IV.A will therefore attempt to give a maximal number of such paths.

## IV. Setting Up The MPT-Game

Our main objective in the following is to practically instantiate $Game_{MPT}^{\mu,\nu}(G,s,r,\Pi)$ for a given protocol Π, which for any MPT protocol requires two initial tasks: 1) enumerate a maximal set of paths to choose from, and 2) compute the most vulnerable points in the network (as those may be the most likely targets for an attack).

As an exhaustive enumeration of paths is infeasible (usually, there are exponentially many of them), and an exhaustive enumeration of attack strategies is also difficult, we shall "approximate" both ingredients to $Game_{MPT}^{\mu,\nu}(G,s,r,\Pi)$, and use the approximations $\widehat{W}_G$ and $\hat{\mathcal{A}}$ in place of $W_G(s,r)$ and $\mathcal{A}$ throughout the rest, where the goodness of this approximation will be in the center of attention now.

### A. Enumerating Transmission Paths

A useful result from graph theory (Theorem 5.17 in [34]) equates the number of node-disjoint paths between any two nodes $s,r$ in $G$ to the cardinality of a minimal vertex cut between $s$ and $r$, where an $s$-$r$-cut} in a graph $G(V,E)$ is a set $C \subseteq V$ so that any $s$-$r$-path $\pi$ has $V(\pi) \cap C \neq \emptyset$. For the adversary, conquering a cut is equivalent to mounting a person-in-the-middle attack, which is the only way to effectively intercept an MPT transmission. The problem of finding a minimal vertex cuts is computationally simple and solvable by min-cut-max-flow techniques. The latter basically work by computing node-disjoint paths (cf. [35]), which we need anyway. So, the node-disjoint paths can be used to run MPT, while the graph cuts that are computed alongside can be used to identify neuralgic points in the network that potentially match attack strategies in $\mathcal{A}$ (hence „approximate" $\mathcal{A}$).

So far, there is no real computational difficulty, whenever the number of paths is known and constant. Here is the problem: the number $k$ of paths is determined by the power of the adversary in terms of how many nodes can be corrupted at the same time. Moreover, even if this number is known, if any logical connection within the network would use the maximal number of paths, network congestions become highly likely and congestion control may randomly cause paths to intersect. Such congestions can be even due to the adversary; a scenario that has received attention in [31], where the reliability of routing was in the focus of interest. This shows another limitation of the aforementioned references in terms of practicability. Although [8] provides a sharp limit on the minimal required amount of bandwidth for MPT, and many results assuring perfectly private communication under certain graph connectivity assumptions are known, a network whose bandwidth and connectivity undercut the theoretical minimum requirements for PSMT may rule out the latter, yet still enable ASMT.

Let $\widehat{W}_G$ denote the set of all candidate paths, from which the protocol Π may select a subset for transmission. This is basically an approximation to the set $W_G(s,r)$ , whose cardinality may be exponential (and thus infeasible to handle). To set up $\widehat{W}_G$ , we compute the maximal number of node-disjoint $s$-$r$-paths by running a conventional edge-capacity based min-cut-max-flow algorithm on a transformed version of $G$. The transformation is well-known and detailed in [35] It basically substitutes each node $v$ in the graph by two connected nodes $v_{in} \to v_{out}$, setting the capacity of the internal (directed) edge to 1 so as to limit the flow through this node. All other undirected edges $u - v$ are replaced by two directed edges $u \to v$ and $v \to u$, both of which have infinite capacity. The only exception to this rule are the sender's node $s$, from which only outgoing edges are drawn, and the receiver's node $r$, having only incoming edges.

It is easy to see that an integral maximal $s$-$r$-flow over a network with vertex capacities all set to 1 equals the number of node-disjoint $s$-$r$-paths. We can also permit intersections of paths in certain selected nodes, say in case that a node has zero vulnerability, and thus cannot be conquered in any scenario in $\mathcal{A}$. To let paths intersect at a node, we simply increase its internal edge weight from 1 to ∞, so that any number of paths may pass this node.

Taking a closer look at the internals of the Ford-Fulkerson min-cut-max-flow algorithm, we see that the algorithm in each step increases the flow by searching for another flow-augmenting path through nodes with positive remaining capacity. On this *residual network*, we can construct a flow-augmenting path by looking for the "most secure" $s$-$r$-path. This is easy by virtue of a shortest-path algorithm that takes the vertex security $-\log \rho_v$ as the length of the internal edge from $v_{in} \to v_{out}$, taking all other edges (connecting different nodes to each other) with zero length. Observe that we now have *two* different weights assigned to each node, one of which is either 1 or ∞ to limit the number of paths through this node; while the other weight $-\log \rho_v$ serves to compute another flow-augmenting path by taking the (next) most secure $s$-$r$-path.

**Algorithm 1 Recursive path enumeration**

**Input**: source and destination sets $S, R \subseteq V$ in a graph $G(V, E)$, as well as a constant $M \in \mathbb{N}$.

**Output**: a set of non-intersecting $S$-$R$-paths.

**Steps**: 1) Insert two artificial nodes $\nu_s, \nu_r \notin V(G)$ and connect $\nu_s$ to every node in S and $\nu_r$ to every node in R. Call the resulting graph $G'$.

2) Compute a maximal $\nu_s$-$\nu_r$-flow with vertex capacities and a minimal cut $C \subset V(G)$ on $G'$.

3) If there are intermediate nodes between $S$ and $C$, respectively $S$ and $R$, call Algorithm 1 to compute node-disjoint paths from $S$ to $C$, resp. from $C$ to $R$.

4) Assemble the partial $S$-$C$ and $C$-$R$ paths to $S$-$R$-paths and put them into a set $P_1$.

5) Construct $\widehat{W}_G$ from the so-obtained "ground set" $P_1$ of paths by selecting $M$ sets of disjoint paths with the desired cardinality.

The max-flow algorithm itself remains intact by this modification, since the Ford-Fulkerson algorithm does not prescribe the method by which the flow-augmenting path is to be found (see [37]).

Observe that the so constructed set of node-disjoint paths is not maximal, as local alternative routes may be taken. In order to exhaust the set of existing $s$-$r$-paths and to construct node-disjoint selections from these, we recursively apply the max-flow technique between $s$ and the minimal $s$-$r$-cut $C$, and between $C$ and $r$. Intuitively, this is correct since the cut is such that any $s$-$r$-path must traverse it at some stage, yet alternative routes towards the cut may exist in the network and need to be found. Algorithm 1 describes this recursive procedure to constructs a set of transmission paths between $s$ and $r$, where $S = \{s\}$ and $R = \{r\}$ at the initial invocation. The role of the constant $M$ is to limit the resulting number of paths to a polynomial number (as will be detailed in the proof of Lemma 1).

Figure 1 illustrates a single step in this algorithm using the graph from Figure 3 as an example. First, we compute the minimal $s$-$r$-cut $C = \{5\}$ and a maximal flow (shown bold) in the initial graph, and then moves onwards to compute a multi-source-multi-sink flow from $S = \{s\}$ to $R = C = \{5\}$. For that matter, it introduces two artificial nodes $\nu_0, \nu_1$ with infinite capacities along incident edges, and computes a minimal cut between $\{s\}$ and $\{5\}$ as $C = \{2,4\}$ in this recursion step, where the corresponding maximal flow is shown bold again. Sparing further recursions for brevity, the assembly of the so-found partial paths between any sub-cuts gives the boldly shown paths illustrated in the right side of Figure 1. The union of all these paths, making up the *ground set* $P_1$. The paths in $P_1$ may indeed intersect and form the basis from which we can select disjoint paths to create $\widehat{W}_G$.

Based on Proposition 1, our path enumeration algorithm attempts to maximize the residual capacity subject to the poly-bound constraint of the honest player. More formally, it seeks

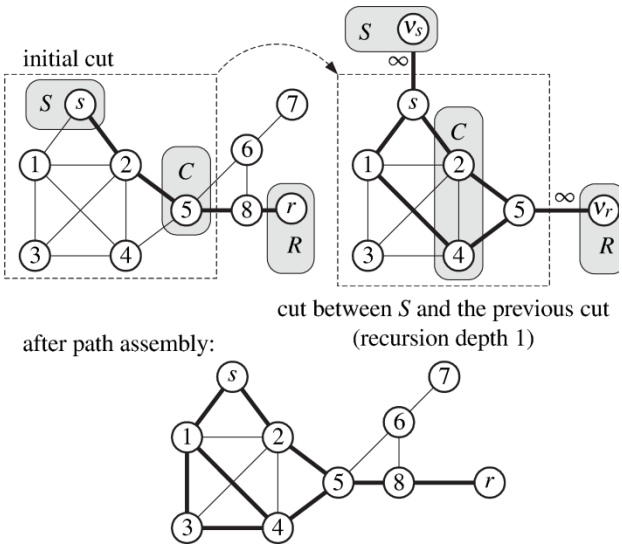



Fig. 1 Illustration of Algorithm 1

the set $\widehat{W}_G$ so that the graph restricted to the paths in $\widehat{W}_G$ only, has maximal residual capacity.

**Lemma 1.** Let a graph $G(V, E)$ with $n$ nodes be given. Algorithm 1 outputs a set $\widehat{W}_G$ of size $n^{O(\log M)}$, with the following property: for any fixed compromised set $Y \in \mathcal{A}$, the residual capacity w.r.t. $Y$ is maximal. Moreover, we cannot get better security by using any more paths than in $P_1$ already, i.e., $\widehat{W}_G$ in that sense is "maximal".

*Proof*. Write $G'$ for the graph consisting only of the chosen $s$-$r$-paths. Take any $Y \in \mathcal{A}$ and assume that the residual capacity $\kappa_Y(s, r)$ is not maximal, i.e., there is a $s$-$r$-path $\pi$ bypassing $Y$ that is not captured by $\widehat{W}_G$. Take the two closest cuts $C_1, C_2$ that "enclose" $Y$ from the left and the right (coming from $s$ and $r$ respectively). Then the $C_1$-$C_2$-flow can be augmented by the path bit of $\pi$ between $C_1$ and $C_2$, thus contradicting the maximality of the flow. Hence, there cannot be such a path $\pi$ unless it has already been found and included in the output at some stage of the recursion.

To see the "maximality" of $\widehat{W}_G$, assume that we would add another path $\pi^* \notin P_1$ and use the set $P_1 \cup \{\pi^*\}$ for constructing $\widehat{W}_G$. Since $\pi^* \notin P_1$, it must differ from at least one path $\pi \in P_1$ in at least one node. So let the paths $\pi$ and $\pi^*$ partially coincide on $\pi_0, \pi_2$, and consider the different bits $\pi_1, \pi_1^*$, as illustrated in Figure 2. Take the bounding cuts $C_1$ and $C_2$ as constructed by Algorithm 1 between which $\pi_1, \pi_1^*$ are located. By construction, the $C_1$-$C_2$-flow is already maximal, so $\pi_1^*$ cannot be more reliable than $\pi_1$. Therefore, the route over $\pi_1^*$ is less secure than the route $\pi$, and adding $\pi^*$ to the ground set $P_1$ is pointless when constructing $\widehat{W}_G$.

It remains to investigate the cardinality of $\widehat{W}_G$. The number of strategies that our divide-and-conquer algorithm digs up is determined by $M$ as follows: let $n = |V|$ be the number of nodes in the network, and let $T(n)$ count the number of strategies constructed in the recursive manner as sketched in Figure 1. Algorithm 1 divides the graph with $2n$ nodes into two

halves of size $n$, and combines the flows in each path accordingly into a set of node-disjoint paths from $s$ to $r$. Hence, $T(2n) \leq 2T(n) + r(n)$, where the remainder term $r(n)$ counts the number of ways in which the partial paths can be connected. The recursion reaches the trivial case after no more than $O(\log n)$ steps. So, if we enumerate no more than a constant number of $M$ connections in the path assembly, then the overall number of paths returned by the algorithm is no more than $M^{\log_2 n} = \left(2^{\log_2 M}\right)^{\log_2 n} = n^{\log_2 M}$ and thus polynomial in $n$. □

The particular choice of $M$ affects how many paths are returned by the algorithm, however, Algorithm 1 returns a limited selection of the most secure paths. Thus, choosing smaller values of $M$ may yield suboptimal network utilization as some perhaps secure routes remain unused. In that case, there may be no security achievable against the given adversary, if the paths are selected from this limited family only. In that case, one can increase $M$ to find more paths in order to ultimately bypass the attacker and gain security of MPT.

Our prototype limits the number of enumerated paths in the matching to $M = 10{,}000$. Moreover, the experiment showed that we can take advantage of the loose connectivity of scale-free network topologies, such as observed on large-scale networks like the Internet. For many of our experiments, the sizes of the cuts (and flows) were actually small, so that even the full enumeration gave a feasible number of path combinations.

Constructing the flows by virtue of most secure paths naturally prefers reliable routes over vulnerable ones. For example, if there is a fully protected channel available, then there is no need to use any other channel (and hence PSMT by single-path transmission is doable). Conversely, if all paths are equally vulnerable, then optimal risk diversion means equiprobable transmission of shares over all available paths. Given different and individual node vulnerabilities, the optimum lies somewhere in between, and Algorithm 1 identifies the most promising paths based on known (or computed) node vulnerabilities.

### B. Approximating the Adversary

Unfortunately, we cannot use the same approach as for the path enumeration to identify the adversary's most likely targets in the same blow. It is indeed true that the adversary, knowing that only the paths $\pi_1, \ldots, \pi_k$ are used, has no incentive to attack elsewhere than on the set $\bigcup_{i=1}^{k} V(\pi_i)$, since no other node contributes to any transmission. Moreover, any hitting set for the family $\{V(\pi_1), \ldots, V(\pi_k)\}$ is a trivial cut for this path set, but hitting sets are infeasible to compute. Without question, the most valuable target for an attack *is* a minimal cut, however its general ambiguity demands care.

Figure 3 displays a network in which a minimal cut derived from the information of the previous execution to get the node-disjoint paths misleads us to a belief in a suboptimal attack strategy. This minimal cut, even if it is taken as the most vulnerable one, would be along the path $s \to 2 \to 5 \to 8 \to r$ and is $C = \{2\}$ (shown gray), since it has the likelihood of $\rho_2 = 1 - 0.3 = 0.7$ to withstand an attack, as opposed to the

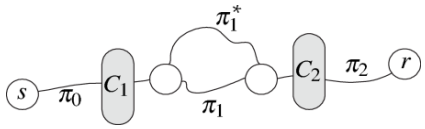



Fig. 2 Alternative routes and bounding cuts

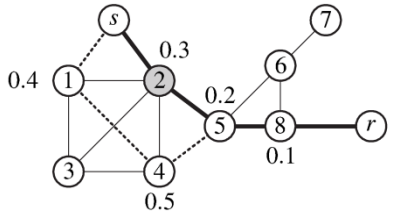



Fig. 3 Transmission routes and cuts

alternative cut at node 5, whose attack resilience is $\rho_5 = 1 - 0.2 = 0.8$. However, the adversary would surely not attack node 2 only, since this leaves the alternative (dotted) route $s \to 1 \to 4 \to 5 \to 8 \to r$ intact.

The reason for the failure of such simple vulnerability analysis by computing cuts lies in its ignorance of local alternative routes. For instance, the route $s \to 1 \to 4 \to 5 \to 8 \to r$, where the local detour is shown dashed (cf. Figure 3), may be less reliable (i.e. less resilient against an attack), however, it can indeed enforce the adversary to attack elsewhere than in node 2, since only part of the payload is delivered over this boldly shown channel. This is yet another reason why Algorithm 1 needs to (recursively) compute local detours for each path.

If the adversary structure $\mathcal{A}$ is partially or entirely unknown, we can approximate $\mathcal{A}$ by a set $\hat{\mathcal{A}}$ for the game-theoretic model by seeking the most vulnerable points in the network. The adversary's most promising target is undoubtedly a minimal cut, since any such cut $C$ has the property that every $s$-$r$-path must intersect with $C$. So there is no point in attacking elsewhere. However, cuts are notoriously non-unique and care is needed when we attempt to take certain possible attack scenarios off the radar when constructing $\hat{\mathcal{A}}$. In analogy to the previous section, we will again use a min-cut-max-flow technique to narrow down the action space for the attacker, however, with two modifications:

*1) We restrict the graph to contain only the total set of candidate paths for transmission (since attacking elsewhere is pointless).*

*2) We seek a cut $C^*$ of maximal vulnerability. Since the node weight $\rho_v$ denotes the likelihood (risk) of a successful attack on the node $v$, it is straightforward to replace the weights with its respective negative logarithmic values so that the weight of the minimal cut equals the smallest likelihood of repelling an attack.*

To make especially the first point precise assume that Algorithm 1 has led to a path ground set $P_1$ from which $\widehat{W}_G$ has been constructed. The respective ground set $P_2$ from which we

construct $\hat{\mathcal{A}}$ is then simply the union of all nodes that are used by at least one path in $P_1$, i.e., $P_2 := \bigcup_{\pi \in P_1} V(\pi)$, and restrict the graph to contain only the nodes in $P_2$ when we consider the adversary's attack strategies by seeking a cut $C^*$ whose weight (as determined by the negative logarithms of the node weights) is maximal. The goodness of the approximation of the unknown adversary structure $\mathcal{A}$ through the set $\hat{\mathcal{A}}$ is readily established, since

$$\Pr[\text{attack on } C^*] = \prod_{v \in C^*} \Pr[\text{attack on } v] \geq \prod_{v \in C} \Pr[\text{attack on } v] = \Pr[\text{attack on } C].$$

for every $s$-$r$-cut $C$. Hence, the adversary is best off attacking nodes in $C^*$, as these form the most vulnerable points in the system. The algorithmic details are unchanged, since the max-flow algorithm directly provides us with the sought cut as the most vulnerable point in the network. This is where we can expect an attack with maximal probability, so that $\hat{\mathcal{A}}$ can be constructed as the family of subsets of $C^*$. Specifically, if a set $Y \in \mathcal{A}$ shall be compromised, then either $X$ is already a cut, in which case $\hat{\mathcal{A}}$ contains a set of larger weight, i.e., better chance to fail under an attack, or $X$ is not a cut, in which case its chances to breach the security of the transmission is less than for any cut, especially those contained in $\hat{\mathcal{A}}$. We conclude these observations as

**Proposition 2.** (*Approximation of the adversary* $\mathcal{A}$) Let $Y \in \mathcal{A}$ be a set in the unknown adversary structure $\mathcal{A}$. Then the likelihood to attack the nodes in $X$ is no larger than the likelihood to attack some set in $\hat{\mathcal{A}}$.

## V. Hierarchial Networks

Many practical networks are organized in a hierarchical manner, such as company networks can be scattered throughout a country with local area networks (LANs) that are interconnected subnets of a larger wide area network (WAN). For instance, if the sub-networks are hosted by some provider, then we can model the provider's network topology only to the extent to which it is known. If so, then we can use the techniques here to get a risk estimate $\rho_G$ for the provider's network $G$. Otherwise, we can (subjectively or based on service level agreements) assign a trust level $\rho_G$ to the provider‘s network and treat it as a black box for the overall risk analysis. As another application scenario, think of a large enterprise network, in which we divide the whole network into local subnets (e.g. designated core-switches for different departments within the company) that are part of the larger network. This is modeled as the „WAN“ although it basically is a condensed view on a big LAN.

Suppose that we have a WAN $H(V, E)$, in which each subgraph node $v \in V$ itself represents another LAN subnet $G_v(V_v, E_v)$, and each edge $e = \{g_v, g_w\} \in E$ connects two “border-gateways” (these are the entry- and exit-points to the subnet) $g_v \in V(G_v), g_w \in V(G_w)$. Figure 4 shows an example. The analysis of the WAN and its subnets is based on the following two intuitions:

*1) From the WAN perspective, each subnet is represented as a single node, whose duty is only the delivery of payload through it. For that matter, we must assume that a subnet is a connected sub-graph, for otherwise, the WAN would contain routes that are physically impossible by the subnet topology. Depending on the particular internal structure of the subnet, we must do an individual and specific vulnerability analysis for the subnet and carry over this information as node-weight to the WAN for the higher-level vulnerability analysis within the WAN. All of these assessments are done using multipath transmission games in the way as described in the previous sections.*

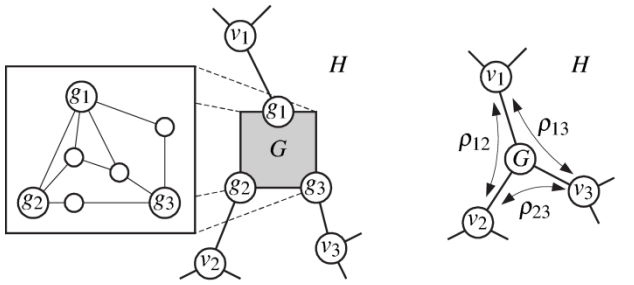
*2) Each subnet $G$ delivers its payload using MPT, exactly as the WAN does. Let $g_1, \dots, g_n \in V(G)$ be the border-gateways within $G$, then we set up and solve an MPT game (yielding the equilibrium $\mu^*, \nu^*$) and get a vulnerability estimate $\rho_{ij} = 1 - Adv_{MPT}^{\mu^*,\nu^*}(i,j)$ for the connection $g_i$-$g_j$ for $1 \leq i, j \leq n, i \neq j$. Within the super-graph $H$, the representation of the subnet is a single node with a weight-vector, and where the particular node-weight to set up the game-matrix is determined according to the exact entry- and exit-gateway when traversing the subnet $G$. If the transmission game $Game_{MPT}^{\mu^*,\nu}(G, i, j, \Pi)$ is played using the optimal choice rule $\mu^*$, then the value $\rho_{ij}$ upper bounds the probability for a successful attack on the “node” $G$ by the properties of the equilibrium distribution $\mu^*$ (see (1)). It can therefore be used to analyze the WAN in the final step, by treating the game as non-deterministic and taking the weights $\rho_{ij}$ in the WAN.*

To exemplify the treatment of subnets, especially the second of the above intuitions, consider a snippet of a WAN $H$, showing a single subnet $G$ that is connected to three other nodes in $H$. Figure 4a sketches the full network. The condensed network $H'$, shown in Figure 4b, has $G$ reduced to a single node with a vector of three weights that represent the vulnerabilities for the channels $g_1 - g_2$, $g_1 - g_3$ and $g_2 - g_3$ through the subnet $G$. Now, consider the multipath game within $H$, and a strategy that takes the (sub)route $v_2 \to G \to v_1$, then within this route, the node $G$ would receive the weight $\rho_{12}$, since from $v_2$, the payload enters $G$ through $g_2$ and leaves towards $v_1$ via the exit-gateway $g_1$. Similarly, in the path $v_1 \to G \to v_3$, the “node” $G$ would have weight $\rho_{13}$ for the same reason. This simple trick extends our multipath transmission game setup to very large scale networks at a computationally efficient level.

## VI. Experimental Evaluation

For a practical evaluation, we implemented the described algorithms in a C++ prototype, and fed a total of 210 random networks with scale-free topology (sampled under the Barabási-Albert model [38]) with node counts ranging from 20 to 150 (in steps of 10).

For outlier robustness, we computed the median running time in seconds for 15 random testcases per network size. Figure 5 displays the growth of the running time dependent on the network size. All benchmarks were carried out on an Intel Core i7 3.4 GHz with four physical cores and four virtual cores (through hyper threading) with 8 GB of RAM and Windows 7 x64 installed.




a) Network $H$ with Subnet $G$ b) Condensed Ne

Fig. 4 Hierarchial Network Transformation

Since the actual running time is strongly dependent on the network topology, we give an empirically and statistically justified estimate of the time-complexity of our method. Calling $n$ the network size and $T(n)$ the median running time, we fitted a linear model of the form $\log_2 T(n) = \lambda \cdot \log_2 n + \varepsilon$ with residues $\varepsilon$ having a Gaussian distribution with zero mean (the exact parameters were of no interest, as the Gaussian distribution was assumed only for theoretical simplicity). The value of $\lambda \approx 3.402$ was obtained using standard techniques of linear regression, and the residue dataset $r_i$ for the $i$-th testcase was tested for a Gaussian distribution using an Anderson-Darling test in the statistical software suite `R` (`www.r-project.org`). This null-hypothesis of Gaussian residues with zero mean was accepted by the test with a $p$-value of 0.5069 at a significance level of $\alpha = 5\%$. In addition, the Pearson-correlation coefficient came to $\rho = 0.98$, thus further substantiating the linear correlation between $\log T(n)$ and $\log n$ empirically. This confirms the expected polynomial relationship between $T(n)$ and $n$, of roughly the form $T(n) \in O(n^{3.4})$ for the median calculation time in seconds for a network with $n$ nodes. Considering the problem and graph algorithms in charge, this growth is unfortunately not surprising. On the bright side, it turned out in the experiments that the analysis was rather fast for networks with up to 100 nodes. Consequently, the analysis remains efficient for hierarchically structured networks. For instance, given a network with 100 nodes, each of which is a subset with 100 internal nodes and an average connectivity of, say 10, this makes $\binom{10}{2} = 45$ *independent* simulations per subnet, and a total of 4500 simulations for all subsets, plus one final simulation for the WAN. Taking the median experimental running time $T(100) \approx 83.77$ seconds as representative, we would expect a running time of approximately 10.47 hours for a network with 10,000 nodes. Considering the obvious potential of parallelizing this process within a cloud (easy since all simulations whether in the same or in different subsets are entirely *independent*), the analysis of large-scale networks is feasible with nowadays available computing power.

## VII. Conclusion

Our results indicate that multipath transmission is indeed doable and feasible in a network with many nodes, provided that some of the "bottleneck" nodes (cuts) can be secured by organizational or non-cryptographic means.

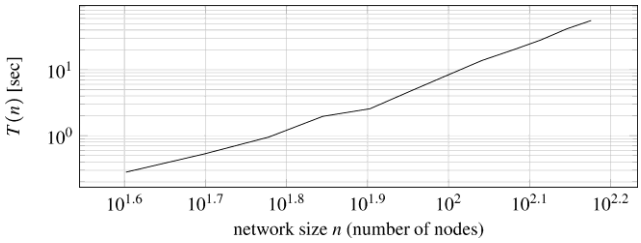



Fig. 5 Analysis times dependent on network size

We showed how to practically set up the (otherwise abstract theoretical) transmission game that models the security of a multipath transmission via the attacker's advantage in breaking the security. Using analysis techniques of game-theory, this gives a quantitative communication risk measure that can soundly be defined for more than just one security and adversary model. At the same time, it comes at serious computational cost, which can be relieved substantially by using heuristics and exploiting the network topology. Our proposed techniques require no change to existing implementations of max-flow or shortest-path algorithms, and therefore impose only little overhead in the implementation. As a by-product, we gain transmission reliability by choosing the most stable paths and as well identify neuralgic points in the network by searching for the most vulnerable cut. All of this remains feasible even for very large networks, thanks to the efficiency of the known min-cut-max-flow algorithms. In a companion paper to this work, we will report on a practical implementation of the scheme in real networks.

## Acknowledgment

This work has been supported by the Austrian Research Promotion Agency, under research grants no. 836287 and 829570.

## References

[1] J. A. Garay and R. Ostrovsky, "Almost-everywhere secure com-putation," in Proc. of EUROCRYPT ,pp.307–323, Springer, 2008.

[2] A. Patra, A. Choudhury, B. V. Ashwinkumar, K. Srinathan, and C. P. Rangan, "Perfectly secure message transmission tolerat-ing mixed adversary." IACR Cryptology ePrint Archive Report 2008/232, 2008.

[3] A. Poppe, M. Peev, and O. Maurhart, "Outline of the SECOQC Quantum-Key-Distribution network in Vienna," Int. J. of Quantum Information, vol. 6, no. 2, pp. 209–218, 2008.

[4] J. Buchmannand, J. Ding (eds.), „Post-QuantumCryptography,", Springer, LNCS 5299, 2008, Proc. of PQCrypto.

[5] M. Franklin and R. Wright, "Secure communication in minimal connectivity models," J. of Cryptology, vol. 13, no. 1, pp. 9–30, 2000.

[6] Y. Wang and Y. Desmedt, "Perfectly secure message transmission revisited," IEEE Trans. On Inf. Theory, vol. 54, no. 6, pp. 2582–2595, 2008.

[7] S. Agarwal, R. Cramer, and R. de Haan, "Asymptotically optimal two-round perfectly secure message transmission," in Proc. of CRYPTO, LNCS 4117, pp. 394–408, Springer, 2006.

[8] M. Fitzi, M. K. Franklin, J. Garay, and S. H. Vardhan, "Towards optimal and efficient perfectly secure message transmission," in Proc. of TCC (S. Vadhan, ed.), LNCS 4392, pp. 311–322, Springer, 2007.

[9] S. Rass and P. Schartner, "A unified framework for the analysis of availability, reliability and security, with applications to quantum networks," IEEE Tran. on Systems, Man, and Cybernetics – Part C, vol. 41, no. 1, pp. 107–119, 2011.

[10] M. Franklin and M. Yung, "Secure hypergraphs: privacy from partial broadcast," in Proc. of STOC, pp. 36–44, ACM, 1995.

[11] K. Srinathan, A. Narayanan, and C. Pandu Rangan, "Optimal perfectly secure message transmission," in Proc. of CRYPTO, LNCS 3152, pp. 545–561, Springer, 2004.

[12] K. KurosawaandK.Suzuki,"Almost secure (1-round, n-channel) message transmission scheme," in Proc. of ICTIS (Y. Desmedt, ed.), pp. 99–112, Springer, 2007.

[13] K. Kurosawa and K. Suzuki, "Truly efficient 2-round perfectly secure message transmission scheme," in Proc. of EUROCRYPT, LNCS 4965, pp. 324–340, Springer, 2008.

[14] K. Kurosawa, "Round-efficient perfectly secure message trans-mission scheme against general adversary," Designs, Codes and Cryptography, vol. 63, no. 2, pp. 199–207, 2012.

[15] H. M. Sayeed and H. Abu-Amara, "Efficient perfectly secure message transmission in synchronous networks," Information and Computation, vol. 126, no. 1, pp. 53–61, 1996.

[16] A. Choudhury and A. Patra, "On the communication complexity of reliable and secure message transmission in asynchronous networks," in Proc. of ICISC (H. Kim, ed.), LNCS 7259, pp. 450– 466, Springer, 2011.

[17] H. Shi, S. Jiang, R. Safavi-Naini, and M. Tuhin, "On optimal secure message transmission by public discussion," IEEE Trans. on Inf. Theory, vol.57,pp.572–585,jan.2011.

[18] J. Garay, C. Givens, and R. Ostrovsky, "Secure message trans-mission by public discussion: a brief survey," in Proc. of IWCC, pp. 126–141, Springer, 2011.

[19] P. Kotzanikolaou, R. Mavropodi, and C. Douligeris, "Secure multipath routingfor mobile adhoc networks," in Proc. ofWONS, pp. 89–96, IEEE, 2005.

[20] Z. Li and Y.-K. Kwok, "A new multipath routing approach to enhancing TCP security in ad hoc wireless networks," in Proc. of ICPP Workshops, pp. 372–379, IEEE, 2005.

[21] T. Zinner, K. Tutschku, A. Nakao, and P. Tran-Gia, "Performance Evaluation of Packet Re-ordering on Concurrent Multipath Transmissions for Transport Virtualization," Int. J. of Communication Networks and Distributed Systems, May 2010.

[22] L. Zhao and J. Delgado-Frias, "Multipath routing based secure data transmission in ad hoc networks," Prof. of IWCMC, pp. 17– 23, 2006.

[23] N. Enomoto, H. Shimonishi, J. Higuchi, T. Yoshikawa, and A. Iwata, "High-speed, short-latency multipath ethernet transport for interconnections," in Proc. of HOTI, (Washington, DC, USA), pp. 75–84, IEEE, 2008.

[24] Department of Computer Science, Network Research Group, "Multipath tcp resources." http://nrg.cs.ucl.ac.uk/mptcp/, May 2012.

[25] T. Dreibholz, "Evaluation and optimisation of multipath transport using the stream control transmission protocol." http://www.nbn-resolving.de/urn:nbn:de:hbz: 464-20120315-103208-1, March 2012. Habilitation Treatise, University of Duisburg-Essen.

[26] V. Singh, S. Ahsan, and J. Ott, "MPRTP: Multipath considerations for real-time media," ACM Multimedia Systems Conference, pp. 190–201, 2013.

[27] Y. Cui, X. Wang, H. Wang, G. Pan, and Y. Wang, "FMTCP: a fountain code-based multipath transmission controlprotocol," in Proc. of ICDCS, pp. 366–375, IEEE, 2012.

[28] A. Jøsang and R. Ismail, "The beta reputation system," in Proc. of the 15th Bled Electronic Commerce Conf., 2002.

[29] S. Rass and S. Kurowski,"On bayesian trust and risk forecasting for compound systems," in Proceedings Conf. on IT Security Incident Management & IT Forensics (IMF), pp. 69–82, IEEE Computer Society, 2013.

[30] M. Ashwin Kumar, P. R. Goundan, K. Srinathan, and C. Pandu Rangan, "On perfectly secure communication over arbitrary networks," in Proc. of PODC, pp. 193–202, ACM, 2002.

[31] S. Rass and S. Konig, "Indirect eavesdropping in quantum networks," in Proc. of ICQNM,pp.83–88, Expert Publishing Services (XPS), 2011.

[32] T. Alpcan and T. Başar, "Network Security: A Decision and Game Theoretic Approach," Cambridge University Press, 2010.

[33] S. Rass, "On game-theoretic network security provisioning," Springer J. of Network and Systems Management, vol. 21, no. 1, pp. 47–64, 2013.

[34] G. Chartrandand P. Zhang, "Introduction to Graph Theory," Higher education, Boston: McGraw-Hill, 2005.

[35] A. Abbas, "A hybrid protocol for identification of a maximal set of node disjoint paths," Int. Arab J. Of Information Technology (IAJIT), vol. 6, no. 4, pp. 344–358, 2009.

[36] S. Rass and S. Konig, "Turning quantum cryptography against itself: How to avoid indirect eavesdropping in quantum networks by passive and active adversaries," Int. J. on Advances in Systems and Measurements, vol. 5, no. 1 & 2, pp. 22–33, 2012.

[37] L. Lovász and M. Plummer, "Matching Theory," Amsterdam-New York: North-Holland, 1986.

[38] A.-L. Barabási, R. Albert, and H. Jeong, "Scale-freecharacteristics of random networks: the topology of the world-wide web," Physica A, vol. 281, no. 1–4, pp. 69–77, 2000.