Complexity of Network Design for Private Communication and the P-vs-NP Question

Stefan Rass

Alpen-Adria University, System Security Group, Department of Applied Informatics,
Universitaetsstrasse 65-67, 9020 Klagenfurt, Austria
stefan.rass@aau.at

***Abstract*—We investigate infeasibility issues arising along network design for information-theoretically secure cryptography. In particular, we consider the problem of communication in perfect privacy and formally relate it to graph augmentation problems and the P-vs-NP-question. Based on a game-theoretic privacy measure, we consider two optimization problems related to secure infrastructure design with constraints on computational efforts and limited budget to build a transmission network. It turns out that information-theoretic security, although not drawing its strength from computational infeasibility, still can run into complexity-theoretic difficulties at the stage of physical network design. Even worse, if we measure (quantify) secrecy by the probability of information-leakage, we can prove that approximations of a network design towards maximal security are computationally equivalent to the exact solutions to the same problem, both of which are again equivalent to asserting that P = NP. In other words, the death of public-key cryptosystems upon P = NP may become the birth of feasible network design algorithms towards information-theoretically confidential communication.**

***Index Terms*—Complexity; NP; Privacy; Security; Game Theory; Graph Augmentation**

I. Introduction

Encryption is a standard mean to establish private communication channels. Mostly, security rests on intractability assumptions (as for public-key cryptography) or empirical investigations (as for many symmetric encryptions). This intractability-based paradigm is opposed by techniques that use properly designed communication infrastructures to provide confidential data transmission channels. Notable examples of the latter are quantum key distribution (QKD) [1], [2] or multipath transmission (MPT) [3], [4], [5], [6], [7]. Contrary to conventional cryptography, these techniques do not hinge on computational intractability, whose related assumptions may become invalidated by increasing computational power, novel computer architectures (such as quantum- or DNA-computing [8], [9]), or new scientific discoveries (e.g., if P = NP, then most public-key cryptography is essentially insecure). Such resilience is the main motivation to look at quantum- or MPT techniques. However, the price for independency on intractability is often expensive infrastructure design, whose complexity-theoretic quantification is our goal in this work. Specifically, we investigate the (in)tractability of network graph design for the sake of running secure multipath transmission (which QKD also requires to achieve end-to-end security from point-to-point unless quantum repeaters become reality [10]).

A. Related Work and Contribution

In the quantum cryptography area, the problem network topology design to optimally support QKD has received attention in [11], [12], [13], [14]. Such considerations are justified and substantiated by previous findings [3] that multipath transmission is actually a *necessity* for confidential conversation (cf. theorem II.4) in the absence of classical cryptography or special-purpose channels (say quantum or wire-tap [15]). On the pure classical road, [4], [5] have identified graph connectivity as a necessary and/or sufficient criterion for secure communication. Related protocols like [6] then simply *presume* multiple paths to be available in a network infrastructure; a luxury that hardly any real-life network will provide. More importantly, most of the prior literature on MPT neglects complexity issues that arise in the necessary network construction. That gap motivated this work, as it poses the question for the minimal extension to a given graph to permit MPT in the sense as [6], [5], [7] and others attempt it. References [12], [13], [14] studied and classified the problem as at least NP-hard, which in turn motivates our search for *approximations* rather than exact solutions.

The contribution of this article is the unfortunate observation that even finding an approximate network design is already equivalent to proving that P = NP. While the problem of whether one can build a secure cryptosystem on the assumption that P $\neq$ NP is still unanswered ([16] provides an interesting discussion, unfortunately leaving the initial question essentially open), the confidence in the strength of nowadays public-key encryption seems well justified, based on the evidence at hand. Still, the work of [17] presents evidence *against* the well-established conjecture that one-way permutations (based on computational intractability) alone would suffice to set up a secret key agreement. We approach the same problem here via graph-connectivity based techniques (i.e., multipath transmission).

Hence, insofar secure communication avoids intractability by switching from encryption to multipath transmission based techniques (which also covers some implementations of quantum networks), intractability arises again, yet only in a different form. The good news, detailed in the concluding section, is nevertheless the observation that for secure communication, we can safely use encryption if we assume P $\neq$ NP, or otherwise construct network infrastructures for perfectly secure multipath transmission, which is feasible if ultimately P $=$ NP is proven.

B. Organization

In order to make this work as self-contained as possible, we use Section II to introduce the notation, network, adversary and security models. Subsection II-D sketches the general approach to private communication by MPT, upon which the game-theoretic privacy measure is defined in Section II-E. The network design problems are stated in Section III, with the analysis and main results following in Section IV.

II. Models and Notation

Vectors are printed as bold-face letters, complexity classes are written in small caps, sets are denoted by upper-case letters, matrices are upper-case bold-printed. For a discrete set X, we write $|X|$ for its cardinality. Whenever x is a string representation (encoding) of a problem, we write $|x|$ to denote its length, and whenever x is a real variable, then $|x|$ is its absolute value. The distinction will always be clear from the context. The symbol $\text{poly}(n)$ denotes an existing yet not further specified polynomial in the given variable (or expression) n.

A. Network Model

Let the network infrastructure consist of a set of V *devices*, and a set $E \subseteq V \times V$ of (bidirectional) communication channels between these devices. Without loss of generality, we can assume that channels cannot be attacked, because a vulnerable channel $u - v$ can be emulated by adding an intermediate vulnerable device w and inserting the two (invincible) channels u–w and w–v to the network model. Our representation for a network infrastructure is thus an undirected graph $G(V, E)$, where V is the set of nodes (devices) and E is the set of edges (point-to-point connections).

Let s, t be two distinct nodes in the graph G. An *s–t-path* π in G is a set of consecutive vertices starting at s and ending in t. We denote the set of vertices in π as $V(\pi)$. Two s–t-paths π_1, π_2 are said to be *node-disjoint*, if their only common points are s, t, i.e. if $V(\pi_1) \cap V(\pi_2) = \{s, t\}$. The *$s$–$t$-vertex connectivity* of G is the cardinality of the smallest set of nodes whose removal renders s unreachable from t in G. The *vertex connectivity* of G is the size of the smallest set of nodes such that after deletion, the graph becomes either disconnected or trivial [18]. We write $G(V \setminus U, E)$ to denote the subgraph induced by $V \setminus U$ and the remaining edges in E. We say that a graph is *k-connected*, if its vertex connectivity is k. The vertex-connectivity number is directly linked to the existence of node-disjoint paths:

Theorem II.1 ([18, Thm.5.17])**.** *A nontrivial graph $G(V, E)$ is k-connected for some integer $k \geq 2$ if and only if for each pair $s, t \in V$ of distinct nodes, there are at least k node-disjoint s–t-paths in G.*

This justifies calling a graph *biconnected* if it is 2-connected, or as equivalently used in [19], G cannot be disconnected by removing a single vertex.

B. Adversary Model

In many practical environments, flaws in some security system might concern a whole set of devices rather than only a single machine (e.g. exploits found in the firmware of a particular router might apply to a set of routers throughout the infrastructure, or also a buffer-overflow exploit in the operating system (OS) might apply to many machine running on the same OS in the same version). As we are after perfectly private communication, we must not assume any bound on the adversary's computational capabilities. Following the common practice in information-theoretic security, we model computationally unbounded adversaries via *monotonous adversary structures*.

Motivated by the above considerations, we represent an adversary $\mathcal{A}$ by a family of subsets $\mathcal{A} \subset \mathcal{P}(V)$, where $\mathcal{P}(V)$ denotes the power-set of V. Such sets within $\mathcal{A}$ may, for example, be characterized by common vulnerabilities. The family $\mathcal{A}$ thus is a collection of potentially compromised sets of devices within the network, each of which represents another possible attack scenario. The set $\mathcal{A}$ is called an *adversary structure*.

We call $\mathcal{A}$ *monotonous* if $Y \in \mathcal{A}$ implies $Z \in \mathcal{A}$ for any $Z \subseteq Y$. This captures the adversary's option to compromise less than the maximal number of nodes, or equivalently, covers situations in which not all of the adversary's servant nodes deliver useful information. A *threshold adversary* is a special case of a monotonous structure, in which all entries have equal cardinality k. Taking a *fixed* such threshold k, the structure has to no more than $|\mathcal{A}| = \binom{|V|}{k} \in O(|V|^k) = \text{poly}(|V|)$ elements, hence is polynomial. On the contrary, assuming that the adversary can conquer up to, say any fraction of $\lceil p \cdot |V| \rceil$ nodes for $0 < p < 1$ makes $|\mathcal{A}| = \binom{|V|}{\lceil p|V| \rceil} = 2^{O(|V| \log |V|)}$, which is exponential. In the following, we will exclusively deal with *polynomial size monotonous adversary structures*.

It should be noted that a threshold adversary might not always be an appropriate model. As [3] points out, the assumed threshold might yield a gross overestimation of the required graph connectivity, hence working with the more general concept of a monotonous structure adds flexibility. The work of [4] is an explicit account for minimal connectivity models, which partially helps to mitigate this issue. With the aid of game-theory, we can further generalize these previous views on perfectly private communication from a discrete yes/no-classification towards a continuous quantitative risk assessment. Details follow in Section II-E.

The physical adversary is assumed capable of capturing any set $Y \in \mathcal{A}$. Those captured nodes are entirely under the adversary's control, meaning that he is free to block, insert, modify or passively read any message passing through nodes in Y. Such an adversary is said to be *k-active*, if he can conquer any union of up to k sets from $\mathcal{A}$. Contrary to this, a *k-passive* adversary is only allowed to extract (read) information, but otherwise strictly follows the protocol without any active fiddling. Moreover, any adversary (regardless of active or passive) is assumed to know the entire protocol specification, message space, topology of the network, and any inputs except for Alice's secret message m and the coin flips r used for transmission by Alice if the protocol uses randomness (such as most cryptographic protocols do).

C. Security Model

We will use the security model put forth in [4]: at the beginning, the adversary chooses the plain text distribution $\Pr$ and the nodes to conquer from the adversary structure $\mathcal{A}$. For the actual transmission of a secret message m, the sender Alice uses a randomized protocol, taking the random coins r as an input that is *unknown* to the attacker. The adversary's *view* is the information acquired from eavesdropping on the protocol. It is denoted as $\mathcal{A}(m, r)$, whenever he extracts the message m from the information in his possession. For $\varepsilon > 0$, we say that the transmission is *ε-private*, if for every two messages $m_0 \neq m_1$ and every r, $\sum_c |\Pr[\mathcal{A}(m_0, r) = c] - \Pr[\mathcal{A}(m_1, r)]| \leq 2\varepsilon$. The probabilities are taken over the coin flips of the honest parties, and the sum is over all possible values of the adversary's view. For $\delta > 0$, we call the protocol δ-reliable, if with probability at least $1 - \delta$, Bob terminates the protocol with the correct result m. The probability is over the choices of m and the coin flips of all internal transmission nodes in V and the adversary. We call a protocol *(ε, δ)-secure*, if it is ε-private and δ-reliable. It is said to be *efficient*, when the round complexity and bit complexity are both polynomial in the size of the network, $\log \frac{1}{\varepsilon}$ and $\log \frac{1}{\delta}$ if $\varepsilon > 0, \delta > 0$. Any $(0, 0)$-secure protocol is called *perfectly secure*, and a communication having this performance guarantee is called *perfectly secure message transmission (PSMT)*. In this work, we will consider a slightly weaker notion, which we will call *arbitrarily secure message transmission (ASMT)*.

Definition II.2 (arbitrarily secure message transmission)**.** *A communication protocol is called* arbitrarily secure*, if for any (small) $\varepsilon > 0, \delta > 0$, we can efficiently run it in a way that achieves efficient (ε, δ)-security.*

Remark II.3. *Note the kind of "duality" between intractability-based and information-theoretic security: for computational (intractability-based) security, we must assume limited computational power of the adversary, while allowing the attacker to listen to all conversation over the channel. Likewise, information-theoretic security imposes no limits on the computational power, yet must assume that not the entirety of the conversation can be eavesdropped. The latter limitation will manifest itself as a polynomial bound on the cardinality of the adversary structure (permitting infinite computational power for the analysis of whatever information the attacker acquires).*

Graph connectivity has been used in [4] with the aim of judging various network types for their suitability for perfectly secure message transmission in the sense of the above security models. An interesting classification that serves as partial motivation here too has been given by [3]. Their characterization relies on a refined graph-connectivity criterion, which explicitly refers to a given adversary structure $\mathcal{A}$. More precisely, the graph G is called *$\mathcal{A}^{(k)}(s, t)$-subconnected*, if for any k sets $Y_1, \ldots, Y_k \in \mathcal{A}$ the deletion of the nodes in $\bigcup_{l=1}^{k} Y_l$ from G does not disconnect s and t within G. A graph G is said to be *$\mathcal{A}^{(k)}$-connected*, if it is $\mathcal{A}^{(k)}(s, t)$-connected for all pairs $s, t \in V$ where $s \neq t$. With this, we have the following security criterion, referring to perfect secure communication in the above sense.

Theorem II.4 ([3])**.** *Perfectly secure message transmission from the sender s to the receiver t in the network G is possible, if and only if G is $\mathcal{A}^{(2)}(s, t)$-subconnected.*

So, it suffices to consider a 2-active adversary in order to decide whether or not PSMT is possible in the given graph. This approach can indeed be improved to better match a real-life setting, using the concepts of *channel-* and *network-vulnerability* [20], which we briefly recap in section II-E later. The next section is devoted to a closer look at the ideas of how to achieve perfectly secure communication within Theorem II.4 and related results (e.g. [6], [5]).

D. Transmission Model

The general idea underlying all (secure) multipath transmissions schemes between a sender s and receiver (target) t is the following: the sender s chooses a set P of node-disjoint s–t-paths, and encodes the message m into n packets. Let the entirety of nodes that are used to convey m be denoted as $V(P) = \bigcup_{\pi \in P} V(\pi)$. The attacker takes over a set $Y \in \mathcal{A}$ of nodes in an attempt to learn everything that flows through the nodes in $V(P) \cap Y$. The sender performs the transmission by encoding m into $|P|$ pieces $c_1, \ldots, c_{|P|}$, and sending those to t over their own individual paths in P. In the simplest case, this can be done by conventional XOR-secret-sharing, i.e. $m = c_1 \oplus c_2 \oplus \cdots \oplus c_{|P|}$, where $\oplus$ is the bitwise XOR, and all but one of the c_i's are random strings. The message is protected from discovery unless the attacker intercepts all paths in P. Since such encoding is prone to transmission errors and blows up the overall transmission overhead, practical schemes [6], [5] employ more flexible and efficient encodings (e.g., based on polynomial secret sharing to add error correction capabilities and thus gain robustness)[1].

Perfectly secure message transmission demands some encoding and transmission paths P such that *every* attack scenario $Y \in \mathcal{A}$ gives insufficient information to recover m. For example, the above XOR-secret-sharing over $n = |P|$ paths displays a one-round PSMT scheme against an attacker with $|Y| < n$ for every $Y \in \mathcal{A}$ (see figure 1; and note that the case $n = 2$ is essentially equivalent to symmetric encryption).

Towards the weaker goal of arbitrarily secure message transmission, we can use randomly chosen (and changing)

[1] Feedback rounds can as well be used to gain efficiency and security [7], however, we confine ourselves to one-round protocols here.

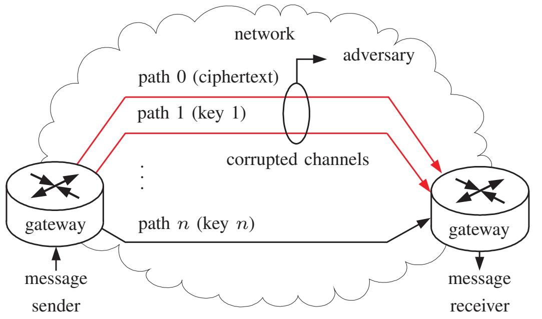


Fig. 1. Basic approach to perfectly secure message transmission

paths to deliver the packets $c_1, c_2, \ldots, c_n$, in an attempt to minimize the attacker's chances to learn enough information to discover m. Like for PSMT, we attempt to bypass the attacker, however unlike in PSMT, the randomly chosen paths are not fixed a-priori, thus making ASMT possible even in some cases where the attacker (e.g., thanks to a sufficient threshold) could break the respective PSMT scheme. Moreover, ASMT is doable even using (a sequence of) single-path transmissions, which cannot be used to run PSMT.

E. Channel- and Network-Vulnerability

Security of multipath transmission hinges on the existence of at least one path that bypasses all hostile nodes in the network. Consequently, it is the sender's (player 1) intention to optimize his path choices against an attacker (player 2) who seeks optimal spots to sniff the network traffic. This optimization can be done using game-theory.

To this end, take the collection of all existing s–t-paths, and group them together into a polynomial number of poly$(|V|)$ different bundles $P_1, P_2, \ldots$ (note that the full enumeration of paths would have exponentially many entries, hence we must work with a feasibly small selection of these). Condense all these bundles in the *strategy set* PS_1. With this set, the game is about the sender taking his best randomized choice of a path set for communication. The *opponent strategy set* PS_2 is exactly the adversary structure $\mathcal{A}$. The game's payoff matrix $\mathbf{A} = (a_{ij})$ can be defined in binary terms as

$$a_{ij} = \begin{cases} 1, & \text{if the } s\text{–}t\text{-transmission remained secret;} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

if $i \in PS_1$ is the chosen pair of paths π_1, π_2, and $j \in \mathcal{A}$ is the compromised set $Y \subset V$ of adversarial nodes within the network $G(V, E)$. We note $a_{ij} = 1$ if the compromised set was insufficient to extract the secret from the adversary's view (transcript). Note that this decision strongly depends on the chosen encoding of m, so the evaluation of equation (1) depends on the particular instantiation of the framework protocol (examples are found in [5], [6]).

The game's solution is the *saddle-point value* $v(\mathbf{A}) = \max_{\mathbf{x}\in S(PS_1)} \min_{\mathbf{y}\in S(\mathcal{A})} \mathbf{x}^T\mathbf{A}\mathbf{y}$, where $S(PS_1), S(\mathcal{A})$ denote the set of (discrete) probability distributions over the player's strategy sets. The *equilibrium* is the pair $(\mathbf{x}^*, \mathbf{y}^*) \in S(PS_1) \times S(\mathcal{A})$, at which the saddle-point value $v(\mathbf{A}) = (\mathbf{x}^*)^T\mathbf{A}\mathbf{y}^*$ is attained. The definition of $v(\mathbf{A})$ directly formalizes the aforementioned competition: the sender tries to maximize his chances of keeping the message secure (maximization over all randomized choices $\mathbf{x} \in S(PS_1)$), while the attacker tries his best to discover the message (minimization of the sender's benefit over all randomized choices $\mathbf{y} \in S(\mathcal{A})$ of nodes to conquer from $\mathcal{A}$).

Such modeling might be inaccurate in a real-life scenario because assuming a zero-sum competition can be a misjudgment of the adversary's intentions. However, as eloquently noted in [21], presuming a zero-sum regime is a valid worst-case approach, since with the binary valuation as above and with $v(\mathbf{A})$ denoting the saddle-point value of the zero-sum game induced by the matrix $\mathbf{A}$, it is easy to prove that

$$\Pr[\text{successful attack}] \leq 1 - v(\mathbf{A}),$$

which holds *regardless* of how the adversary actually behaves, provided that the sender and receiver act according to their zero-sum equilibrium strategy. Notice that the matrix $\mathbf{A}$ specifically models the communication between s and t. In [20], the upper bound $1 - v(\mathbf{A}) =: \rho(s, t)$ has been assigned the name *vulnerability*, since it measures the degree to which an attack will be successful.

Applications in which the outcome of the transmission cannot be classified in binary terms as in (1) or perhaps is even random, can arise in infrastructures that use security measures like firewalls, intrusion detection systems, etc., all of which have some positive rate of failure. A straightforward way to recover a deterministic valuation from a random outcome in a transmission scenario is taking expectations of the random outcome. This changes the game's payoff structure from a 0-1-matrix to a matrix with real values, but does no inherent change to the model nor its solution procedure. Since random or more general than binary outcomes can be treated with the very same framework, we avoid unnecessary complications here by leaving this direction aside. Respective details and examples can be found in [20], but are not needed for our upcoming considerations.

Definition II.5. *Let a graph $G(V, E)$, an integer $k \geq 1$ and a pair of distinct nodes $s, t \in V$ be given. Assume that an s–t-communication runs over k paths in the presence of an adversary (structure) $\mathcal{A}$. The* vulnerability *of this s–t-communication is defined as $\rho(s, t) = 1 - \max_{\mathbf{x}\in S(PS_1)} \min_{\mathbf{y}\in S(\mathcal{A})} \mathbf{x}^T\mathbf{A}\mathbf{y}$, where $\mathbf{A} \in \{0, 1\}^{|PS_1| \times |\mathcal{A}|}$ models the zero-sum communication game with the payoffs as defined through* (1)*.*

As not all nodes in a network might be actively communicating, it makes sense to restrict the attention to only a certain set of pairs $U \subseteq V \times V$ that will eventually attempt a private conversation. We call the entirety of these pairs a *communication relation*, whose vulnerability is our measure of overall security in the network $G(V, E)$.

Definition II.6. *For a communication relation $U \subseteq V \times V$, the network $G(V, E)$ has the* vulnerability

$$\rho(G,U) := \max_{s,t\in U} \rho(s,t). \qquad (2)$$

Convention (2) is justified by the maximum-principle that is common practice in security audits: the overall security of a system is determined by the vulnerability of its weakest component (similarly to a chain being as strong as its weakest element). In the following, we will use the following characterization of ASMT based on the vulnerability.

Theorem II.7 ([20])**.** *Let Alice and Bob set up their game matrix with binary entries $a_{ij} \in \{0,1\}$, where $a_{ij} = 1$ if and only if a message can securely and correctly be delivered by choosing the i-th pure strategy, and the adversary uses his j-th pure strategy for attacking. Then $\rho(\mathbf{A}) \in [0,1]$, and*

1) *If $\rho(\mathbf{A}) < 1$, then for any $\varepsilon > 0$ there is a protocol so that Alice and Bob can communicate with an eavesdropping probability of at most ε and a chance of at least $1-\varepsilon$ to deliver the message correctly.*
2) *If $\rho(\mathbf{A}) = 1$, then the probability of the message being extracted and possibly modified by the adversary is 1.*

F. How ASMT Relates to PSMT and Risk Management

It is worth noting that in case of a pure binary valuation, ASMT becomes PSMT if the vulnerability is either 0 or 1, in which case the incident of zero vulnerability directly implies a certain graph connectivity. We will exploit this fact later.

Moreover, Theorem II.7 remains valid under a modified setting in which the outcome of a transmission is uncertain. More specifically, while PSMT usually presumes all-or-nothing adversarial access to a node, ASMT can be used with probabilistic security models and uncertain behavior of a node's defense (e.g., a firewalls, virus scanners, etc.). The above characterization of (im)possible ASMT still holds. As a further generalization unlike PSMT, ASMT based on games permits using different scales than zero-one, especially nominal or scales used in qualitative risk management. Since the vulnerability is the expected product of likelihood and damage in terms of the given scale, it is nothing else as a *risk*. So, the security guarantees made by ASMT are much better compatible with quantitative (and under a mapping of the vulnerability onto a nominal scale, also qualitative) risk management issues. PSMT is not explicitly designed for integration in such processes. This means that the general problems stated in the next section equivalently refer to the search for a network design that minimizes (general) risk of communication in perhaps even monetary units. Unfortunately, this particular task of risk management will be proven infeasible unless P = NP.

III. Graph Augmentation for Secret Communication

Theorems II.4, II.7 as well as the results of [4] and [5] indicate that – on classical grounds, i.e., in the non-quantum setting – multiple paths are inevitable for perfectly and arbitrarily secure communication. This raises the natural question of graph augmentation in order to meet these needs. Using the aforementioned game-theoretic framework and Theorem II.7 in particular, the problem boils down to asking for an augmentation that yields a vulnerability $\rho(G,U) \leq \varepsilon < 1$ for a given network G, communication relation U and risk threshold ε. Besides the decision-version of the problem, our main interest in the following lies in the respective *search* problem. Suppose that the network is insufficiently connected so that perfectly and arbitrarily secure transmission are both ruled out by any known conventional criterion (e.g. [3], [4], [5]). Then we seek the smallest (cheapest) edge-augmentation to G that would at least give $\rho(G,U) \leq \varepsilon$, so that at least ASMT is possible, even if PSMT might still be out of reach. This is problem III.1.

Problem III.1 Min-Vulnerability-Augmentation

Instance: A graph $G(V,E)$, an adversary structure $\mathcal{A} \subset 2^V$, a set of pairs $U \subseteq V \times V$ that can communicate and a set $\widetilde{E}$ of additional (candidate) edges with costs $c : \widetilde{E} \to \mathbb{Z}^+$, and a budget limit $B \in \mathbb{Z}$.
Solution: An edge augmentation $E^+ \subseteq V \times V \setminus E$ within the budget limit $c(E^+) \leq B$.
Measure: The vulnerability $\rho(G(V, E \cup E^+), U) = \max_{(u,v)\in U} \rho(u,v)$, where $\rho(x,y)$ is the vulnerability of an x–y-communication in G

Towards formulating optimization problems, we let $\widetilde{E} \subset V \times V \setminus E$ be a set of candidate edges not yet existing in the graph $G(V,E)$. Furthermore, let a function $c : \widetilde{E} \to \mathbb{Q}^+$ measure the costs for any of these edges. For reasons of tractability (theoretical as well as computational), we assume that $c(E^+)$ can be computed in poly($|E^+|$) time by a Turing-machine that leaves an encoding of $c(E^+) = \frac{a}{b} \in \mathbb{Q}^+$ on its output tape of the form $\#a\#b\#$, where a, b are natural (radix-based) encodings of the integers a and b.

The "reverse" problem III.2, which asks for the cheapest augmentation that undercuts a given vulnerability limit, is treated later.

In the following sections, we will investigate the complexity of both problems, and discover the existence of efficient exact solution algorithms as equivalent to P = NP. Both problems are known to be NP-hard [13], but even despite this fact, there is no point in looking for approximation algorithms.

Before getting to the complexity-theoretic details, let us consider the obvious variants of the above problems; why not consider vertex-augmentations or mixed (vertex- and edge-)

Problem III.2 Min-Cost-Security

Instance: A graph $G(V,E)$, an adversary structure $\mathcal{A} \subset 2^V$, a set of pairs $U \subseteq V \times V$ that can communicate and a set $\widetilde{E}$ of additional (candidate) edges with costs $c : \widetilde{E} \to \mathbb{Z}^+$, and a vulnerability limit ε.
Solution: An edge augmentation $E^+ \subseteq V \times V \setminus E$ that achieves the vulnerability limit $\rho(G(V, E \cup E^+), U)) \leq \varepsilon$.
Measure: The total cost $c(E^+)$ of the augmentation E^+.

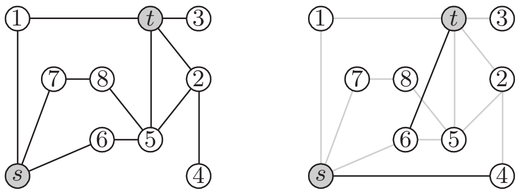
augmentations? It is easy to see that adding only vertices does no change to the vulnerability, since the nodes are all isolated. Adding vertices and edges is equivalent to adding the vertices in first place (leaving the problem's solution unchanged), and afterwards consider a pure edge-augmentation only. So, edge augmentations cover both of these cases.

Example

Problem Min-Vulnerability-Augmentation and Min-Cost-Security both admit representations as mixed-integer programming problems [22]. Therefore, solutions for small networks might be feasible in a practical setting. Moreover, the representation of either problem is trivially converted into a representation of the other, so that linear programming software (e.g. `Cplex` or `lp_solve`) can be applied to both. For example, consider the network shown in Figure 2a, being the yet unaugmented graph. We solve the respective instance of Min-Vulnerability-Augmentation for an adversary structure $\mathcal{A} = \{U \subset V : |U| = 3\}$ and two-path transmission from s to t, where the encoding of the message m is by a $(2,2)$-XOR-secret sharing of the form $m = r_1 \oplus r_2$, where r_1 is random and $r_2 = m \oplus r_1$ (one-time pad symmetric encryption under key r_1). Consequently, the transmission is perfectly private unless both, r_1 and r_2 are intercepted by the attacker. Finally, let the budget limit be $B = 18$ and take the set $\widetilde{E}$ of candidate edges along with edge weights as given by Figure 2c.

Observe that $Y_{\text{cut}} = \{1, 8, 6\} \in \mathcal{A}$ so that no communication from s–t is possible without traversing a node in Y_{cut} in the *unaugmented network* shown in Figure 2a (another cut would be $\{1, 5\}$). Consequently, a fraction of $v = 0$ messages can be delivered secretly and hence the vulnerability is $\rho = 1 - v = 1$ for the unaugmented network. Contrary to this, the fully augmented network including all edges in $\widetilde{E}$ permits 141 different s–t-paths, from which we can form a set PS_1 having 295 pairs of node-disjoint paths. The adversary has – in either case – $|PS_2| = |\mathcal{A}| = \binom{8}{3} = 56$ possible attack strategies (where attacks on s or t are excluded for obvious reasons). Setting up the full game matrix results in a (295×56)-tableau, from which we can iteratively and alternatingly delete rows and columns whose payoff is uniformly worse than for another column (in game-theory terminology, we delete the *dominated strategies*). This reduction leaves us with a 6×4 payoff matrix $\mathbf{A}$, shown in Figure 3b, along with the remaining strategies for both players, listed in Figure 3a. All other existing strategies are either redundant (i.e., yield duplicate rows or columns in the matrix) or give less or equal revenue than another strategy (i.e., are dominated). Solving the linear program (in polynomial time [23]) gives $v(\mathbf{A}) = 0.5$ at the full cost of $c(\widetilde{E}) = 22$. Our goal is finding the *minimal* augmentation obeying the cost limit of 18.

Figure 2b displays the solution $E^+ = \{t\text{–}6, 4\text{–}s\}$ for Min-Vulnerability-Augmentation, having $\rho = 0.5$ as the maximal attack probability, as opposed to $\rho = 1$ in the unaugmented graph. Seeking the minimal cost augmentation to achieve at least $\rho = 0.5$, i.e. solving Min-Cost-Security with $\varepsilon = 0.5$ gives the same solution shown in Figure 2b, coming at price $c(E^+) = 8$, and proving that the previous solution E^+ is as well the cheapest for this security demand.



(a) Original network

(b) Optimal $s - t$-channel-security augmentation

$e \in \widetilde{E}$	1–6	t–6	2–3	2–6	3–4	3–5	4–5	4–s
cost $c(e)$	1	5	4	1	4	3	1	3

(c) Edge augmentation set $\widetilde{E}$

Fig. 2. Example graph augmentation

strategy number	PS_1 (pairs of paths)		$\mathcal{A} = PS_2$ (compromised)
1	s–4–2–t	s–1–t	$1, 5, 6$
2	s–6–t	s–1–t	$1, 4, 6$
3	s–6–t	s–4–2–t	$1, 4, 5$
4	s–7–8–5–t	s–1–t	$4, 5, 6$
5	s–7–8–5–t	s–4–2–t	
6	s–7–8–5–t	s–6–t	

(a) Strategy sets

sender \ attacker	1	2	3	4
1	1	0	0	1
2	0	0	1	1
3	1	0	1	0
4	0	1	0	1
5	1	1	0	0
6	0	1	1	0

(b) Payoff matrix $\mathbf{A}$

Fig. 3. Game-theoretic model for our example

Unfortunately, any heuristic approximation to the general problem (i.e. not all equal edge costs) is doomed to unbounded relative errors, unless P = Np, as we prove in the sequel.

IV. Complexity of Graph Augmentation for ASMT

To answer the question whether or not it is feasible to create suitable networks for multipath transmission efficiently, we will use some complexity classes for search problems, besides the decision-problem classes P, Np, and the set Npc of problems that are complete for Np. The class Fp is the set of all binary relations $P(x, y)$ such that there is an algorithm A that runs in time poly($|x|$) and outputs some y such that $P(x, y)$ holds. The class Fp$^{\text{Np}}$ is defined in exactly the same way, except that A is allowed to make queries to an Np-oracle, where a call to the oracle takes only one step.

An *instance* of an optimization problem is denoted by I. By $\mathsf{A}(I)$, we denote the result of the algorithm A when applied to the instance I of the (general) optimization problem (e.g., MAX-CLIQUE). For many computationally hard problems efficient approximations are known (one example is MAX-CUT, for which an astonishingly good approximation has been found by [24]). An excellent account is given in [25], from which we will repeatedly draw in the following. Here we give our definitions only for minimization problems.

Definition IV.1. *Given an instance I of a minimization (optimization) problem, an algorithm A is called an* approximation algorithm*, if its output $\mathsf{A}(I)$ is a feasible (not necessarily optimal) solution of I. Given $r \geq 1$, we call A an r-*approximation algorithm*, if*

$$opt(I) \leq \mathsf{A}(I) \leq r \cdot opt(I), \tag{3}$$

where $opt(I)$ denotes the optimal (minimal) value of the optimization problem I.

The class APX includes all optimization problems for which a polynomial-time r-approximation algorithm exists. Strictly speaking, one would need to define APX in terms of the class NPO, which is roughly the set of all "NP-optimization problems". Since we will not need these classes any further, we refer the reader to [25] for details, and refrain from granting APX a full-fledged formal definition (which would unnecessarily complicate things here).

The next section contains a number of technical results needed to establish the main contributions in Section IV. First, we are concerned with the computational feasibility of evaluating the vulnerability of a given network.

A. Computing Vulnerabilities

Lemma IV.2. *Let $G(V, E)$ be a graph modeling a communication network, and let $\mathcal{A}$ be an adversary structure of size $|\mathcal{A}| = poly(|V|)$. Then it takes only polynomial time to decide whether or not ASMT is possible over G and if so, the respective channel- and network-vulnerabilities can be computed in polynomial time.*

Proof: Take any two arbitrary fixed and distinct vertices $s, t \in V$. Observe that, if there is a set Y such that any s–t-path π intersects Y, i.e. $V(\pi) \cap Y \neq \emptyset$, then attacking Y is a classical person-in-the-middle attack, which without pre-shared secrets between s and t, trivially rules out any private conversation between s and t (simply because t and the adversary have exactly the same information, so t cannot do anything to decrypt that the adversary could not do equally well). So, fix any ordering of $\mathcal{A} = \{Y_1, \ldots, Y_n\}$ and let us iterate over all elements in $\mathcal{A}$ (note that $|\mathcal{A}| = \text{poly}(|V|)$ and hence feasibly small to iterate over it). We will construct a game-matrix modeling a single-path transmission from s to t that attempts to circumvent the adversary as good as possible. Moreover, observe that we cannot rely on any encryption between s and t, since no (shared) keys are available (public-key cryptography is ruled out by our demand for perfect secrecy).

Each set $Y_j \in \mathcal{A}$ makes yet another attack strategy, so the game-matrix A will have exactly $n = |\mathcal{A}| = \text{poly}(|V|)$ columns. We will iterate through $\mathcal{A}$ and look for a path that lets us securely communicate if the nodes in Y_j are compromised. Technically, we will choose a set of n transmission strategies such that the diagonal of the payoff matrix is composed of all 1's, which will ensure a positive saddle-point value and finally enable ASMT by Theorem II.7.

So let $Y_j \in \mathcal{A}$ be given, and look for an s-t-path that explicitly avoids using any node $v \in Y_j$. This is easily accomplished in polynomial time by running a shortest-path algorithm on a transformed version of G. The required transformation is known from the computation of maximal flows with vertex capacities and can identically be re-used to find paths that avoid certain nodes within a graph. We refer the reader to [26] for a concise representation of this trick (where it has been used for a quite different purpose though). Depending on the outcome of the shortest-path algorithm, distinguish two cases:

Case 1: There is no s–t-path without using nodes in Y_j. Then attacking Y_j will intercept any communication from s to t, and hence no private channel can be set up. In that case, ASMT is ruled out for obvious reasons. Moreover, the vulnerability of the network and the s–t-channel are both 1.

Case 2: There is a path π_j such that $V(\pi_j) \cap Y_j = \emptyset$. Then, private transmission over π_j is possible, and we can assert that $a_{jj} = 1$ in the game-matrix $\mathbf{A}$, since player 1 wins the scenario in which he uses π_j for transmission and Y_j is attacked.

In this way, we obtain a path π_j that avoids Y_j for all $j = 1, 2, \ldots, |\mathcal{A}|$, so that at least on the diagonal of the final game-matrix, we have all 1's. Computing the value of this special matrix game (i.e. a *diagonal game*) is easy, since it is known from game-theory (see [27]) that a diagonal matrix has the saddle-point value $v(\text{diag}(1, \ldots, 1)) = \frac{1}{n}$. So, even if player 1 would lose the private transmission game in all other scenarios except for the diagonal of the game-matrix, we get $v(\mathbf{A}) > 0$. Now, regardless of what the off-diagonal entries in the actual game-matrix $\mathbf{A}$ actually do, we surely have $\mathbf{A} \geq \text{diag}(1, \ldots, 1)$, where the inequality holds per component. This inequality is preserved if we take averages on either side, giving $\mathbf{x}^T\mathbf{A}\mathbf{y} \geq \mathbf{x}^T\text{diag}(1, \ldots, 1)\mathbf{y} > 0$ for all discrete probability distributions $\mathbf{x}, \mathbf{y}$. Hence, ASMT is possible by Theorem II.7.

To compute the exact value of $v(\mathbf{A})$, i.e. the s–t-channel vulnerability, observe that the matrix $\mathbf{A}$ has exactly $n^2 = |\mathcal{A}|^2$ entries. Computing the off-diagonal elements a_{ij} (with $i \neq j$) is easy because row i corresponds to a path π_i, column j corresponds to a compromised set Y_j, and the entry a_{ij} is found as

$$a_{ij} = \begin{cases} 1, & \text{if } V(\pi_i) \cap Y_j = \emptyset \\ 0, & \text{otherwise.} \end{cases}$$

The saddle-point value of the full game-matrix $\mathbf{A}$ can then be computed in polynomial time by solving a linear optimization program [23]. The overall network vulnerability can as well be computed in polynomial time, since there are no more than $O(|V|^2)$ s–t-pairs to look at. ■

As a simple corollary, the following statement assures that the vulnerability of any augmented graph and given communication relation can be computed in polynomial time.

Corollary IV.3. *Let a graph $G(V,E)$ and an adversary structure $\mathcal{A}$ over V be given. Then, for any augmentation $E' \subseteq V \times V$, and any set $U \subseteq V \times V$, the network vulnerability $\rho(G(V, E \cup E'), U)$ of the augmented graph can be calculated in polynomial time.*

The proof is immediate from the proof of Lemma IV.2, when one considers the obvious generalization of the above arguments to transmissions using more than one path and perhaps a different encoding. In any such setup, the same trick as above can be invoked provided that the payoffs can be computed in polynomial time, which is trivially possible in the settings that we consider.

Theorem II.4 classifies perfectly secure transmission in terms of network connectivity. Towards studying the hardness of graph augmentation for security, we relate the problem to graph augmentation for biconnectivity, which is known to be NP-complete in certain variants [19]. If we use two-path transmission and a special adversary structure, we can establish a useful relation between biconnectivity and network vulnerability.

Lemma IV.4. *Let a graph $G(V,E)$ be given. Put $n = |V|$ and define an adversary structure as*

$$\mathcal{A} = \{\{1\}, \{2\}, \dots, \{n\}\}. \qquad (4)$$

Then the following two statements hold for the vulnerability of G w.r.t. $\mathcal{A}$ and any sender-receiver pair $s, t \in V$ that performs two-path transmission:

1) *$\rho \in \{0, 1\}$, and*
2) *G is biconnected if and only if $\rho = 0$.*

Proof: By theorem II.1, we know that G is biconnected if and only if there are two node-disjoint paths between any two vertices in G, i.e. two disjoint channels exist for any pair in $V \times V$. Since the adversary can attack at most one node at a time, $\mathcal{A}$ cannot disconnect any pair that actually has two channels between them. Since the vulnerability is $\rho = \max_{(u,v)\in V\times V} \rho(u,v)$, and the adversary structure is such that $\rho(G,U) \in \{0,1\}$, we conclude that $\rho = 0$ if and only if the adversary can mount a person-in-the-middle attack between at least one pair in $V \times V$. Otherwise, there is at least one pair such that all paths between them run through a node in $\mathcal{A}$, and the graph has vulnerability $\rho = 1$ and is not biconnected. ■

B. On the Existence of Approximations Towards ASMT

Having prepared the groundwork, we are ready to present our main findings. Our first result rules out the existence of efficient approximations for either problem if P $\neq$ NP.

Theorem IV.5. *Unless* P $=$ NP*, there is no r-approximation algorithm for* MIN-VULNERABILITY-AUGMENTATION.

One could equivalently state that MIN-VULNERABILITY-AUGMENTATION $\in$ APX implies P $=$ NP. However, as Theorem IV.7 will later show, there is no point in looking for an approximation algorithm at all, since the existence would imply that there is as well a polynomial-time exact solution algorithm for the problem!

Proof of Theorem IV.5: Suppose there were an r-approximation algorithm A for MIN-VULNERABILITY-AUGMENTATION, and let an instance of the BICONNECTIVITY-AUGMENTATION problem be given, which is known to be NP-complete [19]. This instance is made up by a graph $G(V,E)$, a weight function $w(u,v) \in \mathbb{Z}^+$ for each unordered pair $\{u,v\}$ of nodes in V, and a positive integer B. The question is to decide whether there is a set E' of unordered pairs of vertices from V such that $\sum_{e\in E'} w(e) \leq B$ such that the graph $G(V, E \cup E')$ is biconnected, i.e. cannot be disconnected by deleting a single vertex [19].

We can easily (almost directly) cast this problem into an instance I of MIN-VULNERABILITY-AUGMENTATION as follows: set the network to be G, and use the adversary structure (4). Moreover, define $U := V \times V$, and set the additional edge weights to $w(e)$ as given by the instance of BICONNECTIVITY-AUGMENTATION for all $\widetilde{E} := (V \times V) \setminus E$. The budget limit is also taken from the given instance of BICONNECTIVITY-AUGMENTATION. Lemma IV.4 characterizes biconnectivity in terms of the adversary structure $\mathcal{A}$ and its implied vulnerability. So if we solve the MIN-VULNERABILITY-AUGMENTATION problem under the given budget constraints, Lemma IV.4 implies that G can be biconnected within the budget limit if and only if the optimum vulnerability is $\rho^* = 0$. Now, since we have an r-approximation algorithm, we conclude that

1) In case that $\mathsf{A}(I) = 0$, (3) implies $\rho^* = 0$ since $0 \leq \rho^* \leq A(G)$, and hence there is a feasible edge-augmentation to biconnect G.
2) Otherwise, if $\mathsf{A}(I) > 0$, then again by (3), $0 < \mathsf{A}(I) \leq r \cdot \rho^*$, so $\rho^* \neq 0$. Lemma IV.4(1) implies that $\rho^* = 1$, which means that there is at least one pair that can be disconnected by removing a single node, and G cannot be biconnected within the budget limit. ■

An analogous result holds for MIN-COST-SECURITY too.

Theorem IV.6. *Unless* P $=$ NP*, there is no r-approximation algorithm for solving* MIN-COST-SECURITY.

As before, one can equivalently state this by saying that MIN-COST-SECURITY $\in$ APX implies P $=$ NP. Hence, by the same token as above, looking for approximations to this problem is useless.

Proof of Theorem IV.6: Assume an r-approximation algorithm A for MIN-COST-SECURITY to be available, and let an instance of a HAMILTONIAN-CIRCUIT problem be given, which is a graph $G(V,E)$ and the question of whether it has a spanning circle. The reduction will be in two steps. We start by reducing the HAMILTONIAN-CIRCUIT to an instance of the BICONNECTIVITY-AUGMENTATION problem, by modifying the construction of [28]. Consider the biconnectivity augmentation problem on the set V, where the edge weights are set to

$$w(u,v) = \begin{cases} 1, & \text{if } (u,v) \in E; \\ 1 + rn, & \text{if } (u,v) \notin E, \end{cases}$$

and the budget limit is $n = |V|$. [28, Theorem 4] states that G has a Hamiltonian circuit if and only if there is an edge augmentation of cost less than or equal to $|V|$. Now, suppose that we apply an r-approximation algorithm for MINIMUM-COST-SECURITY to exactly this instance, with the adversary structure being (4) again. So the condition $\rho(G,U) \leq \frac{1}{2}$ enforces the approximation algorithm to look at only biconnected extensions of the network, by Lemma IV.4.

If G admits a Hamiltonian cycle, then the edge augmentation has cost $\leq n$ and our approximation algorithm returns at most $\mathsf{A}(I) \leq rn$. On the other hand, if G does not admit a Hamiltonian cycle, then the costs come back $> n$ and at least one edge with cost $1+rn$ must have been used (since G is not Hamiltonian). The minimal costs found by the approximation algorithm for MINIMUM-COST-SECURITY must therefore be at least $\mathsf{A}(I) \geq (n-1) + (1+rn) = (r+1)n > rn$. ■

Knowing that neither of the problems stated in section III admit a polynomial time r-approximation, it is interesting to notice that they indeed admit an exact solution using polynomially many queries to an NP-oracle. The proof is based on a discretization of the optimization measure function, which uses Farey-sequences, and found in [14].

Theorem IV.7. MIN-VULNERABILITY-AUGMENTATION $\in$ FP$^{\text{NP}}$

As before, the same result (yet with a different proof) holds for MIN-COST-SECURITY. This as well admits an exact solution in polynomially many steps and calls to an NP-oracle. The proof as well employs Farey-sequences and bisective searching to discretize and narrow down the search space. A different version of this result also appears in [14], however, the proof given here is new and much simpler.

Theorem IV.8. MIN-COST-SECURITY $\in$ FP$^{\text{NP}}$

Proof: Let n be the size of the given instance of MIN-COST-SECURITY. By definition, the measure function $c : V \times V \to \mathbb{Q}^+$ can be computed in polynomial time, i.e. there is a Turing-machine taking at most $p(n)$ steps to leave an encoding of $c(E) = \frac{a}{b}$ on the tape. This encoding takes the form $\#a\#b\#$, where a and b are nonnegative integers with radix encodings. Since this is printed within $p(n)$ steps, it follows that $a, b \leq 2^{q(n)}$, for some polynomial q (in fact, the polynomial q is proportional to the polynomial p, with a constant that depends on the radix for the encoding of a, b). Consider the normalized costs

$$0 \leq \frac{a}{2^{q(n)} b} \leq 1. \tag{5}$$

Since $2^{q(n)} b \leq 2^{2q(n)}$, we conclude that expression (5), as having a bounded denominator, must lie within a Farey-sequence of order $2^{2q(n)}$. Using Theorem 28 in [29], we can lower-bound the distance between any two different such fractions as $\left|\frac{a}{2^{q(n)}b} - \frac{a'}{2^{q(n)}b'}\right| \geq \frac{1}{2^{4q(n)}}$. We multiply the last inequality by $2^{q(n)}$ to obtain

$$\left|\frac{a}{b} - \frac{a'}{b'}\right| \geq 2^{-3q(n)} = 2^{-O(p(n))} \tag{6}$$

Since $a, b \leq 2^{q(n)}$, we can bound the measure value as $|c(E)| \leq 2^{O(p(n))}$. Now, we can continue as in the proof of Theorem IV.8 by running a bisective search over the interval $[0, 2^{O(p(n))}]$, which terminates as soon as the search space has shrunk below the size of $2^{-O(p(n))}$. To this end, we introduce problem IV.1 for the decision version of MIN-COST-SECURITY in the analogous way as before.

Problem IV.1 CHEAP-SECURITY

INSTANCE: the same as for MIN-COST-SECURITY, with an additional cost threshold C.

QUESTION: Is there an edge augmentation E^+ achieving a desired maximal vulnerability $\rho(G(V, E \cup E^+), U) \leq \varepsilon$ such that the cost for E^+ are limited as $c(E^+) \leq C$?

A nondeterministic Turing-machine can easily guess a solution E^+ and verify it in polynomial time, since by Lemma IV.2, the vulnerability threshold can be checked efficiently, and by definition of CHEAP-SECURITY, the measure can as well be calculated within $p(n)$ steps. It follows that CHEAP-SECURITY $\in$ NP.

For the bisective search, we make a call to a CHEAP-SECURITY-oracle (i.e. an NP-oracle) in order to decide the direction where to continue our search. The number of steps until we may terminate is, by (6), no more than $O(p(n)^2)$, since by then, the search space has been narrowed down to contain at most one element. This element is obtained by a final (nondeterministic) guess and returned as the result. ■

Finally, we can state the following relation between our graph augmentation problems towards perfectly private transmissions and the P-vs-NP-question:

Corollary IV.9. *The following statements are equivalent:*

1) MIN-VULNERABILITY-AUGMENTATION *can be solved in polynomial time (i.e., the problem is in* FP*)*
2) MIN-COST-SECURITY *can be solved in polynomial time (i.e., the problem is in* FP*)*
3) P = NP.

Proof: Observe that FP = FP$^{\text{P}}$ obviously and that FP$^{\text{P}}$ = FP$^{\text{NP}}$ if P = NP. Together with Theorem IV.7, this implies

MIN-VULNERABILITY-AUGMENTATION $\in$ FP.

The claim for MIN-COST-SECURITY follows from Theorem IV.8. On the other hand, if either problem admits a polynomial time solution, then this is trivially an 0-approximation too, so that P = NP by Theorems IV.5 or IV.6. ■

V. Discussion and Conclusions

We stress that our treatment is entirely classical, in the sense of leaving aside arbitrarily long distance secure communication via quantum repeaters [10], [30]. Until these techniques have reached a level of maturity to see a wide range rollout, security is necessarily somewhat tied to computational intractability. However, our treatment may be extended towards further security goals failure resilience (availability) or authenticity. Both are relevant in the quantum setting with and without quantum repeaters. By a trivial change to the modeling, similar equivalences between P = NP and reputation-based authentication [31] or network path redundancy may be derived. One aspect of future considerations will thus be looking for siblings of corollary IV.9 and its related approximation problems for reliable and authentic communication. Alas, the infeasibility of graph augmentation for perfectly private transmissions is strong, since it implies that every heuristic approach to the graph augmentation problem will inevitably perform arbitrarily bad in infinitely many cases. Hence, looking for good approximations for perfect security graph augmentations is (unconditionally) pointless.

As prefigured in remark II.3, we have demonstrated that information-theoretic security and computational security both strongly relate to computational infeasibility, only in quite different ways. The situation in which we would – in the perfect security paradigm – permit the adversary an unlimited number of compromised nodes is trivial, as there is no way of perfectly secure communication without pre-shared secrets, assuming the adversary to keep the transmission network fully under his control.

The final conclusion is nevertheless a positive one: either $P \neq NP$, then strong encryptions like McElice encryption [32] or related will continue to provide a good protection against eavesdropping. Otherwise, if $P = NP$, then we can feasibly construct networks that permit communication in arbitrarily strong privacy. So, no matter how $P \overset{?}{=} NP$ is ultimately settled, confidentiality remains an achievable goal.

References

[1] C. Elliott, "The DARPA quantum network," 2007, arXiv:quant-ph/0412029v1.

[2] 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.

[3] M. Ashwin Kumar, P. R. Goundan, K. Srinathan, and C. Pandu Rangan, "On perfectly secure communication over arbitrary networks," in *PODC '02: Proc. of the twenty-first annual symposium on Principles of distributed computing*. New York, NY, USA: ACM, 2002, pp. 193–202.

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

[5] Y. Wang and Y. Desmedt, "Perfectly secure message transmission revisited," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2582–2595, 2008.

[6] M. Fitzi, M. K. Franklin, J. Garay, and S. H. Vardhan, "Towards optimal and efficient perfectly secure message transmission," in *4th Theory of Cryptography Conf. (TCC)*, ser. LNCS LNCS 4392, S. Vadhan, Ed. Springer, 2007, pp. 311–322.

[7] S. Agarwal, R. Cramer, and R. de Haan, "Asymptotically optimal two-round perfectly secure message transmission." in *CRYPTO*, 2006, pp. 394–408. [Online]. Available: http://www.iacr.org/cryptodb/archive/2006/CRYPTO/1885/1885.pdf

[8] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. on Computing*, vol. 26, pp. 1484–1509, 1997.

[9] L. Adleman, "Molecular computation of solutions to combinatorial problem," *Science*, vol. 266, pp. 1021–1024, Nov 1994.

[10] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, "Quantum repeaters based on entanglement purification," *Phys. Rev. A*, vol. 59, no. 1, pp. 169–181, Jan 1999.

[11] R. Alleaume, F. Roueff, E. Diamanti, and N. Lütkenhaus, "Topological optimization of quantum key distribution networks," *New J. of Physics*, vol. 11, p. 075002, 2009.

[12] S. Rass and P. Schartner, "Game-theoretic security analysis of quantum networks," in *Proc. of the Third Int. Conf. on Quantum, Nano and Micro Technologies*. IEEE Computer Society, February 2009, pp. 20–25.

[13] S. Rass, A. Wiegele, and P. Schartner, "Building a quantum network: How to optimize security and expenses," *Springer J. of Network and Systems Management*, vol. 18, no. 3, pp. 283–299, 2010, (published online: 23 March 2010).

[14] S. Rass and P. Schartner, "The NP-complete face of information-theoretic security," *Computer Technology and Application*, vol. 2, no. 11, pp. 893–905, 2011, david Publishing Company, ISSN 1934-7332.

[15] Y. Liang, H. V. Poor, and S. Shamai, *Information-Theoretic Security*. now Publishers Inc., 2010.

[16] O. Goldreich and S. Goldwasser, "On the possibility of basing cryptography on the assumption that P = NP," Cryptology ePrint Archive, Tech. Rep. 005, 1998/005.

[17] R. Impagliazzo and S. Rudich, "Limits on the provable consequences of one-way permutations," in *Proc. of the twenty-first annual ACM symposium on Theory of computing*, ser. STOC '89. New York, NY, USA: ACM, 1989, pp. 44–61. [Online]. Available: http://doi.acm.org/10.1145/73007.73012

[18] G. Chartrand and P. Zhang, *Introduction to Graph Theory*, ser. Higher education. Boston: McGraw-Hill, 2005.

[19] M. R. Garey and D. S. Johnson, *Computers and intractability*. New York: Freeman, 1979.

[20] S. Rass and P. Schartner, "A unified framework for the analysis of availability, reliability and security, with applications to quantum networks," *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 40, no. 5, pp. 107–119, 2010.

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

[22] S. Rass, "Information-theoretic security as an optimization problem," *J. of Next Generation Information Technology*, vol. 2, no. 3, pp. 72–83, August, 31st 2011.

[23] L. Khachian, "A polynomial algorithm in linear programming," *Soviet Math. Dokl.*, vol. 20, 1979.

[24] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. of the ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[25] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation – Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.

[26] 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.

[27] R. Gibbons, *A Primer in Game Theory*. Pearson Education Ltd., 1992.

[28] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM J. on Computing*, vol. 5, no. 4, pp. 653–665, 1976.

[29] G. Hardy and E. Wright, *An introduction to the theory of numbers*, 5th ed. Oxford Science Publications, 1984.

[30] H.-K. Lo and H. F. Chau, "Unconditional security of quantum key distribution over arbitrarily long distances," *Science*, vol. 283, pp. 2050–2056, 1999, arXiv:quant-ph/9803006.

[31] S. Rass and P. Schartner, "Multipath authentication without shared secrets and with applications in quantum networks," in *Proc. of the Int. Conf. on Security and Management (SAM)*, vol. 1. CSREA Press, July 12–15 2010, pp. 111–115.

[32] J. Buchmann and J. Ding, Eds., *Post-Quantum Cryptography*, ser. Lecture Notes in Computer Science 5299. Springer, 2008, Proceedings of the Second International Workshop, PQCrypto 2008 Cincinnati, OH, USA.