Secure Communication over Software-Defined Networks

Stefan Rass[1(✉)], Benjamin Rainer[1], Matthias Vavti[1], Johannes Göllner[2], Andreas Peer[2], and Stefan Schauer[3]

[1] Alpen-Adria-Universität Klagenfurt, Universitätsstrasse 65-67, 9020 Klagenfurt, Austria
{stefan.rass,benjamin.rainer}@aau.at, matthias.vavti@edu.aau.at
[2] National Defence Academy of the Austrian Federal Ministry of Defence and Sports, Vienna, Austria
{johannes.goellner,andreas.peer}@bmlvs.gv.at
[3] AIT Austrian Institute of Technology GmbH, Lakeside B10A, 9020 Klagenfurt, Austria
stefan.schauer@ait.ac.at

Abstract. We report on work in progress towards a practical implementation of a software defined overlay network that provides data delivery services at a freely definable and provably optimized quality of service. Our example implementation establishes transparent secure transmission, where security is in terms of confidentiality, authenticity and availability. Using general techniques from game-theory, we show how to simultaneously optimize several performance indicators of a transmission service, taking care of interdependencies and using security as a showcase application.

Keywords: Communication · Security · Pareto-optimality · Game theory

1 Introduction

Software-defined networking (SDN) provides new means of managing computer networks. It eases the provisioning of forwarding strategies, and resource allocation, and provides means to monitor traffic by separating the control plane and data plane on network devices [8] (cf. Sect. 1.1). We pursue the goal of providing secure communication over networks. Security is here a "joint" property consisting of *confidentiality*, *authenticity* and *availability*[1].

We employ SDN for realising strategies (transmission paths) in the network that provides the highest quality of security ("QoS") for a specific communication between two entities (the sender and the receiver). Nodes in-between the

[1] We deviate from the standard setting in enterprise security, where *integrity* replaces *authenticity*. However, since authenticity usually implies integrity on a technical level, we can safely go with our modified "definition" here.

© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2015
R. Giaffreda et al. (Eds.): IoT360 2014, Part II, LNICST 151, pp. 211–221, 2015.
DOI: 10.1007/978-3-319-19743-2_31

communication path may be subject to an attack. Each node has certain properties (e.g., software version, accessibility for external personnel). With the use of these properties, we derive the set of nodes that are likely to be attacked. For example, a certain software version may indicate that there exists an exploit that grants root access. The adversary may attack any node in the network, except the sender and receiver. The sender and receiver may use multipath transmissions (MPT) for communicating by employing an appropriate protocol (cf. Sect. 1.2). In order to find secure communication paths through the network, we employ game theory, where the adversary *plays* against the sender and receiver (each party trying to maximize its payoff). Our testbed allows to model a network, i.e., enterprise networks and it provides the possibility of implementing the security strategies in the network by means of an application layer protocol (cf. Sect. 1.3).

The paper is organized as follows. Section 2 provides the theoretical preliminaries on our game theoretic approach. Section 3 provides practical insights on the implementation of our approach using our testbed. Finally, Sect. 4 summarizes the paper and provides an outlook on future work.

1.1 Enterprise Communication

Secure communication in a perhaps widely distributed enterprise infrastructure is strongly dependent on the user's awareness and willingness to follow guidelines and best practices. Security breaches may indeed occur, due to users finding it difficult or cumbersome to apply proper encryption or digitally sign a message for authentication.

On the contrary, technology like virtual private networks (VPN) or transport layer security (TLS, formerly known as secure socket layer – SSL), enjoy wide acceptance and are examples of what is nowadays called "usable security" [4][2].

Pursuing this idea further, why not have a *software-defined virtual network* in a system that transparently delivers messages in a secure manner, without burdening its user with details of security?

The benefits of having a software defined network on top of a physical one (yet sharing its topology) is manifold, as (1) it minimizes risks of accidental misuse, as users are no longer directly responsible for or involved in technical matters of security, and (2) it presents a neat dual use of network redundancy for purposes beyond availability, by adding naturally to the enterprise risk management (details of which will briefly be explained in Sect. 1.2).

1.2 Multipath Transmission

Briefly speaking, multipath transmission (MPT) delivers a message over a network by utilizing multiple mutually disjoint paths (intersecting only at the sender and receiver's nodes). Using different encodings of the payload, we can use such techniques to increase throughput (split a message into parts and transmit them

[2] Here, we neglect issues of IT administration to properly set up and run the underlying system, which may be far from a trivial task.

in parallel), increase reliability (send several copies of a message in parallel) or increase security (use secret-sharing techniques to hide information from eavesdroppers on a limited subset of channels [5]).

A technical difficulty of setting up MPT in a real life network is the lack of respective support in layers below the application. Although the internet protocol – theoretically – could do source-routing along pre-defined paths (as it is necessary for MPT in wired networks with fixed topology), such features are mostly not supported by network devices or otherwise deactivated for security reasons (note the irony). The problem is less prevalent in wireless (ad hoc) networks, and suitable protocols are more developed and more intensively investigated [1]. Software-defined networks (on the application layer) let us elegantly overcome obstacles known from wired (or partially wireless) networks that would otherwise hinder the effective use of MPT.

MPT has seen fruitful applications in wireless and wired networks, towards goals of security [6,7,16], reliability [3,15] or media delivery [13]. However, common to most of that preliminary work is their focus on a *single particular* goal, leaving effects on other performance indicators of interest mostly aside. Exploiting the full potential of MPT in all its applications, is a matter of theoretical considerations on how to use MPT to get the most from it, and practical matters on how to properly run it over a network whose hardware would not play the game properly. These issues are both discussed in Sects. 2 and 3.

1.3 Our Testbed

To properly set up, test and verify the services of a software-defined network, it is useful to rebuild standard enterprise network topologies in the lab, so as to have a realistic testbed on which a software defined overlay network can be studied. To this end, let us construct our enterprise infrastructure as being a globally acting pharmacological corporation, with many subsidiaries all over the world (distribution sketched in Fig. 1) that are interconnected over an MPLS network. Intranets at each branch follow reference network topologies, such as sketched in Fig. 1 or simpler.

Now, suppose that we seek to establish a software-defined overlay network in this enterprise for a transparent, reliable and secure delivery of content within the company. The goal is thus to simultaneously optimize several performance indicators, including at least the following: (1) reliability, (2) confidentiality, (3) authenticity and (4) bandwidth/latency. The first three indicators are quantified in terms of probability (for a successful transmission), and the fourth indicator is a bandwidth estimate. Hence, we seek to establish a good quality of service (QoS), where the service level agreement is made up of several things that are potentially interdependent and require a "holistic" treatment. Section 2 sketches how this can be done in theory, and Sect. 3 reports on practical implementations thereof.

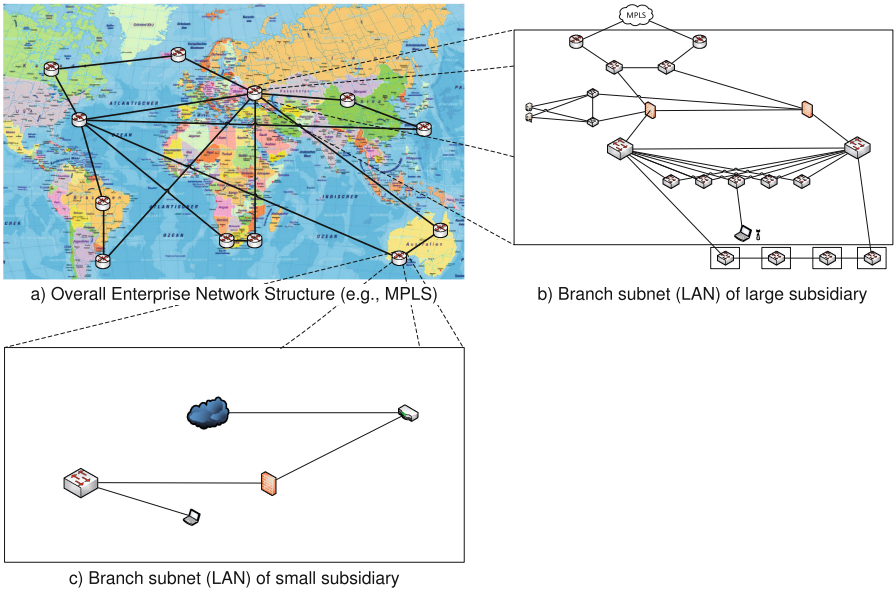

Fig. 1. Testbed

2 Theoretical Groundwork

A concept to assure optimal performance of multiple performance indicators has been discussed in [9]. While this work has been focused on security, the idea is not limited to this application. In brief, the idea of running a communication infrastructure in a way to optimize several of its performance indicators is based on certain degrees of freedom on how load is balanced in the network and how the routing is done (this is the point where a software-defined overlay network becomes most useful as it spares changes to running infrastructure).

The basic idea is bought from the concept of zero-sum games, where two players engage in a competition towards selfishly maximizing their own good at the cost of the other. Adopting the perspective of one of these players, the zero-sum strategy gives a minimum guaranteed performance, no matter what the opponent actually does (as long as his actions remain within known action sets) [2]. This can be extended towards multiple goals (latency, security, etc.), as was first done by [14] and revisited in [9]. In a nutshell, let $u_1, \ldots, u_n$ be performance indicators referring to all aspects of interest (can be probabilities, bandwidths, etc.). Given a finite set PS_1 of network provisioning strategies (i.e., different (transmission) configurations) and a finite set PS_2 of potential problem scenarios (e.g., node failures, security breaches, etc.), computing performance indicators under specific scenarios from $PS_1 \times PS_2$ is a trivial matter of *simulating* the protocols (say, in OmNet++ or any handcrafted script program). For example, MPT is straightforward to analyse, when the transmission configuration is the

set of chosen paths, and the problem scenario is a set of outage nodes. Computing the effects on the security, bandwidth, etc. is easy by a plain protocol simulation. Computing the network performance in all scenarios from $PS_1 \times PS_2$ creates a set of matrices (each having dimension $|PS_1| \times |PS_2|$, where $|\cdot|$ denotes a set's cardinality), from which we can compute an optimal network provisioning strategy $\boldsymbol{x}^*$ (being a randomized choice rule on all valid configurations from PS_1), and performance level vector $\boldsymbol{v} = (v_1, \ldots, v_n)$, with the following two properties [9]: given that the transmission parameters are (drawn from) x^*, we have

Assurance, meaning that $u_i \geq v_i$, i.e., v_i is the minimum guaranteed performance, regardless which problem scenarios from PS_2 arise with which frequency, and

Efficiency, meaning that any different transmission configuration $\boldsymbol{x} \neq \boldsymbol{x}^*$ deteriorates the performance in at least one of our indicators, i.e., there is an index i so that $u_i < v_i$.

In the background, computing $\boldsymbol{x}^*$ and $\boldsymbol{v}$ is a matter of solving an $(n+1)$-person game, in which the network provider (player 0) competes with n opponents, each of which seeks to minimize the network performance in a different regard (zero-sum regime on each indicator). This zero-sum construction yields assurance and efficiency exactly as it does in the scalar case of a single performance indicator. It is therefore occasionally referred to as a *(Pareto-optimal) security strategy* [14]. For example, in seeking minimal latency, we may define $\boldsymbol{x}^*$ as the rule to always choose the currently most reliable path(s). Likewise, towards best security, we may choose the paths with best protection (not necessarily being the most reliable ones). The framework of [9] shows how to simultaneously take care of all these goals. The numerical computation of $\boldsymbol{x}^*$ and $\boldsymbol{v}$ is possible by an iterative algorithm, adapted from [12] (showing how to solve "one-against all"-type games), which we implemented in our prototype. The tricky part is to have the infrastructure do the MPT according to the optimal (randomized) configurations $\boldsymbol{x}^*$, which is where software-defined networks come in extremely handy.

3 Practical Implementation

Basically, our prototype implementation does randomized source-routing on the application layer (OSI layer 7). More precisely, given a topology with redundant connections (found in most reference network structures such as sketched in Fig. 1), $\boldsymbol{x}^*$ is a set of paths (or path bundles) that shall be selected with prescribed probabilities (i.e., $\boldsymbol{x}^*$ is actually a probability distribution supported on the transmission configuration set PS_1). The network itself is defined by a set of instances of the client software, running at different machines in the network. Each client can act as sender, receiver or (passive) relay on layer 7, where

confidentiality is assured by cryptographically enhanced MPT[3]. Availability is determined by whether or not *all* packets reach their destination. Authenticity is achieved by a simple multipath authentication scheme detailed in [11].

The computation of $\boldsymbol{x}^*$ and its implied quality-of-service vector v are computed by an enhanced version of the system described in [10], implementing the method of [12] to compute x^* and $\boldsymbol{v}$ as defined in [9].

A Worked Example: To practically test and demonstrate the feasibility and security of our system, we implemented a Java demonstrator application that handles the routing and cryptographic operations necessary to deliver a message securely from a sender to a receiver. The example network that we treat here is simplified for plausibility without requiring the reader to do the math underneath the theoretical groundwork (as sketched in Sect. 2) to "verify" the correctness of the results and the example.

The network consists of five nodes that are interconnected as shown in Fig. 2. For authentication, the protocol in [11] adds message authentication codes (MACs) to the payload that are based on secrets shared with other nodes in the network (common secrets being indicated by dashed connections). To verify the authenticity of a received message, the receiver simply asks other nodes in the network to verify the MAC. In that sense, the system sort of resembles how handwritten signatures can be checked on printed documents without electronic or cryptographic help.

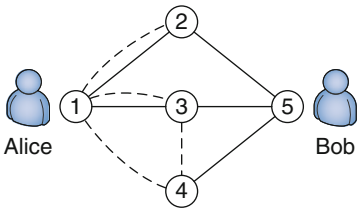

Fig. 2. Example network

The efficient assurance of optimal security in terms of authenticity, availability and confidentiality uses single-path transmission over a randomly chosen path $1 \to x \to 5$ where x is chosen uniformly from $\{2, 3, 4\}$. It is easy to see that an attacker gaining control over one or two nodes has only a one-out-of-three chance to learn the information, which yields a 66 % chance of the message being

[3] Actually, a rather simplified version of perfectly secure MPT, which splits a message m into a set of random strings so that their XOR recreates m. Despite there being much better practical protocols, in case of two-path transmissions, our scheme is isomorphic to a one-time pad and thus unbreakable. This security is, however, bought at a higher risk of communication failure in case that one or more packets get lost. Thus, the two goals "confidentiality" and "availability" are somewhat conflicting.

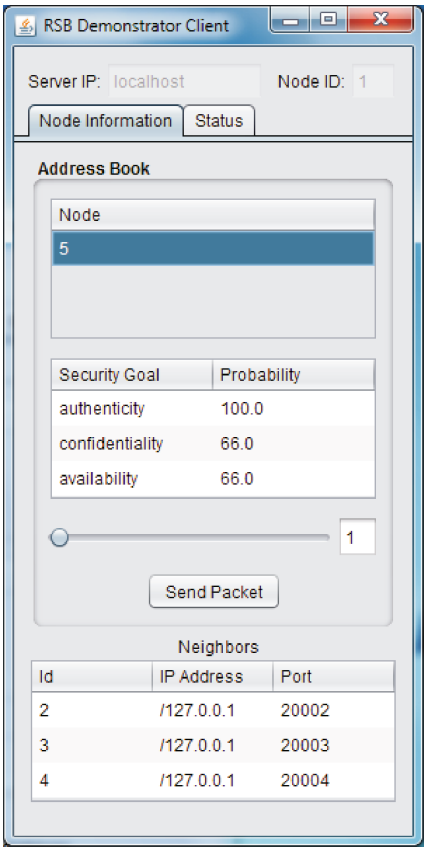

Fig. 3. Sender's (Alice's) view (demonstrator prototype)

delivered (availability) in privacy (confidentiality). If the receiver Bob asks all three nodes $\{2, 3, 4\}$ to verify the attached MACs, then there is even a 100 % guarantee of a forgery to be detected upon one rejected MAC verification. This security assurance is displayed in Alice's window (corresponding to node 1 and shown in Fig. 3). The address book shows to which receivers (in our case only node 5, which is Bob) she can deliver messages to. The lowest part of the window shows this node's direct neighbors in the network. This is important to demonstrate that a node needs only local information on the network topology, as it is concerned only with where to send the packet away, but it does not need to know the full network topology. The entire transmission works along several (in our case only one) intermediate node, any of which needs only local (and hence minimal) information about the full network.

The likelihoods of a confidential delivery can, at the cost of investing much additional transmission overhead, be raised arbitrarily close to 1 by repeating the process to deliver a set of random numbers $r_1, \ldots, r_{n-1}$ and encrypting

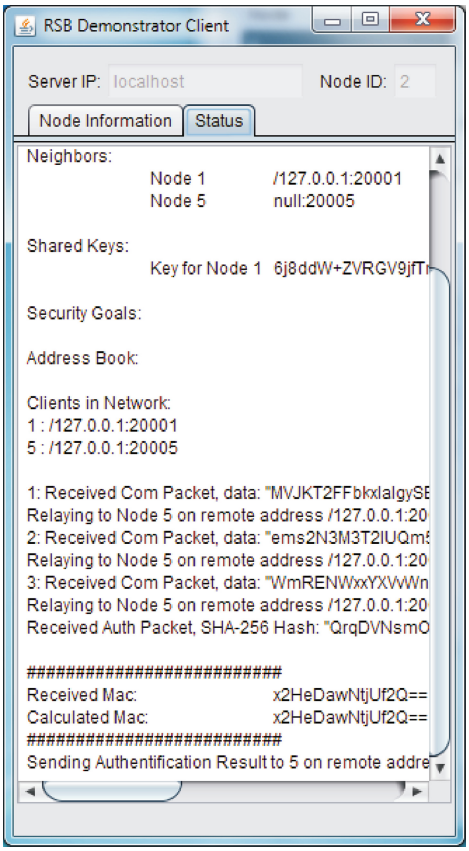

Fig. 4. View and data of intermediate node 2 (demonstrator prototype)

the payload m in the final blow as $r_n := m \oplus r_1 \oplus \cdots \oplus r_n$ (where $\oplus$ is the bitwise XOR), so that each $r_1, \ldots, r_{n-1}$ on its own would perfectly conceal the message m like a one-time pad. In turn, this requires all packets to be correctly delivered, thus lowering availability of the channel in much the same way. More sophisticated error-correcting transmission schemes (see e.g., [5]) can elegantly cover for this tradeoff, but are outside our scope here.

Our prototype can be configured to take any number of given rounds; in the example this would be $n = 5$ repetitions. Figure 5 shows a log print of all information that this node receives, displaying the recovered message ("Hello Bob!") in the middle of the window. As the log of the intermediate node 2 shows, see Fig. 4, this node receives some but not all (only three of the five) packets necessary to reconstruct the final message for Bob; the entirety of required messages being listed in the log of Fig. 5. So, a potentially hostile node 2 would – in an information-theoretic sense – be unable to learn anything from sniffing on the network traffic.

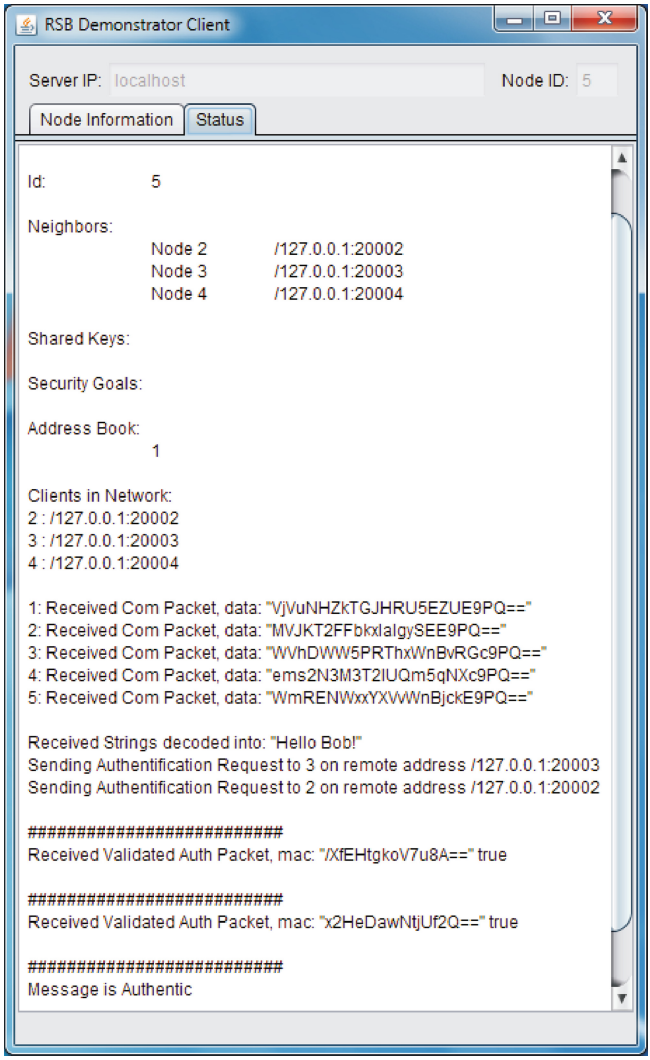

Fig. 5. Receiver's (Bob's) view (demonstrator prototype)

The other information displayed in the window(s) relates to network topology information and the (base64-encoded) authentication keys shared with the neighbors (in case of node 2, this would be only node 1, with which the dashed edge indicates the existence of a shared key for MAC verification. In turn, node 1 would use this shared key to have node 2 verify the MAC that Alice attached originally (the details of this authenticity verification conversation are as well displayed in the log files of the involved nodes; Figs. 4 and 5).

4 Conclusion

The lesson learnt from our practical experiments on the theoretical concept of security strategies (SS) is twofold: first, an SS is a way in which a network can be utilized towards a guaranteed quality-of-service in multiple and interdependent aspects. This QoS is assured independently of any problem occurrence within a known set of scenarios. Despite the name "security strategy" and security being a nice showcase application, the concept sketched in Sect. 2 is in *no way restricted to security* and can be applied to many other QoS indicators straightforwardly. Second, software defined networks make an implementation of such security strategies most simple and feasible, as SDN give the full freedom to implement such optimal network utilization regimes without having to worry too much about underlying technical circumstances or limitations. Thus, applications reaching far beyond the security scope are imaginable, which this work may stipulate.

Acknowledgements. This work was supported by the Austrian Research Promotion Agency (FFG) under project grant no. 836287.

References

1. Abbas, A.: A hybrid protocol for identification of a maximal set of node disjoint paths. Int. Arab J. Inf. Technol. (IAJIT) **6**(4), 344–358 (2009)
2. Alpcan, T., Başar, T.: Network Security: A Decision and Game Theoretic Approach. Cambridge University Press, New York (2010)
3. Djukic, P., Valaee, S.: Reliable packet transmissions in multipath routed wireless networks. IEEE Trans. Mob. Comput. **5**, 548–559 (2006). http://doi.ieeecomputersociety.org/10.1109/TMC.2006.72
4. Finley, K.: Online security is a total pain, but that may soon change (2014). http://www.wired.com/2014/06/usable-security/
5. Fitzi, M., Franklin, M., Garay, J., Vardhan, S.H.: Towards optimal and efficient perfectly secure message transmission. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 311–322. Springer, Heidelberg (2007)
6. Kotzanikolaou, P., Mavropodi, R., Douligeris, C.: Secure multipath routing for mobile ad hoc networks. In: International Conference on Wireless on Demand Network Systems and Service, pp. 89–96. IEEE Computer Society, Los Alamitos (2005).doi:http://doi.ieeecomputersociety.org/10.1109/WONS.2005.31
7. Li, Z., Kwok, Y.K.: A new multipath routing approach to enhancing TCP security in ad hoc wireless networks. In: International Conference Workshops on Parallel Processing, pp. 372–379 (2005). doi:10.1109/ICPPW.2005.11
8. Nunes, B., Mendonca, M., Nguyen, X., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. In: Communications Surveys Tutorials, vol. (99), pp. 1–18. IEEE (2014). doi:10.1109/SURV.2014.012214.00180
9. Rass, S.: On game-theoretic network security provisioning. J. Netw. Syst. Manage. **21**(1), 47–64 (2013). http://www.springerlink.com/openurl.asp?genre=articleid=doi:10.1007/s10922-012-9229-1

10. Rass, S., Rainer, B., Vavti, M., Schauer, S.: A network modeling and analysis tool for perfectly secure communication. In: Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications, pp. 267–275. IEEE Computer Society Press (2013, in press)
11. Rass, S., Schartner, P.: Multipath authentication without shared secrets andwith applications in quantum networks. In: Proceedings of the International Conference on Security and Management (SAM), vol. 1, pp. 111–115. CSREA Press (2010)
12. Sela, A.: Fictitious play in one-against-all multi-player games. Economic Theory 14, 635–651 (1999). doi:10.1007/s001990050345
13. Singh, V., Ahsan, S., Ott, J.: MPRTP: multipath considerations for real-time media. ACM Multimedia Systems Conference (2013)
14. Voorneveld, M.: Pareto-optimal security strategies as minimax strategies of a standard matrix game. J. Optim. Theory Appl. **102**(1), 203–210 (1999)
15. Wen, H., Lin, C., Yang, H., Ren, F., Yue, Y.: Modeling the reliability of packet group transmission in wireless network (2007). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.132.483
16. Zhao, L., Delgado-Frias, J.: Multipath routing based secure data transmission in ad hoc networks. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communication, pp. 17–23 (2006). doi:http://doi.ieeecomputersociety.org/10.1109/WIMOB.2006.1696359