appeared in: Proceedings of the International Conference on Security and Management (SAM), Vol. 1, CSREA Press, pp. 111-115, 2010.

Multipath Authentication without shared Secrets and with Applications in Quantum Networks

Stefan Rass, Peter Schartner
Institute of Applied Informatics
System Security Group
Klagenfurt University
9020 Klagenfurt, Austria
Email: firstname.lastname@uni-klu.ac.at

***Abstract*—Authentication is possible in various ways, the most obvious of which is by re-calculating a given authentication tag upon a private secret shared only with the message's origin. What, however, if no such shared secrets are available? Public-key cryptography has elegantly solved the problem by relying on computational intractability assumptions. In the light of increasing computational power, can we achieve the same thing in the secret-key paradigm? In a quantum network, we may have shared secrets between adjacent nodes, making an authentic communication between these simple, but what about end-to-end authentication? We present a simple method of authenticating a transmission that requires shared secrets only between neighboring nodes in a (quantum) network, but the sender and receiver (being farther apart) do not need to share any secret knowledge. Our approach is inspired by multipath transmission and thus naturally compatible with this paradigm. The security of the proposed method is analyzed based on decision-theoretic considerations, and therefore does not hinge on any computational intractability assumption.**

I. Introduction

Authentication is the task of ensuring an entity's identity prior to or during a communication, and as such, a crucial ingredient to many security systems. In the public-key world, various ingenious authentication schemes have been established, ranging from challenge-and-response protocols to sophisticated zero-knowledge proofs and powerful credential systems. In the world of secret-key cryptography, authentication is much simpler, yet less powerful, as it is based on shared secrets and verification of key-based message authentication codes (MACs). However, if public-key cryptography should be avoided and if no pre-shared information between the sender and receiver is available, is there another way of authenticating a message? Our approach to a positive answer is inspired by multipath transmission and public-key cryptography: similarly as for public-key infrastructures, we can have other instances in the network provide the certificates that some message is authentic. So why not use a multipath authentication approach to gain confidence that some message is truly coming from the purported source? Using a decision-theoretic approach, we establish a simple authentication mechanism that works without pre-shared secrets and requires a moderate communication overhead, whilst enjoying provable security under the weak assumption stated below.

In a nutshell, the idea is stated as follows: unless Alice shares some secret with Bob, the best she can do is referring Bob to others that she shares secrets with for validating a given authentication tag. Bob, similarly as in a public-key infrastructure, can query these other instances to convince himself about the correctness of the tag. In turn, he would neither want to send the message all over again, for the sake of privacy and network traffic overhead. Hence, instead of tagging the message, Alice tags the hash-value of the message, which does not release much information about the message and causes only minor additional traffic. To avoid person-in-the-middle attacks somewhere on the channel between Alice and Bob, Bob can use multiple (non-intersecting) paths from his node to Alice's neighbors (acting as verification authorities). Even if the adversary is sufficiently powerful to cut the channel once, changing the paths and repeating the trial can be done in a way that reduces the adversary's chance of forgery below any acceptable threshold. This is the core idea behind the game-theoretic security arguments presented below. Before coming to the details, however, let us loose a few words about related work in this area.

II. Related Work

Quantum key distribution (QKD) is claimed to bring perfect secrecy to a line connecting two parties in a network. Endowing all links with QKD devices, we end up with a *quantum network*. Authentication is a widely recognized problem, and has seen different solutions. A standard method, known as *continuous authentication*, is described in [1], but is only applicable to point-to-point authentication. Most closely related to our work is [2], describing how to achieve fundamental goals like confidentiality and authenticity if some secret is shared, but an arbitrarily large portion of this is known to the adversary. We attempt to achieve the same thing without any pre-shared secret at all. The concept of multipath authentication has as well appeared in the field of sensor networks and SSL [3], [4], [5], where the approach can be combined with multipath transmission for efficiency. Another infrastructure-based authentication proposal is found in [6]. Contrary to the prior work in this field, our decision theoretic approach is novel from both points of view: theoretical treatment and the assumption of no pre-shared information.

appeared in: Proceedings of the International Conference on Security and Management (SAM), Vol. 1, CSREA Press, pp. 111-115, 2010.

III. Authentication Schemes

Consider a quantum network in which each pair of adjacent nodes shares a secret, e.g. established by means of QKD. Furthermore, suppose that Alice and Bob do not share any secret information that could be used for end-to-end authentication. How should Alice convince Bob that some message from her is authentic? Assume that Alice has n neighboring nodes, each of which she shares some secret with. Why not use one of them as reputation that she really is Alice? If single-path transmission is employed in the way sketched in figure 1, then each forwarding node can check authenticity of the incoming message (verification block V), and can verify its origin. Upon passing it onwards, it attaches the tag t_i, which is a MAC created from the secret k_i shared between the current and the next hop. By the time Bob receives the message, he may be truly convinced that the message is really coming from his immediate neighbor, but farther than this one, he ought to trust all nodes that forwarded the message from Alice's node onwards. One way to relieve this burden is to employ several paths to check Alice's identity tag on the message, similarly as for multipath transmission. However, to avoid handing over the message to too many instances for checking, Alice should not authenticate the message itself, but rather send an authentication code referring to a universal hash-value of the message (see [7], [8], [9] for appropriate constructions). Then, this hash-value, along with its authentication code can be given to other nodes for checking, making the message itself invisible to them, but still allowing them to verify the MAC on behalf of Bob.

Formally, let $\mathcal{H} := \left\{ h : \{0,1\}^* \times \{0,1\}^l \to \{0,1\}^k \right\}$ be a strongly universal hash-family with key-length l, and let $m \in \{0,1\}^*$ denote a general message in the following. Assume that a publicly available hash-function $h(\cdot, K)$ with a fixed and known key K is available that can be used for general-purpose hashing. Standard functions such as SHA-1 or RIPEMD-160 are candidates, but we shall stick with universal classes here for the sake of available theoretical assertions about them.

Having multiple paths, a simple way of protecting messages from an active threshold adversary is, sending the message over one path and transmitting its hash-value over other node-disjoint paths. This however, does not prove the message authentic for Bob, because what should stop the adversary from inserting a correct hash for a forged message into the channel? Recalling that we assume the channels unprotected because of unlimited power of the adversary, this method will not work. Hence, we require some secret knowledge of Alice to determine the hash-value. The only secrets that Alice shares with others are the secret keys held by her and her direct neighbors. So, recycling the idea from before, Alice could perform as follows to make Bob accept her message as authentic.

1) For each neighboring node $X_i, \quad (i = 1, \ldots, n)$, that Alice shares a secret k_i with, she creates the message authentication code $MAC_i = h(t_i, k_i)$ where $t_i = h(m, K)$. She sends the message m over one path, and sends each MAC over its own distinct path, being disjoint to all other paths, to Bob.
2) On each path we have hop-by-hop authentication, and Bob accepts her message as authentic if all incoming verifications are positive.

A slight drawback of this method is the requirement of all paths being live and available between Alice and Bob during the communication. Relaxing this need makes the protocol more suitable for ad hoc networks. Why not have Bob actively query a selection of instances that Alice has created MACs for to verify authenticity? In the next protocol, Bob receives all MACs and hands them over to Alices neighbors for verification. In that case, Alice does not need to have a live connection to her neighbors by the time of MAC verification. For sending an authentic message, however, only one path is sufficient. The protocol is as follows:

1) For each neighboring node $X_i, \quad (i = 1, \ldots, n)$, that Alice shares a secret k_i with, she creates the message authentication code $MAC_i = h(t_i, k_i)$ where $t_i = h(m, K)$, and attaches the sequence of MAC_i's as an authentication tag to the message.
2) Upon receiving the tuple $(m, MAC_1, \ldots, MAC_n)$, Bob recreates $t_i := h(m, K)$ and sends the pair $(t_i, MAC_i, ID_{\text{Alice}})$ to X_i for MAC verification over non-intersecting paths (the string ID_{Alice} can be any descriptor of Alice's identity in the network). He accepts m as authentic, if and only if each verification poll comes back positive (over the same path that the query has traveled over).

Notice that this also gives rise to a neat way of authentication in ad hoc networks of highly unstable topology. Alice can establish and store keys with any node sufficiently close to her for later usage for authentication. She maintains a list of nodes that she shares secrets with. At a later stage, she can create MACs with secrets that she shares with nodes from the list and send the corresponding tags to Bob. For verification, Alice only needs to tell which nodes can verify her MAC on her behalf (observe the analogy to real-life social networking, where people give reference to other persons for confirming their reputation).

It is easy to see that unless the adversary has conquered all of the X_i's, or equivalently, one node on each path between X_i and Bob for all i, at least one MAC verification will fail for a forged message with high probability. Notice the strong similarity to multipath transmission, where it has been shown [10] that the existence of several non-intersecting paths is a necessary condition for perfectly concealed communication. As far as confidentiality and communication overhead is concerned, the amount of transmitted data is no more than n times the size of the two MACs for transmission of the MAC and the hash of the message. The neighbors of Alice will not gain more than $m = |t_i|$ bit of information, because the message m has been compiled into a hash-value t_i. On the other hand, the secrets shared between Alice and any of her neighbors X_i will not become visible to Bob, unless he

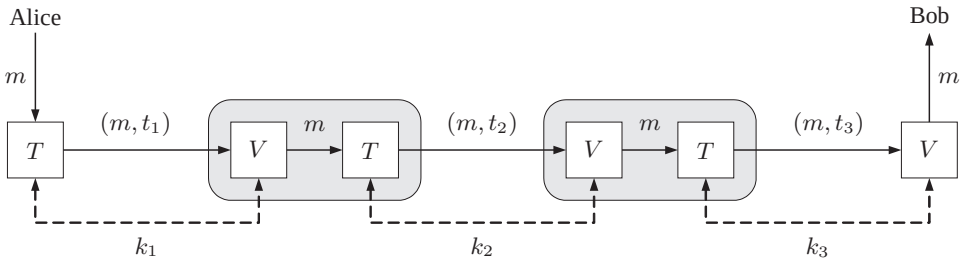

Fig. 1. Hop-by-hop MAC verification V and (re-)tagging T

manages to solve the equation $h(h(m,K),k_i) = MAC_i$ for k_i. If an evaluation hash family [8] is chosen, then k_i is a root of a polynomial equation of a degree less than the number of blocks in the message. In some cases, this task may be feasible, so it is advisable, in any case, to discard the authentication secrets after usage. Alternatively, one could use another hash function H (like RIPEMD-160 or similar) in between to destroy algebraic properties possibly revealing the key k_i. One proposal is using $MAC_i = h(h(m,K),[k_i \cdot H(k_i)])$, but others are imaginable.

IV. Security Analysis

It is easy to prove that once the graph is n-vertex-connected, then at least n node-disjoint paths exist between Bob and the n neighbors of Alice [11]. Equally obvious is that no adversary with threshold strictly less than n is able to successfully forge a message as coming from Alice. However, n-connectivity is a rare property and hardly observed in real-life networks. Can we obtain a result that holds for general networks, not imposing any constraint on the connectivity? The answer is yes, by relying on a decision-theoretic line of arguments.

A. The Authentication Game

Our security analysis will be based on the ideas that have been applied to analyze end-to-end transmission security in quantum networks [12]. The idea is easily applied to our setting (an example is provided in section V): suppose that a (possibly singleton) set of paths, say of cardinality n, between Alice and Bob exists. Call t the threshold of the adversary structure under consideration, i.e. if $\mathcal{A}$ is an adversary structure, then $t := \max\{|M| : M \in \mathcal{A}\}$. If $t \geq n$, then the adversary could be able to entirely intercept the channel between Alice and Bob, and without pre-shared secrets, authentication is doomed in any case. Otherwise, there may be scenarios in which authentication is working, and others in which it will fail. Let Alice and Bob set up a matrix with entries $a_{ij} \in \{0,1\}$, where $a_{ij} = 0$ indicates a successful attack, whilst $a_{ij} = 1$ means that authentication worked correctly. The index i ranges over the degrees of freedom that Alice and Bob enjoy for authentication (i.e. the set of neighbors that Alice can use for proving her reputation). We call this set PS_1 and refer to it as *pure strategy set*. The column index j indicates what attack strategy, i.e. set of conquered nodes $A \in \mathcal{A}$ is under the adversary's control at the moment. This set is as well called a *pure strategy set*, and denoted as PS_2. The particular choice $j \in PS_2$ is assumed unknown to Alice and Bob.

The approach put forth in [12] proceeds by defining the quantity $\rho(A) = 1 - v(A)$, where $v(A)$ is the Nash-equilibrium saddle-point value of the game induced by the matrix A. The quantity $\rho(A)$ is called the *vulnerability*. The value $v(A)$ can be found by solving a standard linear optimization program (most textbooks on game-theory, e.g. [13], provide full details on this), and is defined as $v(A) = \max_{x \in S_1} \min_{y \in S_2} x^T A y$, where S_1, S_2 are simplexes over the strategy sets of Alice and Bob. In other words, S_1 is the set of probability distributions over the actions that Alice and Bob can take, and S_2 is the set of probability distributions over the set of attack strategies of the adversary. Everything that follows hinges on Alice and Bob acting according to their optimal Nash-equilibrium strategy, while the adversary is free to act anyhow, as long as no unknown strategy is played. In this case, the adversary model would be wrong.

The intuition behind this framework is simple: if a communication is modeled by a game between the honest instances and the adversary, then pure strategies refer to the degrees of freedom that all participants enjoy when taking actions. The Nash-equilibrium is a probability distribution that describes an optimal selection of strategies to achieve maximum average revenue from the game under infinite repetitions. In other words, if strategies are chosen according to the Nash-equilibrium distribution, then the expected outcome $v(A)$ is optimal for both sides (i.e. for Alice and Bob, as well as the attacker). Notice, however, that this implies a "zero-sum assumption", i.e. the adversary considers any damage to Alice and Bob as his direct revenue. This may be a totally inaccurate modeling, but provides a valid worst-case assumption from Alice and Bob's point of view. Finally, the quantity $\rho(A) = 1 - v(A)$ is the difference between the maximum outcome $(= 1)$ and the actual average outcome. This is the loss that Alice and Bob suffer on average, and therefore called vulnerability. This quantity enjoys various other useful theoretical properties; see [14] for details, and section V for a numerical example.

B. Results

We define the random outcome of the i-th authentication attempt as $a_{ij} := U_i$, where $U_i = 1$ indicates a truly authentic

message delivery, and $U_i = 0$ indicates a successful forgery by the adversary. It is important to notice that this random variable is never observable, and that Alice and Bob cannot tell apart the case $U_i = 1$ from the case $U_i = 0$. Still, they can be sure that authentication works correctly with arbitrarily high probability, as we will show.

Our analysis will be based on the following result, whose generalized version appears in [14, Thm. 5.3.32].

Theorem IV.1. *Let A denote the game matrix that Alice and Bob set up for capturing different scenarios (as described above). In r (possibly interdependent) repetitions of the given protocol, in which Alice and Bob exhibit a Nash-equilibrium behavior, and for any $\varepsilon \geq 0$,*

$$\mathsf{P}\left(\min_{1\leq i\leq r} L_i \geq \rho(A) + \varepsilon\right) \leq \exp\left(\frac{-r\varepsilon^2}{2}\right),$$

if $L_i := 1 - U_i$ denotes the loss if the adversary mounts an attack on the i-th transmission. The adversary is not *bound to follow the Nash-equilibrium strategy for the result to hold.*

By construction, Bob will not accept a message as authentic unless *all* tags have been verified positively. Hence, the adversary is successful, if and only if $L_i = 1$ for all i (neighbors of Alice). However, the event $(L_i = 1 \quad \forall i)$ is the same as $\min_{1\leq i\leq r} L_i = 1$ (remember that $L_i \in \{0, 1\}$), which happens with probability at most $\exp\left(\frac{-r(1-\rho(A))^2}{2}\right)$, by theorem IV.1.

So, Bob can repeatedly ask Alice to send the message to have the probability of success for the adversary arbitrarily small. However, for theorem IV.1 to hold, Alice and Bob are required to choose their strategies randomly according to the Nash-equilibrium distribution derived from the zero-sum game $\Gamma(A)$. If an attacker fiddles with some messages in between, then Bob will reject the message, which amounts in nothing else as a denial-of-service for Alice. This, in turn, can be prevented by further measures, which are not subject of this work. However, if Bob demands that the message must be authentic with probability at least p, then the last $\Omega(\log(p))$ message verifications should confirm authenticity.

Summarizing these considerations, we can state the following conclusion: let Alice and Bob set up an authentication game, where they have n strategies (i.e. degrees of freedom) for proving authenticity by relying on neighbors of Alice for authentication tag verification. Let the adversary be able to mount an attack using s different scenarios. Then a matrix $A \in \{0, 1\}^{n\times s}$ can be set up, and the vulnerability value $\rho(A)$ can be derived using standard techniques from linear optimization. For any (small) security parameter $p > 0$, Bob can run $\Omega(\log(p))$ repetitions of the transmission to be sure that Alice's message is authentic with probability at least $1 - p$. This requires no pre-shared information between Alice and Bob to work, and only relies on secrets shared between adjacent nodes.

It is worthwhile to loose a few words about what happens if $\rho(A) = 1$. In that case, we have $v(A) = 0$, and the average number of forgery detections vanishes in the long run. This means that the adversary will succeed with probability 1, and authentication will not work. To see this, recall that $\rho(A) = 1 - v(A)$, and that $v(A)$, by definition, is the average outcome of the game under infinite repetitions. However, if $v(A) = 0$ then the average number of forgeries must asymptotically outweigh the number of detections, and the probability of succeeding in detecting the adversary is zero.

Theorem IV.2. *Let A denote the matrix, modeling the authentication game as described above, and let Alice share secrets of length l with $n > 1$ neighbors of hers, but no secret with Bob is shared. If $\rho(A) < 1$, then Alice can transmit a message to Bob in an authentic manner, where the probability of forgery is less than 2^{-l}. If $\rho(A) = 1$, then the adversary will be able to forge messages with probability 1.*

We emphasize that this result calls Alice and Bob to act according to the Nash-equilibrium strategy. However, to remain synchronized in their (pseudo-random) actions, Alice and Bob would in turn require some secret to be used as seed for their random number generators. To avoid this need, Alice could in general attach tags for all her neighbors, while Bob selects some of them for verification as prescribed by his strategy. This spares interaction with Alice and the responsibility to abide by the Nash-equilibrium is on Bob's side.

A further way of fiddling with the scheme is by forging the answer to the verification queries. However, unless the path has been compromised, there will be no change to this information. If the path has been compromised, then either the forgery will succeed with acceptance by Bob, or it will not. If it succeeds, then the adversary is required to perform as good in the next rounds. The chance for this will become negligible as the number of rounds grows.

It is important to notice that the network topology is implicitly assumed as known to every node in the network. As such, the scheme may appear difficult to implement in wireless ad hoc networks as it is. This is subject of ongoing work. We therefore advise to combine it with existing multipath transmission methods (such as [15], [16], [17]), where this problem has been considered already.

V. Example

Let a network topology be given, as shown in figure 2. Assume that the adversary can compromise up to 2 nodes from the set $\{1, 3, 4\}$ (the adversary structure is thus $\mathcal{A} = \{\{1, 3\}, \{1, 4\}, \{3, 4\}\}$). Alice and Bob perform multipath authentication over two paths, i.e. Alice attaches two MACs to be verified by her neighbors upon Bob's enquiry.

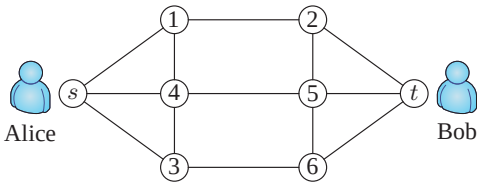

Fig. 2. Example network topology

Assume that Alice and Bob have picked three pairs of shortest node-disjoint paths, disregarding other possibly longer paths. So the set of pure strategies is $PS_1 = \{s_1^{AB}, s_2^{AB}, s_3^{AB}\}$, and given by

s_1^{AB} : Use paths s—1—2—t and s—3—6—t,
s_2^{AB} : Use paths s—1—2—t and s—4—5—t,
s_3^{AB} : Use paths s—3—6—t and s—4—5—t.

Eve is considered a 1-active adversary with structure $\mathcal{A}$, hence her strategies are attacking one set in $\mathcal{A}$. This means that her set of pure strategies is given by $PS_2 = \mathcal{A} = \{s_1^E, s_2^E, s_3^E\}$, where

s_1^E : Compromise nodes 1 and 3,
s_2^E : Compromise nodes 1 and 4,
s_3^E : Compromise nodes 3 and 4.

Writing down every possible combination of pure strategies in a matrix A, with entry 1 if the attack fails, we end up with the following table:

A	s_1^E	s_2^E	s_3^E
s_1^{AB}	0	1	1
s_2^{AB}	1	0	1
s_3^{AB}	1	1	0

The Nash-equilibrium value is found as $v(A) = 1/3$, and in turn $\rho(A) = 2/3$. Since $\rho(A) < 1$, we know that an authentic transmission is possible, and by solving the equation $\exp(-n(1-\rho(A))^2/2) < p$, we can instantly determine the number n of rounds to attain the threshold probability p. Notice, however, that the overall communication overhead is still small, considering that we only transmit tags of expectedly short length (128 bit for example).

It is equally easy to see from the example, and in general, that if the adversary's threshold is strictly less than the number of paths used for MAC verification, then there is no way of successfully forging a message. In that case, the vulnerability is zero, and multipath transmission, along with multipath authentication, with the marriage of both being obvious, is a method of perfectly secure communication.

VI. Conclusion

We presented a simple method of authenticating a transmission between two parties that do not initially share a common secret. Considering a network where only adjacent nodes share common secrets (a situation that can be expected to become reality in future quantum networks), other instances in the network can act as verification authorities for a given message authentication code. By creating a MAC for the hash-value of the message rather than for the message itself, the network traffic is kept low for the verifying nodes, whilst letting them correctly check the MAC on behalf of the recipient. We established a connex to multipath transmission and demonstrated that a game-theoretic security analysis yields a simple criterion for deciding whether or not authentication will fail eventually. It is obvious that this method can easily be combined with multipath transmission mechanisms, and requires only little computational overhead, as the entire workload is well distributed across the network. Regarding communication overhead, the scheme is cheap as it requires only the transmission of short tags rather than the original message for MAC verification.

References

[1] G. Gilbert and M. Hamrick, "Practical quantum cryptography: A comprehensive analysis (part one)," 2000, uRL: http://www.citebase.org/abstract?id=oai:arXiv.org:quant-ph/0009027.

[2] R. Renner and S. Wolf, "Unconditional authenticity and privacy from an arbitrarily weak secret," 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.125.1945

[3] H. Vogt, "Exploring message authentication in sensor networks," in *ESAS*, 2004, pp. 19–30.

[4] D. Wendlandt, D. Andersen, and A. Perrig, "Perspectives: Improving SSH-style host authentication with multi-path probing," in *Proc. of USENIX Annual Technical Conference*, Boston, MA, June 2008.

[5] M. Alicherry and A. D. Keromytis, "Doublecheck: Multi-path verification against man-in-the-middle attacks." in *ISCC*. IEEE, 2009, pp. 557–563.

[6] B. Xie, A. Kumar, and D. P. Agrawal, "Secure interconnection protocol for integrated internet and ad hoc networks," *Wirel. Commun. Mob. Comput.*, vol. 8, no. 9, pp. 1129–1148, 2008.

[7] D. R. Stinson, "Universal hashing and authentication codes," in *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1992, pp. 74–85.

[8] V. Shoup, "On fast and provably secure message authentication based on universal hashing," in *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1996, pp. 313–328, http://www.shoup.net/papers/.

[9] J. Bierbrauer, "Authentication via algebraic-geometric codes," *Rend. Circ. Mat. Palermo (2) Suppl.*, vol. 51, pp. 139–152, 1998.

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

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

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

[13] P. Morris, *Introduction to game theory*. Springer, 1994.

[14] S. Rass, "On information-theoretic security: Contemporary problems and solutions," Ph.D. dissertation, Klagenfurt University, Institute of Applied Informatics, June 2009.

[15] W. Lou, W. Liu, Y. Zhang, and Y. Fang, "SPREAD: Improving network security by multipath routing in mobile ad hoc networks," *Wireless Networks*, vol. 15, no. 3, pp. 279–294, 2009.

[16] Z. Li and Y.-K. Kwok, "A new multipath routing approach to enhancing TCP security in ad hoc wireless networks," in *International Conference Workshops on Parallel Processing*, june 2005, pp. 372–379.

[17] P. Kotzanikolaou, R. Mavropodi, and C. Douligeris, "Secure multipath routing for mobile ad hoc networks," in *International Conference on Wireless on Demand Network Systems and Service*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2005, pp. 89–96.