

Article

Secured by Fluctuating Topology Using the Fluctuating Topology of MANETs to Secure Key Exchange

Ariel Stulman ^{1,*}  and Alan Stulman ²¹ Department of Computer Science, Jerusalem College of Technology, Jerusalem 91160, Israel² Department of Industrial Engineering, Jerusalem College of Technology, Jerusalem 91160, Israel; alan@jct.ac.il

* Correspondence: stulman@jct.ac.il

Received: 2 August 2019; Accepted: 8 October 2019; Published: 16 October 2019



Abstract: VANETS, IOT, and many other acronyms represent exciting new technologies that build upon pervasive computation and communications to achieve their goal. As an underlying communication model, the (mobile) ad hoc network (MANET) paradigm is used, which implements a peer-to-peer communication model rather than the more traditional infrastructure model. Of course, privacy, confidentiality, integrity, and security related issues are of utter importance in such contexts as well. In this paper, we wish to present a key exchange technique, which builds upon the inherent characteristic of MANETS: their fluctuating topology. By splitting key exchange information into multiple parts and spraying them over space or time, the ever-changing topology of the network almost completely removes an active attacker's success ratio. Algorithms are then simulated, and the results presented and discussed. We further point to future directions and uses for this research.

Keywords: key exchange; confidentiality; ad hoc networks (MANETS); mobile security; security protocols

1. Introduction

Mobile ad hoc networks (MANETS) are becoming the basic building block in up-and-coming technologies such as vehicular networks (VANETS), internet of things (IoT), and many others. What differentiates between these networks and previous, well-studied networks, is their communication model, which is based on a peer-to-peer paradigm, rather than the infrastructured alternative [1]. MANET nodes act as both hosts (receiver and sender) and routers, passing control and data packets throughout the network using any means of communication links available (see [2]). Such capabilities are indispensable in domains where extensive setup is nonexistent or impossible (i.e., disaster zones [3], etc.).

Security of these networks is of utter importance, and depends on context. Closed MANETS, in which only preauthorized actors can join, are easily protected. Previously distributed keys (out-of-band) can control access to the network, and be used for both integrity and confidentiality [4,5]. Other primitives can be setup to provide anonymity from external eavesdroppers as well. Open MANETS, however, pose a more difficult problem. Unidentified nodes joining and leaving at will coupled with the peer-to-peer routing paradigm, give an attacker a powerful set of tools for manipulation and tainting of data. It is for this context that we wish to provide confidentiality and integrity.

To accomplish this goal, we must construct a method for in-band key exchange (KE). Probably, the most famous in-band, no trusted-third-party (TTP) key exchange algorithm is the Diffie–Hellman (DH) algorithm [6,7], standardized in [8]. Under DH's threat model, it is possible to coordinate a symmetric key between two parties with an eavesdropping adversary not being able to deduce that key. It was soon discovered [9], however, that under a more prevalent threat model, where an adversary has the

ability to manipulate communication packets, a man-in-the-middle (MiTM) attack compromising the confidentiality of future data transmission can be accomplished. Eve can orchestrate two separate exchanges: one between Alice, the sender, and herself, and the other between herself and Bob, the receiver. Thus, all communication from Alice can be decrypted by Eve and re-encrypted for Bob's use. In this scenario, neither the sender nor the receiver are aware of the MiTM, and with the loss of integrity there is a loss of confidentiality as well. Anonymous DH was reduced to providing perfect forward secrecy (PFS—see Section 5.1.3) as a service to other authenticating protocols (i.e., SSL/TLS [10], SSH [11], IKE [12], and many others).

In this work, we wish to describe spraying Diffie–Hellman (SDH)—a technique for using anonymous DH—yet still provides security primitives. This is accomplished by utilizing the fluctuating topology of MANETs to combat attackers. Keying material is space sprayed or time sprayed over different, ever-fluctuating in-band paths, causing an active attacker to become passive; albeit, only a probabilistic guarantee is provided, not a provable one. Once keys have been established, spraying is forgone and direct communication commences.

Probabilistic security algorithms, known as Leap of Faith (LoF) algorithms, have been identified in [13] and discussed in [14]. Their purpose is not to guarantee security; rather, *raise the bar* for attackers, minimizing the attack surface to the original exchange. The spraying technique described in this work belongs to this category, and it joins many other protocols functioning within this group (e.g., SSH [11], BTNS for IPSEC [15], opportunistic HIP [16–18], and others).

The paper is organized as follows. Section 2 sets the stage for the entire paper. It includes the threat model, assumptions made, some definitions, and the simulations set-up that was used at different points throughout the paper. We then proceed, in Section 3, to describe space spraying, the algorithm, and the simulation results. Some claims and proofs on these results are also provided. The same is done in Section 4 for time spraying. A discussion of the results, comparison of the two methods, possible deficiencies, and other practical considerations are given in Section 5. In Section 6, we provide concluding words and future research directions.

Previous Work

Keeping keys secret is the basic requirement for cryptographic algorithms that wish to secure data transmission. As such, key coordination between communicating parties is the most vulnerable point, regardless if either a symmetric or asymmetric scheme is used. Therefore, secure key coordination is of utter importance in thwarting the many MiTM attacks centered around key exposure. Today, key coordination is based on a TTP (e.g., Kerberos [19]), on out-of-band coordination techniques (e.g., SSH [20]), on public key infrastructures (PKI) [21] (e.g., SSL [22] and TLS [10]), or on some combination of the above (e.g., IPSEC [23]).

For the MANET environment, where self-organization [24] is a basic premise, the network must be operated and managed by the nodes themselves [25]; hence, a static TTP is unsuitable for coordinating and managing keys. Solutions requiring pre-shared certificates, as in [4,26,27], cannot be used.

To provide key exchange, a number of alternative approaches have been introduced in literature. These include extensions of the PKI model via distributed certificate authorities (CAs) (e.g., shared secret cryptographic schemes [1,28–30] (for a survey see [31,32])), trust routing [33,34], and self-organized PKIs (e.g., use of side-channel key exchange [24,35,36] and general pairing techniques [37]). As one of the latter group we specifically reference ZRTP [38], a solution that provides protection against MiTM attacks, confidentiality, and authentication (if the signaling protocol provides end-to-end integrity protection) in-band. It uses a short authentication string (SAS), which users read and compare verbally during the first handshake, and ephemeral DH with hash commitment, to allow for future detection of a MiTM.

All methods have their advantages and downsides. Shared secrets have an inherent threshold, k , such that if the attacker(s) is able to compromise more than k shares, the system is insecure. There must also be some initial setup by some TTP to empower the first active members of the MANET [32]. Self-organized PKIs assume that trust is transitive [36]. For the aforementioned ZRTP, prior knowledge of the communicating parties voices must be known. The security is built upon the fact that each party

recognizes the other side's voice for verifying their authenticity. This protocol specifically targets voice data, and will not work for other types of data transfers. In addition, as stated in [39] and shown in [40], overcoming authentication and conducting a MiTM attack can be accomplished with voices spoofed with a synthesizer.

2. Preliminaries

2.1. Threat Model

The current work places no limitations on the attacker. Eavesdropping, injecting data into the communication stream or stopping the stream itself are feasible. Thus, for every route under her control, she can drop packets (DOS) or taint the data to initiate a MiTM attack.

We allow attackers to collude, bringing under their joint control multiple routes. For simplicity's sake, we will refer to all of the colluding parties as one entity.

2.2. Assumptions and Definitions

To allow for communication, the only prior knowledge assumed is the address of both parties. This address can be in any form: IP address, unique IMEI embedded in the device, or phone number allocated to the smart node. To avoid collisions, we assume that this address is unique and cannot be spoofed. To justify this assumption, we note that spoofing an address only influences some of the routing tables; not all of them. This allows for the correct delivery of packets not traveling through malicious nodes. Last, no trusted third party (TTP) is available for the coordination or authentication of sender and receiver.

Definition 1. Let $G = (V, E)$ be a topology graph of a MANET at a specific time, where $V = \{v_1, v_2, \dots, v_n\}$ are the hosts in the network, and $E = V \times V$ are the bidirectional communication links between these hosts.

Let $R = \{r_1, r_2, \dots, r_n\}$ be a noncyclic route ($r_i \neq r_j \forall r_i, r_j \in R$ and $(r_i, r_{i+1}) \in E$) in G , between the sender (r_1) and receiver (r_n). Let $|R|$ denote the length of the path.

Let \mathbb{R} denote the set of all possible such routes, $|\mathbb{R}|$ the size of \mathbb{R} , and R^c the set of chosen routes from \mathbb{R} ; implying, $R^c \subseteq \mathbb{R}$.

Let $A = \{a_1, a_2, \dots, a_w\}$, such that $A \subseteq \mathbb{R}$ and $\forall a_i \in A, \exists a_{i_p} \in a_i$, which is under the attacker's control. Let $|A|$ equal the size of A .

In essence, we define that it suffices for one host, a_{i_p} , to come under the attacker's control for the entire route, a_i , to be tainted.

Definition 2. Let $F = \{f_1, f_2, \dots, f_m\} = \mathbb{R} - A$. That is $F \cup A = \mathbb{R}$ and $F \cap A = \emptyset$, denoting all the paths not under the attacker's control. Let $|F|$ be the size of F .

Based on the above definition, we assume that the MANET is such that $|A| < |\mathbb{R}| \implies |F| > 0$, meaning that at least one route between the sender and receiver is not under the attacker's control. That the attacker has not taken over the entire network, for which no leap of faith algorithms can succeed.

2.3. Simulation Parameters

Prior to implementing our algorithm on a real-world MANET application (e.g., Serval [3]), we ran multiple preliminary simulations to test its feasibility. In this section, we describe our simulation model.

2.3.1. Random Number Generator

All simulations must have the means of generating random data. Of the many algorithms available (e.g., Mersenne Twister [41]), we chose the combined multiple recursive random number generator (CMRG) algorithm [42], an algorithm that passed a wide range of empirical tests of randomness [43]

and is the algorithm of choice in many simulation software (e.g., Arena [44,45]). Of course, similar results should be achieved using all good random number generators.

2.3.2. Mobility Model

The mobility of nodes in the real world exhibits vastly varying behaviors. Some walk, others drive. Some move about randomly, others in specific formations or patterns. Each of these have their own characteristics, and must be modelled accordingly (for a survey see [46]). Due to its prevalence (see, e.g., [47–50]), in this work, all simulations followed a random walk model, which captures the pattern of people in open spaces or recreational parks. We leave other models to future research.

2.3.3. Routing Algorithm

Network simulation requires that one choose the means by which packets are forwarded on the path to the destination. Being the basis behind OSPF [51], implementation of the Dijkstra [52] graph theory best route algorithm using standard dynamic programming techniques, allows us to find the optimal (shortest) route. We randomly chose between multiple paths having similar lengths, classifying one as the better of the bunch. All packets were sent through optimal paths when possible.

2.3.4. Attacker Dispersion

Attackers are randomly placed on the graph, allowing for all possible attacker dispersion scenarios. The number of attackers are a function of the size of the population, with a linear increase in the population entailing a similar increase in the number of attackers.

These attackers are assumed colluding. Knowledge gleaned by one attacker (e.g., intercept of a micro-KE message) and/or spoofing requirements are instantly shared with all others through external means.

2.3.5. Simulation Round

Each round was preset with a specific combination of parameters that we are checking for (see Sections 3.3 and 4.2). Next, we randomly chose some of the nodes in the network to represent colluding attackers; their specific number a function of predetermined parameter. Last, assuming there is a connection between sender and receiver, $|\mathbb{R}| > 0$, we sprayed the k micro-KE messages documenting whether all ($R^c \cap F \neq \emptyset$), none ($R^c \cap A \neq \emptyset$), or some ($R^c \cap A \neq \emptyset$ and $R^c \cap F \neq \emptyset$) of the messages were intercepted. Rounds for which $|\mathbb{R}| = 0$, were discarded.

The actual spraying algorithm was done using either random message spraying or even message spraying algorithm (see Section 3.2). Both techniques were executed on the same network graph so we can get comparable results.

2.3.6. Miscellaneous

All parameter combinations were simulated 10^5 times, and repeated 10 times (for a total of 10^6 simulation rounds) per parameter combination; allowing us to estimate the standard deviation. We recorded the number of times, per 10^5 , the attackers were able to completely intercept some, all or none of the micro-KE messages. This was done both for random spray and for even spray (see Section 3.2).

3. Space Spraying

The crux of the Rivest and Shamir's attack [9] on DH, blocks legitimate data coming from Alice and injects false alternate information for Bob. Utilization of the ever-changing network topology inherent to MANETs, allows us to revert to the original threat model; namely, remove the attackers ability of injecting rough data for Bob. Under this scenario, DH based confidentiality still holds.

To achieve this we notice that Eve cannot know, a priori, all of the paths between sender, Alice, and receiver, Bob; she cannot predict \mathbb{R} . By utilizing more than one path for KE, we increase

the probability that we chose a path $R_i \in (R^c \cap F)$, reducing Eve's capabilities of manipulating the KE. Only if, by chance, $R^c \subseteq A$, will she succeed in intercepting the entire KE, carrying out a full-scale MiTM attack.

3.1. Algorithm

3.1.1. KE Sending Protocol

Given a KE message, msg , that must be transferred from Alice to Bob, Alice must execute the following protocol steps.

- Alice appends a cryptographic hash (i.e., MD5 [53], SHA1 [54], SHA3 [55], etc.) to msg , creating $msg' = msg + h(msg)$. The purpose of this hash is for Bob to be able to confirm that he received all parts of msg (see next step) untainted.
- The derived msg' is then divided into k smaller parts, $msg_1, msg_2, \dots, msg_k$, such that
 1. each msg_i is composed of all $(c \cdot k) + i$ bits of msg' , where $0 \leq c \leq \left\lfloor \frac{|msg'|}{k} \right\rfloor$; e.g., msg_3 is composed of bits $3, (k+3), (2k+3), \dots$
 2. when $\frac{|msg'|}{k} - \left\lfloor \frac{|msg'|}{k} \right\rfloor = r > 0$, the remaining r bits not previously allocated are divided such that bit $\left(k \cdot \left\lfloor \frac{|msg'|}{k} \right\rfloor \right) + m$ is augmented to msg_m ; of course, $1 \leq m \leq r < k$.
- Each msg_i is to be sent through a different network route, $R_i \in R^c$, starting with Alice's immediate neighbors, and making its way to Bob in possibly independent paths.

3.1.2. KE Receiving Protocol

In order to reconstruct the message, Bob must:

- receive all k micro-messages (msg_i) so that he can recompose msg' .
- remove and compare the cryptographic hash $h(msg)$ received with one computed locally to check the integrity of the data.

Under these conditions, Eve must intercept all k micro-messages ($R^c \subseteq A$) so as to manipulate the key in such a way so a MiTM attack can be carried out. Even one micro-message escaping interception, i.e., $R^c \cap F \neq \emptyset$, will alert Bob of the possibility of the existence of a MiTM. Armed with this knowledge, appropriate measures can be taken (e.g., send a RESET message, alert the network of the problem, refuse to accept confidential information, or any other similar measures). This is functionally equivalent to Eve's capabilities being reduced to eavesdropping; tainting the data is not possible.

3.2. Spraying Methodology

The last step of the sending protocol (Section 3.1.1) describes the dispersion of msg' through different routes starting from Alice's adjacent neighbors. This dispersion (or spraying) is dependent upon the choice of routes for each msg_i from among available routes, \mathbb{R} . Choosing routes, $R^c \subseteq \mathbb{R}$, can be done by having Alice randomly select some path for each msg_i ("random spray"). As random choice allows for both msg_i and msg_j to be sent through the same $R_h \in R^c$, it follows that $|R^c| \leq k$ even when $k \leq |\mathbb{R}|$. Alternatively, she can evenly spread all micro-messages among available paths, picking as many distinct routes as possible ("even spray"). Hence, if $k \leq |\mathbb{R}|$ then $|R^c| = k$.

Intuitively, random spray should have a higher secure channel setup success rate. This can be explained by the fact that if there are any insecure routes, $A \neq \emptyset$, there is a higher chance that they will be included in R^c if one was forced to evenly spread micro-messages [56].

To illustrate the differences between these two schemes, we analyze a topology in which all paths from Alice to Bob are distinct. Such a topology can be defined such that $\forall R_i, R_j \in \mathbb{R}$ only $R_{i_1} = R_{j_1}$ and $R_{i_{|R_i|}} = R_{j_{|R_j|}}$. For all other hosts on each of the paths, $R_{i_d} \neq R_{j_s}$, where $2 \leq d < |R_i|$ and $2 \leq s < |R_j|$.

The probability that the communication will not be compromised (whether a secure channel is actually setup or an attack attempt is detected and thwarted) for random spray is given by

$$P = 1 - \left(\frac{|A|}{|\mathbb{R}|} \right)^k \quad (1)$$

This is based on the probability that at least one msg_i will get through some $f_j \in F$ to alert Bob as to the existence of MiTM ($R^c \cap F \neq \emptyset$).

For even spray, under the same topology scenario, the probability for noncompromise is

$$P = 1 - \begin{cases} 0 & k > |A| \\ \frac{\binom{|A|}{k} \binom{|F|}{0}}{\binom{|\mathbb{R}|}{k}} & k \leq |A| \end{cases} \quad (2)$$

As an example, we fix the parameters such that $k = 4$ (msg' is divided into four micro-messages), $|\mathbb{R}| = 6$ (Alice has six paths to choose from), of which $|A| = 2$ and $|F| = 4$. If we assume random spread, there is a

$$\left(\frac{4}{6} \right)^4 = \frac{256}{1296} = 19.75\% \quad (3)$$

chance of all micro-messages escaping capture, so that $R^c \cap A = \emptyset$, and (based on Equation (1)) 98.77% for any positive result ($R^c \cap F \neq \emptyset$).

For the same parameter values with even spread, the chances that $R^c \cap A = \emptyset$ drops to

$$\frac{\binom{2}{0} \binom{4}{4}}{\binom{6}{4}} = \frac{1 * 1}{15} = 6.67\% \quad (4)$$

Full compromise, requiring Eve to capture all micro-KE messages, cannot be achieved with even spread (i.e., $P_{noncompromise} = 1$, based on Equation (2)) for the given scenario. Random spread, however, would allow for a

$$\left(\frac{2}{6} \right)^4 = \frac{16}{1296} = 1.23\% \quad (5)$$

chance of being compromised.

The converse case, in which for the same six path options $|A| = 4$ and $|F| = 2$, would produce opposite results. There is a $\left(\frac{4}{6} \right)^4 = 19.75\%$ probability of being fully compromised with random spread, with a detection probability of 79%. This, compared to a probability of

$$\frac{\binom{4}{4} \binom{2}{0}}{\binom{6}{4}} = \frac{1 * 1}{15} = 6.67\% \quad (6)$$

for fully compromising the communication, and a high probability of

$$\frac{\binom{4}{3} \binom{2}{1}}{\binom{6}{4}} + \frac{\binom{4}{2} \binom{2}{2}}{\binom{6}{4}} = \frac{4 * 2}{15} + \frac{6 * 1}{15} = \frac{14}{15} = 93.33\% \quad (7)$$

detection of attack for even spread. Without random spread, however, success is impossible (i.e., $P_{noncompromise} = 0$). Therefore, although random spread increases the possibility of being attacked, it leaves the possibility of success open ($\approx 1.23\%$).

Of course, MANETs are self-evolving, and have ever-changing topologies. Paths are anything but distinct. They are bound to cross, join, or split anywhere between Alice and Bob. The only real choice Alice can make, however, is which path the micro-KE message will begin on. Therefore, we must turn to simulation in order to deduce the effectiveness of the spraying techniques.

3.3. Simulation Results

As alluded to in Section 2.3, we used 640 different value combinations to test the proposed methodology. These included the following,

- population size varied between 100 and 900 nodes in increments of 100;
- percent of colluding attackers, taking the values of 1%, 2%, 3%, 4%, and 5%;
- number of micro-KE messages, k ; instantiated to either 2, 3, 4, or 5; and
- the allowable communication distance fluctuated from $\frac{3}{10}$ of the network physical size to $\frac{6}{10}$ of that value.

The simulation outcomes were consistent with the hypothesis put forth by [57]; mainly, results that require all k micro-KE's to behave similarly, whether negative (i.e., complete intercept ($R^c \subseteq A$)) or positive (i.e., full success ($R^c \subseteq F$)) increase with random spray and decrease with even spray. This was true for all combinations tested. The converse case, in which we see better results with random spray when compared to even spray, was true for the intermediate situation where $0 < |R^c \cap A| < k$ (i.e., some, but not all, of the k micro-KE messages were intercepted). These results were accompanied with relatively low standard deviation values, which lends credibility to these conclusion.

Allowing for only population size to fluctuate, we observed that, although complete MiTM ($R^c \subseteq A$) decreases for both random and even spray ("randomBad" and "evenBad" in Figure 1, respectively), they seem to asymptotically level off at an equivalent value, the value of which is a function of the other parameters. That is, that the advantage of random spray decreases or is entirely lost as the population size increases (see first claim in Section 3.4).

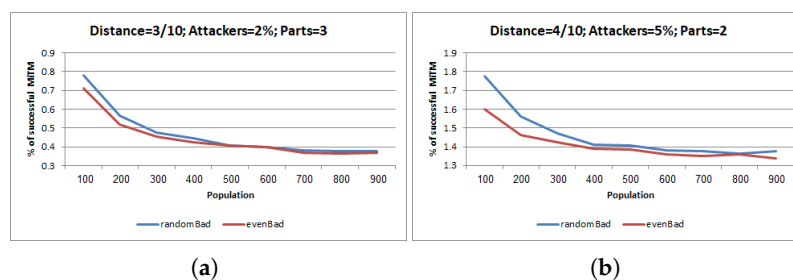


Figure 1. Fluctuating population. (a) Distance = 3/10; Attackers = 2%; Parts = 3. (b) Distance = 4/10; Attackers = 5%; Parts = 2.

We observed that for any given set of fixed parameters with only the percentage of attacker fluctuating, the probability that $R^c \subseteq A$ increases for both random and even spray, with random spray generating a marginally worse result. For the few instances where even spray appeared worse (higher on the graph), the results are close enough (well within one standard deviation) such that they can easily be attributed to simulation randomness. Some examples can be seen in Figure 2.

It does appear, however, that with higher attacker ratios the benefit of even spray increases. We propose that this is because, by evenly spreading the micro-KE message on multiple paths, the probability that $R^c \cap F \neq \emptyset$ increases. Albeit, with the direct cost of inducing a Denial-of-Service (DoS) attack. The probability that $R^c \cap A = \emptyset$ also decreases, implying at least some micro-KE parts will be captured preventing a complete KE. This can be seen in Figure 3, with "EvenDetect" and "RandomDetect" shown for even spread and random spread, respectively.

Increasing the number of micro-KE messages, with all other parameters remaining fixed, has a drastically positive effect on attack prevention (i.e., attacks decrease). The difference between random and even spread is barely distinguishable (with random spread performing slightly better), well within one standard deviation, which can be attributed to randomness. Examples are shown in Figure 4.

One cannot discern any major distinction between even and random spread (something outside one standard deviation) when fixing all parameters and allowing only the density of the population

to fluctuate (a sample of results can be seen Figure 5). There is, however, a noticeable decrease in the possibility of attack that comes with an increase in density, but that also levels off at some point.

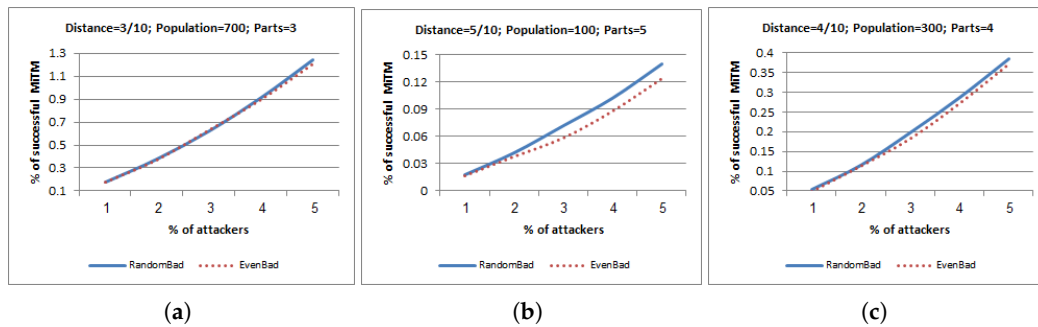


Figure 2. Fluctuating attacker as % of population. (a) Distance = 3/10; Population = 700; Parts = 3. (b) Distance = 5/10; Population = 100; Parts = 5. (c) Distance = 4/10; Population = 300; Parts = 4.

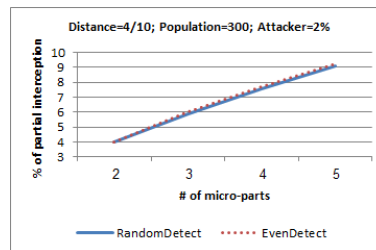


Figure 3. Number of connections at least one part was captured.

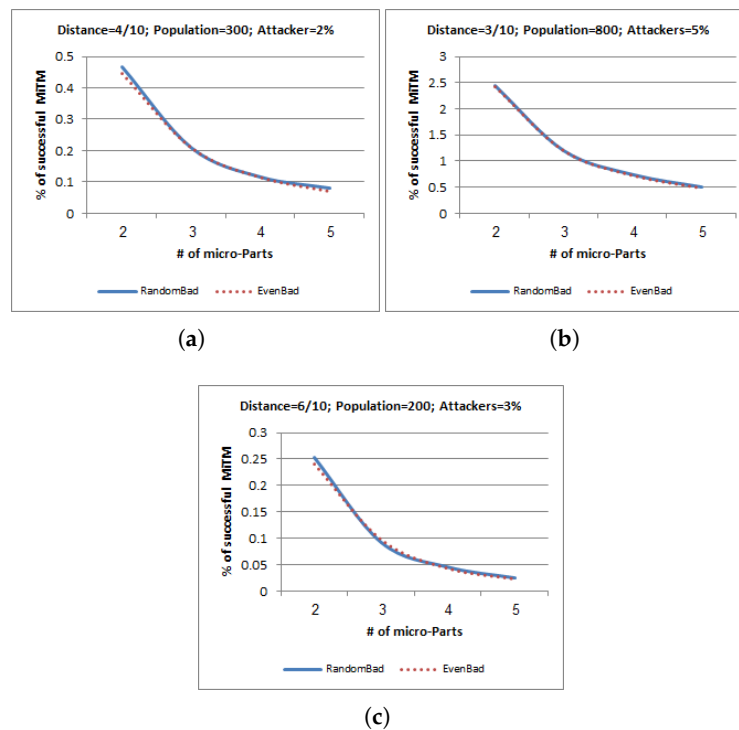


Figure 4. Fluctuating micro-KE parts. (a) Distance = 4/10; Population = 300; Attackers = 2%. (b) Distance = 3/10; Population = 800; Attackers = 5%. (c) Distance = 6/10; Population = 200; Attackers = 3%.

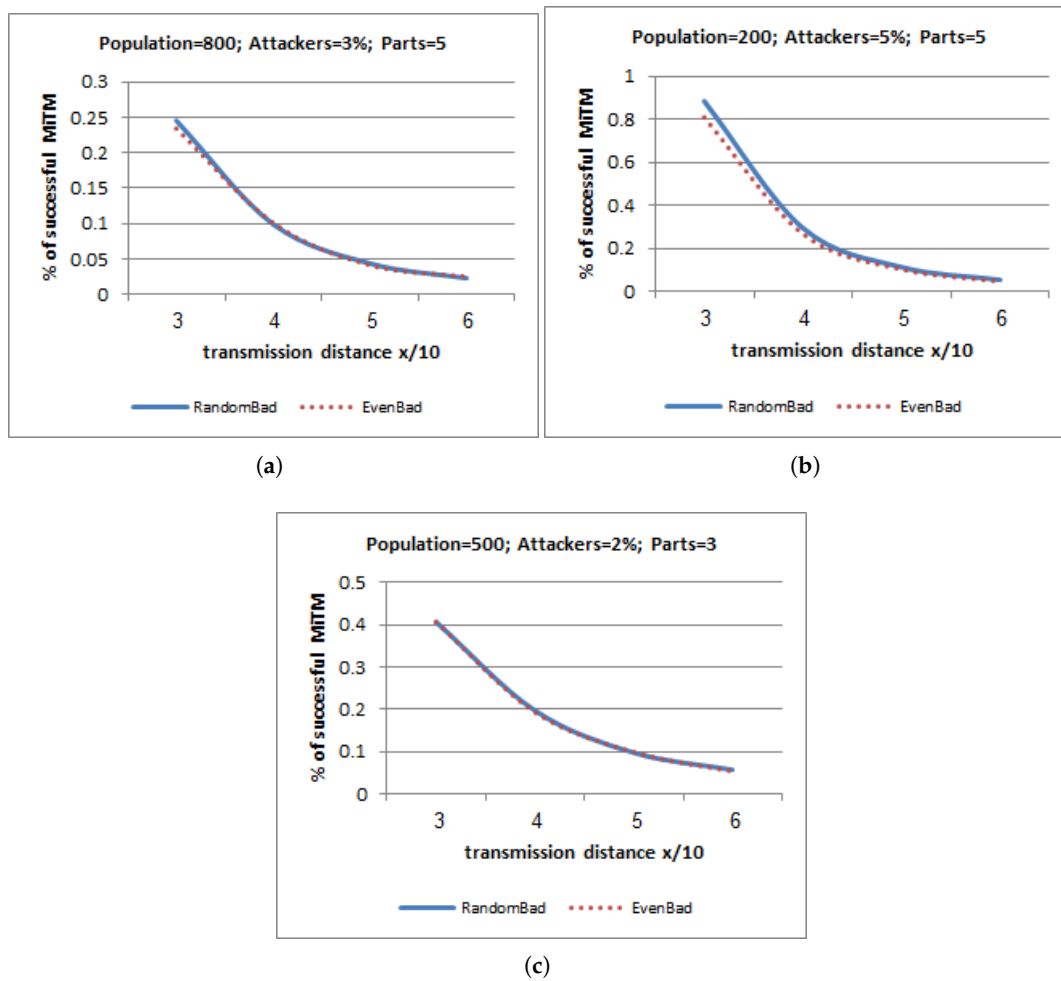


Figure 5. Fluctuating density. (a) Population = 800; attackers = 3%; Parts = 5. (b) Population = 200; attackers = 5%; Parts = 5. (c) Population = 500; attackers = 2%; Parts = 3.

3.4. Insights

The main factor causing random and even spray to differ is the ratio of micro-KE messages to available paths between Alice and Bob, $\frac{k}{|R|}$. Therefore, a change in network population size or network density (as a function of network physical area) would have a directly influence on the efficiency of the spraying methodologies.

This is easily explained by noticing that with an increase in network population or density, assuming random positioning, the expected number of neighbors each node has will increase as well. As a direct consequence, the probability that random spray would assign more than one micro-KE piece to any specific neighbor will decrease. As this happens, we would expect that random spray would start mimicking even spray with an ever increasing probability. With similar initial neighbor assignments for the micro-KE messages, similar final results are to be expected.

Claim 1. *The distinction between the two spraying methods would disappear with an increase in node population size.*

Proof. Let n be the number of nodes that are within range of the initial source. Let k be defined as the number of micro-KE message parts the original message will be broken up to.

Assume that $n \geq k$. Then, the number of equally likely ways of randomly assigning the k micro-KE messages to the available n nodes (random spray) is n^k .

The number of ways that the k message parts can be spread over the available n nodes, without any two or more parts being assigned to any one node (even spray), is the number of permutation of n objects taken k at a time:

$$P_{n,k} = \frac{n!}{(n-k)!} \quad (8)$$

Then, the probability of a random spray producing an even spray, $r \Rightarrow s$, is given by

$$P(r \Rightarrow s) = \frac{P_{n,k}}{n^k} = \frac{\frac{n!}{(n-k)!}}{n^k} \quad (9)$$

Equation (9) is known in the classical probability literature as the Birthday Problem [58].

For a fixed k , let us examine this probability as n gets larger.

Using Stirling's Formula [58], we calculate that our probability is approximately

$$P(r \Rightarrow s) \approx \frac{\frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{\sqrt{2\pi(n-k)} \left(\frac{n-k}{e}\right)^{n-k}}}{n^k} \quad (10)$$

manipulating Equation (10), we get

$$\begin{aligned} P(r \Rightarrow s) &\approx \sqrt{\frac{n}{n-k}} \left(\frac{1}{e}\right)^k \frac{n^{n-k}}{(n-k)^{n-k}} \\ &= e^{-k} \left(\frac{n}{n-k}\right)^{n-k+0.5} \end{aligned}$$

which is clearly an increasing function in n for a given fixed k .

Continuing, we get

$$\begin{aligned} P(r \Rightarrow s) &\approx e^{-k} \left(\frac{1}{1-\frac{k}{n}}\right)^n \left(\frac{n}{n-k}\right)^{-k+0.5} \\ &= e^{-k} \left(1 - \frac{k}{n}\right)^{-n} \left(\frac{n}{n-k}\right)^{-k+0.5} \end{aligned} \quad (11)$$

Taking limits of Equation (11), we get

$$\begin{aligned} \lim_{n \rightarrow \infty} P(r \Rightarrow s) &= e^{-k} \lim_{n \rightarrow \infty} \left(1 - \frac{k}{n}\right)^{-n} \lim_{n \rightarrow \infty} \left(\frac{n}{n-k}\right)^{-k+0.5} \\ &= e^{-k} e^k 1 = 1 \end{aligned} \quad (12)$$

Thus as n gets larger, we would expect the results for both random and even spray to asymptotically approach each other. \square

Claim 2. A decrease in the network area with the associated increase in node density would cause both spraying methods to behave similarly.

Proof. An increase in network density would give the sender more direct neighbors to choose from. Using the same analysis of previous claim, random spray should start mimicking even spray, eradicating any discernible difference between them. \square

These results are clearly observed in the simulation results of Figures 1 and 5 for first and second claims, respectively.

4. Time Spraying

As an alternative to spraying micro-KE messages over the network (space spraying of Section 3), it is possible to spread them over time. By refraining from transmitting msg_{i+1} immediately after msg_i , we reach the same result of having the network evolve enough so that attackers are circumvented by the new routes created. To carry out a complete MiTM, for each $f_m \in F$ chosen for msg_i , f_m must be an element of A at the time msg_i was sent. This greatly increases the difficulty for an attacker to succeed.

4.1. Algorithm

4.1.1. KE Sending Protocol

The sending protocol is similar to what is described in Section 3.1.1 for space spraying, with the third step replaced by the following step.

- Each msg_i is to be sent though a random optimal path starting with time t , and with a delay of Δ between them. Thus, msg_1 is sent at time t , msg_2 is sent at time $t + \Delta$; in general, msg_i is sent at time $t + (i - 1)\Delta$. For more information regarding Δ , see Section 4.2.

4.1.2. KE Receiving Protocol

The receiving is similar to what was described for space spraying in Section 3.1.2.

As in space spraying, Eve must intercept all k micro-messages (msg_i) so as to change the data. Bob will be alerted to the possible MiTM if even one micro-message escapes interception, if $F \cap R^c \neq \emptyset$; this is functionally equivalent to Eve's capabilities being reduced to eavesdropping, losing the ability of tainting the data.

4.2. Simulation Results

To simulate time spraying the following parameters were passed to NS3 [59], a well-known network simulation tool:

- population size varied from 250 to 500 in increments of 50;
- colluding attacker were set at either 3%, 4%, or 5% of population size;
- number of micro-KE messages, k ; msg' was split into 3, 4, or 5 parts;
- the time between micro-KE messages, Δ , was fixed at 4 s; and
- the transmission strength of nodes was set to the default provided by NS3.

As seen in Figure 6, the density of the nodes has direct positive influence on the success off an attacker. Regardless of whether we used three micro-messages (Figure 6a), four micro-messages (Figure 6b), or five micro-messages (Figure 6c), and whether they were 3%, 4%, or 5% attackers, the overall trend was similar. Albeit, the overall results of the four and five micro-messages are an order of magnitude better than three micro-messages.

To investigate the cause of the huge jump between three and four micro-messages, we decreased the number of messages ($k = 2$) and increased Δ (to 20 s). In Figure 7, one can clearly see that the success MiTM rate has decreased to the same order of magnitude as for the four and five micro-messages of Figure 6.

This leads us to the conclusion that it is not the number of micro-messages that has the major positive effect on time spraying success. Rather, it is the amount of time allowed to lapse between each of the micro-KE messages that has the required effect.

This result can be attributed to the rate of topology change before a contaminated optimal path becomes obsolete. At the walking speed of 1–2 m/s, it takes time for the topology to change, and exploiting that time frame is required. For other, faster-changing networks (e.g., VANETs), the Δ value will be much smaller.

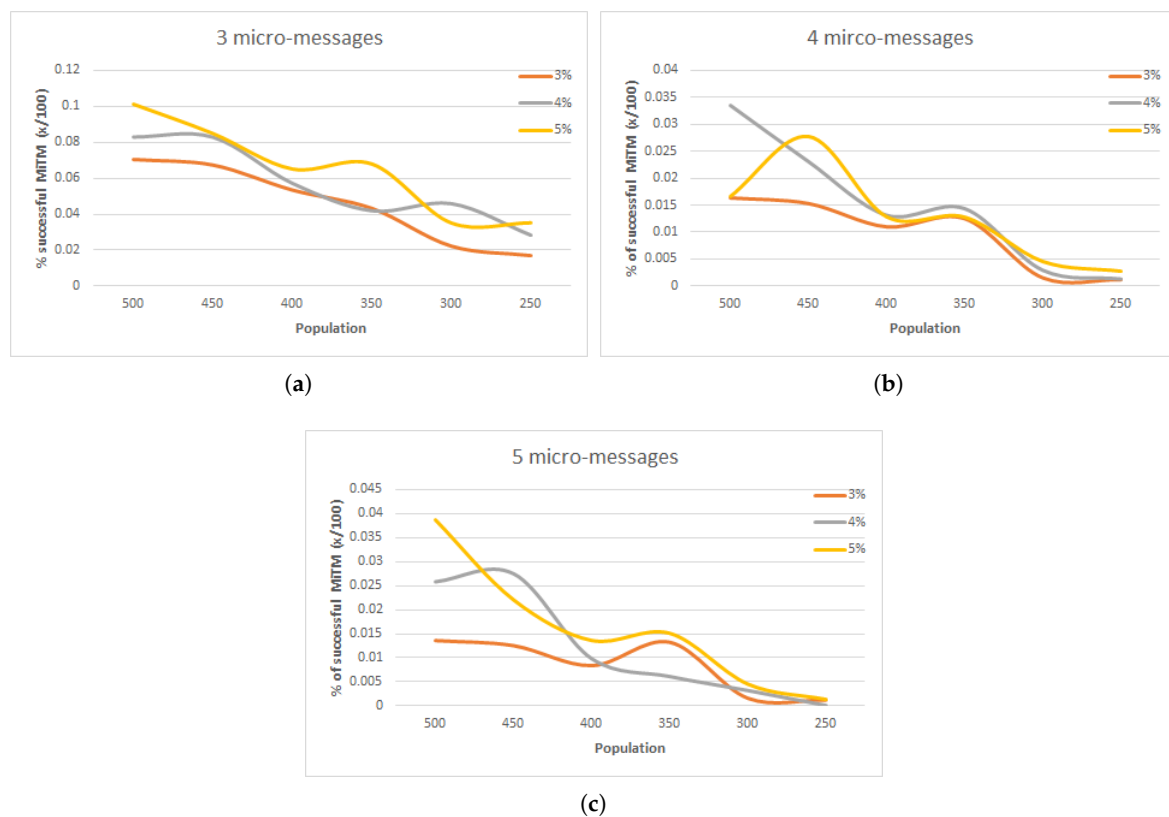


Figure 6. Percentage of successful MiTM attacks. (a) Exchanging key info in 3 micro-messages; (b) Exchanging key info in 4 micro-messages; (c) Exchanging key info in 5 micro-messages

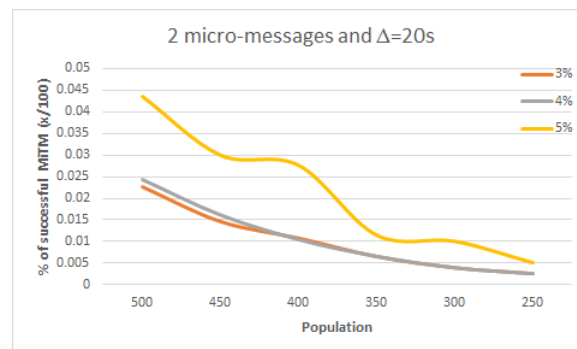


Figure 7. Percentage of successful MiTM when Δ is increased.

5. Discussion

5.1. Benefits of Spraying

Spraying KE remedies some of the deficiencies in related MANET algorithms. This includes required prior knowledge, user intervention, and the lack of forward security.

5.1.1. Prior Knowledge

A key issue when selecting a protocol for a specific architecture is the prior knowledge each communicating party is required to have. On the one extreme, pre-shared key (PSK) protocols, e.g., [60], or pre-sharing of certificates, e.g., [4,5], require that the keying material be correlated by-hand out-of-band. This scheme is perfectly suitable for VPNs where parties meet and setup a network. The same is true for trust PKIs (or pairing techniques, in general), unless trust is allowed to be transitive. When parties

have never previously met, as in our model, (e.g., *A* tells *B* about *C*), one cannot count on *B* and *C* having pre-shared keying material for communication exchange; therefore, this scheme is excluded.

The same is true for ZRTP. Using SAS for authentication requires that both parties recognize each other prior to session setup. Therefore, a referral system is precluded. This, besides the voice spoofing techniques stated in [39] and shown in [40], overcomes authentication.

SDH can utilize the fluctuation in the network topology to detect, with high probability, the existence of a MiTM attack; terminating the connection if needed. This, without prior cooperation between communicating parties.

5.1.2. User Intervention

Requiring a user to actively participate in the security of a protocol or application, is widely accepted as a problem. For most humans, usability is of uttermost importance, and trusting them with a product's security merely means that they are less likely to do so.

In ZRTP, for example, for every new conversation the voices of both parties must be recognized (via the SAS mechanism) for the shared secret to be generated and authenticated. Device pairing techniques require a manual, conscious operation, with many people naturally inclined not to do so. On the contrary, sooner or later it gets viewed as a burden, and done away with. This, obviously, totally voids the ability of these protocols to detect MiTM.

For spraying, user intervention is not required. The protocol detects MiTM automatically, inducing preventive actions as seen fit. Override, of course, is also viable if the user so deems, but this is external to the protocol and can be acceptable in certain situations.

5.1.3. Forward Secrecy

Session keys, and by extension the data they protect, can remain confidential even in the event that long-term keying material is compromised. This is known as perfect forward secrecy (PFS) [26], and is exhibited (and normally implemented) by ephemeral DH. The independent generation of the symmetric, unique, one-time key during the handshake by each of the parties, allows them to refrain from transmitting it over the communication medium. Therefore, each session must be cracked separately, and compromising one session does not influence the others. Under these definitions, SDH adheres to PFS, as it reduces to the ephemeral DH algorithm.

5.2. Cost and Shortcomings of SDH

5.2.1. Overhead

All features, be it security, authenticity, reliability, etc., regardless of effectiveness, come at some cost. SDH is no different, and comes with two cost factors: size of handshake data and length of paths taken.

As *msg* is divided into multiple packets, the optimum packet size is not used. As the same amount of payload is sent, the overhead increases in a linear fashion as a direct function of *k*.

For example, suppose we had an overhead of

$$\rho = \frac{\text{headers}}{\text{data} + \text{headers}} \quad (13)$$

then, by splitting *msg* into *k* micro-messages, we get an increase by a factor of *k*

$$\rho' = \frac{k * \text{headers}}{\text{data} + k * \text{headers}} \quad (14)$$

as the overall data sent is constant, but each *msg_i* has its own header.

The difference between Equations (14) and (13), $\rho' - \rho$, is the direct additional overhead incurred, but there are indirect costs as well. As multiple paths are used for each *msg_i*, only one would actually take the known optimal path between communicating parties. All others are forced into possibly

suboptimal paths, adding to the total work incurred by the protocol. Of course, there is a direct correlation between the level of security achieved and the overhead incurred. Choosing a large k for dividing msg , allows for increasing $|R^c|$. We leave exact calculations for the optimal k as a function of $|R|$ to future research.

5.2.2. Bottlenecks

Vulnerable convergence points around the communicating parties is the basic premise behind SDH; hence, the spraying embedded in the algorithm. It is perfectly reasonable, however, that topological bottlenecks exist, creating additional vulnerability points on the transmission path. The convergence of all paths to a single link or node, creates a convenient location for the attacker to compromise the network. As a simple such scenario, imagine x inter-city connection points, through which all traffic must pass when traversing from one city to the next. These x points constitute a vulnerability, regardless of the multiple paths SDH started with.

In [61,62], a linear algorithm is proposed to discover bottlenecks in an OLSR [63]-based MANET. This is quite useful, but must be further extended for other routing protocols as well.

5.2.3. Single Side Mitigation

SDH exhibits an interesting phenomenon: single side mitigation. Spraying is done on the sender's side, without any further control over which paths any msg_i will take once sent. As all msg_i 's are targeting the same destination, it is quite possible that paths will converge long before it is reached. This lends to the observation that although an attacker close to the sender is neutralized, an attacker near the receiver, however, will still have the ability intercept communications with much less effort. Thus, it would seem that the algorithm's security properties are influenced by Eve's position.

This, however, might not be a major problem. As all KE algorithms (including DH) require the receiver to respond with some unforeseen data, the receiver simply needs to employ the same spraying technique on its side to neutralize potential local attackers. Now, for an attack to succeed, we must have colluding attackers strategically placed near two moving targets; a feat difficult to execute "on the fly".

We leave the investigation of this phenomena ("single side mitigation") to future research.

5.2.4. Availability

The benefit achieved by SDH negatively effects DoS. With even one micro-KE message captured, a handshake re-start is required. As previously stated (Section 3.3), even spray is more prone to this problem than random spray. Therefore, it is a conscious decision that an implementer (or user) will need to make; whether availability or confidentiality and integrity is of higher importance.

In addition, having multiple micro-messages sent for each handshake packet on unreliable links, increases the probability that some will naturally get dropped or lost. This further reduces the availability of the system. We believe, however, that as SDH is only executed during the handshake process, which is almost instantaneous (with all other transmissions being routed normally); this deficiency should not greatly impact availability. We leave the exact drop rates for future experimentation.

5.2.5. Other Shortcomings

As SDH is a LoF algorithm, it cannot guarantee authentication; rather, it hardens the existing infrastructure, complicating an attacker's task. This implies that one is not guaranteed as to the identity of this communication partner. We assumed that by fixing network addresses, we can circumvent network routing attacks. With routing information prevalent and constantly updated by all, this assumption is mostly true. The probability of being able to taint everyone's information, all of the time, is small. Theoretically speaking, however, if an attacker were able to do that, the SDH does not provide authenticity. This, however, is the basic premise of all LoF algorithms, and is not a deficiency of SDH specifically. To achieve authentication of parties as well, we must incorporate mechanisms that will operate above the actual connection. We leave that to further research.

5.3. Comparison of Space vs. Time

Both space and time spraying generally possess the same positive characteristics; namely, there is no need for prior knowledge, user intervention is not required, the property of PFS, and they can be used for all types of data communications. Similarly, they suffer from similar shortcomings, including overhead costs, the lack of authentication, and prone to DOS attacks. Space or time spraying do, however, differ on two points: bottlenecks and user patience.

5.3.1. Bottlenecks

Space spraying, as mentioned in ([64], §4.2), is weakened when a bottleneck exists in the network topology. Since these bottle necks require all traffic between parties to pass through these nodes, an attacker can compromise the connection. Although time spraying suffers from the same weakness, it imposes another requirement on an attacker attempting to situate herself as a bottleneck. She must retain this vulnerable position over time. Being a bottle neck for some of the KE-micro messages does not constitute a vulnerability. As KE-micro messages disperse over time, retaining hold of the bottle neck for some of the handshake further reduces attack likelihood. Of course, it is possible that the attacker can retain hold of the vulnerable node over time (e.g., were only *x-intercity* connection points exist), but it is still an improvement over space spraying, which is vulnerable to temporary bottlenecks as well.

5.3.2. User Patience

Users are quite impatient. They expect a response time of less than two seconds (61%) and are willing to tolerate a response time of four seconds (49%) [65]. Thus, waiting 20 s for a connection, although marginally acceptable today might be unacceptable in the future. Although users are willing to wait when they perceive that an operation should take time, connection times are not within this scope. Users have accustomed to almost instant connections and might not be willing to forgo this privilege.

For MANETs based on people's movements, we do not see MANET topology fluctuation rate increase in the future. Therefore, time spraying contains an inherent flaw that might inhibit its acceptance in the long run; it requires time. There are futuristic MANETs, however, that might contain the property of quick topology change (e.g., vehicular ad hoc networks, also known as VANETs). For those networks time spraying might be viable.

5.4. Practical Considerations

There are numerous practical considerations that have to be taken into account before our proposal can be fully implemented. For one, SDH is not a communication protocol, but rather a security add-on. Therefore, the specific details of a pre-existing protocol's handshake must be hammered out. Some already have DH exchange embedded within, making our technique just a communication tweak. Others, might need an additional, transparent layer beneath, or prior to, the handshake, to use SDH. Still, others might want to add SDH to the actual handshake, so as to take advantage of the technique suggested.

Another practical issue is with the movement of the network nodes within a given time frame. For the purpose of the simulations, it was assumed—at least for space spraying—that the delivery of the micro-KE is instant; i.e., the topology does not fluctuate during this time frame. This, of course, might not be the case. A link might disappear during the spray, causing some micro-KE message to get lost. To alleviate this problem, either timers or some other backup mechanism (e.g., secret sharing techniques [1,29]) must be introduced into the system. These types of problems are out of the scope of this paper, and are left to future research.

6. Conclusions and Future Research

In this paper, we presented a technique for hardening an anonymous leap of faith KE handshake. This is accomplished by either spraying keying material over network space or over time, allowing for an automated multipath transfer as a consequence of network topology fluctuation. The resulting effect is in “raising the bar” for attackers before they can orchestrate a full MiTM attack.

The novelty of the work can be summarized as accomplishing KE in line, without prior knowledge or some external trusted third party (TTP, CA or otherwise). It allows the MANET to be open to all and self-configuring, as expected. This is in stark contrast to other previous solutions in which at least one of these conditions is not met.

As pointed to in the paper, cost estimation, single side mitigation, and routing guarantees require more work. Also, incorporation of these techniques into running scenarios for extraction of real-time data would be beneficial. These issues coupled with the implementation with specific protocols is left for future research.

Last, we note that the results shown might be a function of the mobility model used (see Section 2.3.2), with other models possibly achieving different results. We specifically note VANETs in which the average speed does not match that of walking people, allowing for a smaller Δ (see Section 4.2). We leave investigation of these topics to future research as well.

Author Contributions: Conceptualization, A.S. (Ariel Stulman); Data curation, A.S. (Ariel Stulman); Formal analysis, A.S. (Alan Stulman); Methodology, A.S. (Ariel Stulman) and A.S. (Alan Stulman); Writing—original draft, A.S. (Ariel Stulman).

Funding: This research received no external funding.

Acknowledgments: The authors would like to acknowledge Jonathan Lahav and Avi Shmueli for their insights, thoughts and ideas that helped bring this research forward. In addition, much gratitude to Omer Achreck for running many of the simulations.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|-------|---------------------------|
| MANET | Mobile Ad-hoc NETworks |
| IOT | Internet Of Things |
| VANET | Vehicular Ad-hoc NETworks |
| KE | Key Exchange |
| DH | Diffie–Hellman |
| SDH | Spraying DH |
| LOF | Leap Of Faith |

References

1. Zhou, L.; Haas, Z.J. Securing ad hoc networks. *IEEE Netw.* **1999**, *13*, 24–30.10.1109/65.806983. [CrossRef]
2. AllJoyn. 2012. Available online: <https://openconnectivity.org/developer/reference-implementation/alljoyn> (accessed on 29 August 2019).
3. The Serval Project. Available online: <http://www.servalproject.org/> (accessed on 29 August 2019).
4. Adjih, C.; Clausen, T.; Jacquet, P.; Laouiti, A.; Muhlethaler, P.; Raffo, D. Securing the olsr protocol. In Proceedings of the Med-Hoc-Net, Mahdia, Tunisia, 25–27 June 2003; pp. 25–27.
5. Chen, T.; Mehani, O.; Boreli, R. Trusted routing for vanet. In Proceedings of the 2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), Lille, France, 20–22 October 2009; pp. 647–652.
6. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
7. Hellman, M.; Diffie, W.; Merkle, R. Cryptographic Apparatus and Method. U.S. Patent 4,200,770, 29 April 1980.

8. Rescorla, E. Diffie–Hellman Key Agreement Method. RFC 2631 (Proposed Standard) (1999). Available online: <http://www.ietf.org/rfc/rfc2631.txt> (accessed on 29 August 2019).
9. Rivest, R.L.; Shamir, A. How to expose an eavesdropper. *Commun. ACM* **1984**, *27*, 393–394. [[CrossRef](#)]
10. Dierks, T.; Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard) (2008). Available online: <http://www.ietf.org/rfc/rfc5246.txt> (accessed on 29 August 2019).
11. Ylonen, T.; Lonvick, C. The Secure Shell (SSH) Transport Layer Protocol. RFC 4253 (Proposed Standard) (2006). Available online: <http://www.ietf.org/rfc/rfc4253.txt> (accessed on 29 August 2019).
12. Kaufman, C. Internet Key Exchange (IKEv2) Protocol. RFC 4306 (Proposed Standard) (2005). Available online: <http://www.ietf.org/rfc/rfc4306.txt> (accessed on 29 August 2019).
13. Arkko, J.; Nikander, P. Weak authentication: How to authenticate unknown principals without trusted parties. In *Security Protocols*; Christianson, B., Crispo, B., Malcolm, J.A., Roe, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 5–19.
14. Pham, V.; Aura, T. Security analysis of leap-of-faith protocols. In *Security and Privacy in Communication Networks*; Rajarajan, M., Piper, F., Wang, H., Kesidis, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 337–355.
15. Touch, J.; Black, D.; Wang, Y. Problem and Applicability Statement for Better-Than-Nothing Security (BTNS). RFC 5387 (Informational) (2008). Available online: <http://www.ietf.org/rfc/rfc5387.txt> (accessed on 29 August 2019).
16. Gurtov, A. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*; John Wiley & Sons: Hoboken, NJ, USA, 2008; Volume 21.
17. Henderson, T.; Heer, T.; Jokela, P.; Moskowitz, R. Host Identity Protocol Version 2 (hipv2); 2015. Available online: <https://datatracker.ietf.org/doc/rfc7401/> (accessed on 8 October 2019)
18. Moskowitz, R.; Heer, T.; Jokela, P.; Henderson, T.R. Host Identity Protocol Version 2 (HIPv2). RFC 7401 (2015). Available online: <https://rfc-editor.org/rfc/rfc7401.txt> (accessed on 29 August 2019).
19. Neuman, C.; Yu, T.; Hartman, S.; Raeburn, K. The Kerberos Network Authentication Service (V5). RFC 4120 (Proposed Standard) (2005). Available online: <http://www.ietf.org/rfc/rfc4120.txt> (accessed on 29 August 2019).
20. Ylonen, T.; Lonvick, C. The Secure Shell (SSH) Protocol Architecture. RFC 4251 (Proposed Standard) (2006). Available online: <http://www.ietf.org/rfc/rfc4251.txt> (accessed on 29 August 2019).
21. Cooper, D.; Santesson, S.; Farrell, S.; Boeyen, S.; Housley, R.; Polk, W. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard) (2008). Available online: <http://www.ietf.org/rfc/rfc5280.txt> (accessed on 29 August 2019).
22. Freier, A.; Karlton, P. The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Proposed Standard) (2011). Available online: <http://www.ietf.org/rfc/rfc6101.txt> (accessed on 29 August 2019).
23. Kent, S.; Seo, K. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard) (2005). Available online: <http://www.ietf.org/rfc/rfc4301.txt> (accessed on 29 August 2019).
24. Čapkun, S.; Hubaux, J.P.; Buttyan, L. Mobility helps peer-to-peer security. *IEEE Trans. Mob. Comput.* **2006**, *5*, 43–51. [[CrossRef](#)]
25. Buttyán, L.; Hubaux, J.P. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mob. Netw. Appl.* **2003**, *8*, 579–592. [[CrossRef](#)]
26. Diffie, W.; Van Oorschot, P.C.; Wiener, M.J. Authentication and authenticated key exchanges. *Des. Codes Cryptogr.* **1992**, *2*, 107–125. [[CrossRef](#)]
27. O’Higgins, B.; Diffie, W.; Strawczynski, L.; de Hoog, R. Encryption and isdn—A natural fit. In Proceedings of the 1987 International Switching Symposium, Phoenix, AZ, USA, 15–20 March 1987.
28. Joshi, D.; Namuduri, K.; Pendse, R. Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: An analysis. *EURASIP J. Wirel. Commun. Netw.* **2005**, *2005*, 579–589. [[CrossRef](#)]
29. Kong, J.; Petros, Z.; Luo, H.; Lu, S.; Zhang, L. Providing robust and ubiquitous security support for mobile ad hoc networks. In Proceedings of the Ninth International Conference on Network Protocols, Riverside, CA, USA, 11–14 November 2001; pp. 251–260.
30. Wu, B.; Wu, J.; Fernandez, E.B.; Ilyas, M.; Magliveras, S. Secure and efficient key management in mobile ad hoc networks. *J. Netw. Comput. Appl.* **2007**, *30*, 937–954. [[CrossRef](#)]
31. Hegland, A.M.; Winjum, E.; Mjolsnes, S.F.; Rong, C.; Kure, O.; Spilling, P. A survey of key management in ad hoc networks. *Commun. Surv. Tutor.* **2006**, *8*, 48–66. [[CrossRef](#)]
32. Merwe, J.V.D.; Dawoud, D.; McDonald, S. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Comput. Surv.* **2007**, *39*. [[CrossRef](#)]

33. Leligou, H.C.; Trakadas, P.; Maniatis, S.; Karkazis, P.; Zahariadis, T. Combining trust with location information for routing in wireless sensor networks. *Wirel. Commun. Mob. Comput.* **2012**, *12*, 1091–1103. [CrossRef]
34. Yu, Y.; Li, K.; Zhou, W.; Li, P. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *J. Netw. Comput. Appl.* **2012**, *35*, 867–880. [CrossRef]
35. Balfanz, D.; Smetters, D.K.; Stewart, P.; Wong, H.C. Talking to strangers: Authentication in ad hoc wireless networks. In Proceedings of the the Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA, 6–8 February 2002.
36. Čapkun, S.; Hubaux, J.P.; Buttyán, L. Mobility helps security in ad hoc networks. In Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, MobiHoc '03, Annapolis, MD, USA, 1–3 June 2003; pp. 46–56.
37. Mirzadeh, S.; Cruickshank, H.S.; Tafazolli, R. Secure device pairing: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 17–40. [CrossRef]
38. Zimmermann, P.; Johnston, A.; Callas, J. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189 (Proposed Standard) (2011). Available online: <http://www.ietf.org/rfc/rfc6189.txt> (accessed on 29 August 2019).
39. ERK. On the Security of Short Authentication Strings. Available online: <http://www.imc.org/ietf-rtpsec/mail-archive/msg00608.html> (accessed on 7 September 2012).
40. Cryptologic Quarterly. 2007. Available online: <https://www.nsa.gov/about/contact-us/#subject:history> (accessed on 8 October 2019).
41. Matsumoto, M.; Nishimura, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **1998**, *8*, 3–30. [CrossRef]
42. L'Ecuyer, P. Combined multiple recursive random number generators. *Oper. Res.* **1996**, *44*, 816–822. [CrossRef]
43. L'ecuyer, P. Good parameters and implementations for combined multiple recursive random number generators. *Oper. Res.* **1999**, *47*, 159–164. [CrossRef]
44. Automation, R. Arena Simulation Software. Available online: <https://www.arenasimulation.com> (accessed on 29 August 2019).
45. L'Ecuyer, P. Software for uniform random number generation: Distinguishing the good and the bad. In Proceedings of the 33rd Conference on Winter Simulation, Arlington, VA, USA, 9–12 December 2001; pp. 95–105.
46. Camp, T.; Boleng, J.; Davies, V. A survey of mobility models for ad hoc network research. *Wirel. Commun. Mob. Comput.* **2002**, *2*, 483–502. [CrossRef]
47. Kohls, M.; Hernandez, T. Expected Coverage of Random Walk Mobility Algorithm. *arXiv* **2016**, arXiv:1611.02861.
48. Nayak, P.; Sinha, P. Analysis of random way point and random walk mobility model for reactive routing protocols for MANET using NetSim simulator. In Proceedings of the 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), Kota Kinabalu, Malaysia, 2–4 December 2015; pp. 427–432.
49. Schweitzer, N.; Stulman, A.; Margalit, R.D.; Shabtai, A. Contradiction based gray-hole attack minimization for ad hoc networks. *IEEE Trans. Mob. Comput.* **2016**, *16*, 2174–2183. [CrossRef]
50. Schweitzer, N.; Stulman, A.; Shabtai, A.; Margalit, R.D. Mitigating Denial of Service Attacks in OLSR Protocol Using Fictitious Nodes. *Mob. Comput. IEEE Trans.* **2016**, *15*, 163–172. [CrossRef]
51. Moy, J. OSPF Version 2. RFC 2328 (Standard) (1998). Available online: <http://www.ietf.org/rfc/rfc2328.txt> (accessed on 29 August 2019).
52. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
53. Rivest, R. The MD5 Message-Digest Algorithm. RFC 1321 (Informational) (1992). Available online: <http://www.ietf.org/rfc/rfc1321.txt> (accessed on 29 August 2019).
54. Lilly, G.M. *Device for and Method of One-Way Cryptographic Hashing*; 2004. Available online: <https://patents.google.com/patent/US20020122554> (accessed on 9 October 2019).
55. Dworkin, M.J. Sha-3 Standard: Permutation-Based Hash and Extendable-Output Functions (2015). Available online: <http://dx.doi.org/10.6028/NIST.FIPS.202> (accessed on 2 August 2019).
56. Stulman, A.; Stulman, A. Spraying techniques for securing key exchange in large ad hoc networks. In Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Cancun, Mexico, 2–6 November 2015; pp. 29–34.

57. Stulman, A.; Lahav, J.; Shmueli, A. Manet secure key exchange using spraying diffie-hellman algorithm. In Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions, London, UK, 10–12 December 2012; pp. 249–252.
58. Feller, W. *An Introduction to Probability Theory and Its Applications*, 2nd ed.; Wiley: Hoboken, NJ, USA, 1968; Volume 1.
59. The ns-3 Simulator. Available online: <http://www.nsnam.org> (accessed on 29 August 2019).
60. Cusack, F.; Forssen, M. Generic Message Exchange Authentication for the Secure Shell Protocol (SSH). RFC 4256 (Proposed Standard) (2006). Available online: <http://www.ietf.org/rfc/rfc4256.txt> (accessed on 29 August 2019).
61. Schweitzer, N.; Stulman, A.; Hirst, T.; Margalit, R.; Armon, M.; Shabtai, A. Detecting bottlenecks on-the-fly in olsr based manets. In Proceedings of the 2014 IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI), Eilat, Israel, 3–5 December 2014.
62. Schweitzer, N.; Stulman, A.; Hirst, T.; Margalit, R.; Shabtai, A. Network bottlenecks in OLSR based ad hoc networks. *Ad-Hoc Netw.* **2019**, *88*, 36–54. [CrossRef]
63. Clausen, T.; Jacquet, P. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental) (2003). Available online: <http://www.ietf.org/rfc/rfc3626.txt> (accessed on 29 August 2019).
64. Stulman, A.; Lahav, J.; Shmueli, A. Spraying diffie-hellman for secure key exchange in manets. In *Security Protocols Workshop*; Lecture Notes in Computer Science; Christianson, B., Malcolm, J.A., Stajano, F., Anderson, J., Bonneau, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8263, pp. 202–212.
65. Failing to Meet Mobile App User Expectations: A Mobile User Survey. Dimentional Research (2015). Available online: https://techbeacon.com/sites/default/files/gated_asset/mobile-app-user-survey-failing-meet-user-expectations.pdf (accessed on 29 August 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).