Towards Unconditional Tor-Like Anonymity

Iris Safaka, László Czap, Katerina Argyraki
EPFL, Switzerland

Christina Fragouli
EPFL & UCLA

***Abstract*—We design and evaluate a traffic anonymization protocol for wireless networks, aiming to protect against computationally powerful adversaries. Our protocol builds on recent key-generation techniques, that leverage intrinsic properties of the wireless together with standard coding techniques. We show how to exploit the security properties of such keys to design a Tor-like anonymity network, without making any assumptions about the computational capabilities of an adversary. Our analysis and evaluation on simulated ad-hoc wireless networks, shows that our protocol achieves a level of anonymity comparable to the level of the Tor network.**

I. Introduction

As a significantly large fraction of our personal and sensitive data is carried out on wireless systems, encryption and anonymity are in many cases essential. The Tor anonymity network [1] is an overlay network that combines Onion Routing with a light-weight system design for Internet traffic anonymization, and it is rapidly becoming the prevalent approach to anonymity today. In the core of its design, basic cryptographic primitives are used, e.g., the Diffie-Hellman key-agreement, RSA and AES encryption. The security of such cryptographic schemes relies on computational-hardness assumptions: an adversary cannot breach security in useful time, since she does not possess the necessary computational power. We ask the question: can we design an alternative, Tor-like communication scheme for wireless networks, that offers a level of anonymity comparable to the level of anonymity that Tor does, without assuming anything about the computational and memory capabilities of an adversary?

Recent work has shown that, by exploiting inherent wireless network properties, such as channel variability and noise, along with standard network coding techniques, we can fast and reliably create keys among network nodes, where the security of the keys does not rely on the computational limitations of an adversary. Algorithms that create such unconditionally secure keys were studied theoretically in [2] and translated into practical protocols for 1-hop networks in [3], and for multi-hop networks in [4]. We briefly summarize this work and, building on it, we show how we can, using the created keys and their properties, design a Tor-like anonymization network.

Similarly to the Tor anonymity approach, our goal is to enable nodes connect to the Internet, while hiding their identity within a set of potential users. Tor achieves anonymity by bouncing encrypted communications around a distributed network of relays; we similarly bounce encrypted communications among the wireless network nodes. In our use-case scenario, an information packet travels from the source node along a randomly selected path towards a final hop to the Internet. We use layered, one-time pad encryption to both secure the messages against eavesdropping, and ensure that each relay along the path is aware of only a fraction of the entire communication path, in a fashion similar to Tor.

As a use-case, consider a street protest, where participants use local communication (e.g. WiFi) to cooperate and hide the identity of someone who needs to use cellular Internet connectivity to send reports to the media (and thus might be a target for the authorities eavesdropping the local communication). In addition to eavesdropping, the authorities might interrogate a participant and force him to reveal his knowledge on the on-going communications. While Tor preserves anonymity as long as the cryptographic primitives used remain unbreakable, we aim, with our approach, to ensure anonymity even if the adversary has unlimited computational power.

The main contribution of this paper is in the design of a traffic anonymization protocol, that exploits the security properties of the keys produced using the technique in [4]. Our privacy analysis demonstrates that we can achieve a Tor-like level of anonymity and our experimental evaluation shows that we can achieve almost perfect anonymity within a group of approximately half the network size. We note that we do not advocate our protocol to substitute Tor and the existing cryptographic primitive used; however, as the bulk of our data is increasingly carried through wireless, and becomes vulnerable to new computational attacks, we believe scientists should explore new techniques to complement existing practices.

The paper is organized as follows. First, we present our setup and some background material in Section II. Next, we present our traffic anonymization scheme and its privacy analysis in Section III, and we experimentally evaluate its performance in Section IV. Section V summarizes related work and Section VI concludes the paper.

II. Setup and Background

A. System and Adversary Model

We consider a network of n wireless nodes that form a k-hop ad-hoc network, where k refers to the maximum distance (in hops) between any two nodes. From the nature of wireless, each node's transmission can potentially be received by its neighbors, i.e., all nodes within its transmission radius. We assume that every node has a unique identifier that is revealed to all other nodes in the network. We also assume that every node is an honest-but-curious node: it legitimately participates in the protocols used, but tries to breach security using the information at its disposal.

I. Safaka is supported by ERC Starting Grant ERC-2009-StG-240317 and ArmaSuisse Science and Technology Project no. 8003505807. C. Fragouli is supported by the NSF CCF award 1321120.

978-1-4799-1911-6/15/$31.00 ©2015 IEEE

In the network there exists also a passive adversary, Eve, who eavesdrops but does not reveal her presence with any form of communication, and can be located anywhere inside the network, at an unknown location. We assume that Eve has access to the same physical layer (radio technology, number of antennas etc.) as the legitimate nodes, and is not omni-present in the network. However, we assume that Eve may have infinite memory as well as unbounded computational capabilities at her disposal; this would follow the model of an adversary that does not want to reveal her identity by using specialized equipment, yet has offline access to unbounded resources to breach security. In the following we will call the adversary Eve, without specifying (unless needed) if she is a passive eavesdropper or an honest-but-curious node.

B. The Basic Tor Operations

We here summarize the basic Tor [1] operations, without describing in full detail the whole system architecture; we rather focus on the key agreement procedure and the use of the keys for anonymous communication.

A node S wants to send a message m to a public destination D (e.g. a web-server) using the Tor anonymization network, i.e., a set of collaborating nodes, the so-called Onion Routers (OR), that will relay m toward its final destination. In a first phase, S negotiates a symmetric key with each relay. Assume S selects two nodes (the minimum required, assuming S is an OR as well) R_1, R_2, as shown in Fig. 1, and agrees on two symmetric keys with each one of them:
1) S sends to R_1 the first half of the Diffie-Hellman handshake g^{x_1}, encrypted with the public key $K^+_{R_1}$ of R_1. R_1 responds back with the other half of the handshake g^{y_1}, and a hash of the negotiated key (with $F(\cdot)$ denoting a secure hash function). S and R_1 compute the key $K_{SR_1} = g^{x_1 y_1}$.
2) S sends to R_1 the packet $K_{SR_1}\{R_2, K^+_{R_2}\{g^{x^2}\}\}$, that is a request to negotiate a symmetric key with R_2, encrypted with the key K_{SR_1} (128-AES encryption). $R1$ and R_2 perform the same actions as S and R_1 respectively in step 1.

In a second phase, S communicates a message m to D by sending the packet $K_{SR_1}\{R_2, K_{SR_2}\{D, m\}\}$ to R_1, which extracts the first layer of encryption and forwards the inner packet $K_{SR_2}\{D, m\}$ to R_2; finally, R_2 extracts the second layer of encryption and sends m to D.

We note two fundamental properties of the Tor design:
• Property 1: R_1 cannot compute the key K_{SR_2}, since g^{x_2} is protected with the public key of R_2. It cannot, namely, decrypt the packet $K_{SR_2}\{D, m\}$ and reveal the message m and its final destination D.
• Property 2: R_2 does not know if it is setting up a symmetric key with R_1 or any other node in the network (in our example, node S). In other words, it does not know which is the originator of the packet $K_{SR_2}\{D, m\}$; from R_2's perspective the originator could be R_1, S or any other network node with equal probability.

These two properties ensure the basic premise of Tor: a relay knows only two nodes along the communication path, its predecessor and its successor, but cannot ultimately link S to D and m. Anonymous communication is, therefore, preserved under the presence an adversary Eve, who in this case has bounded computational power and cannot breach the security of the cryptographic primitives used.

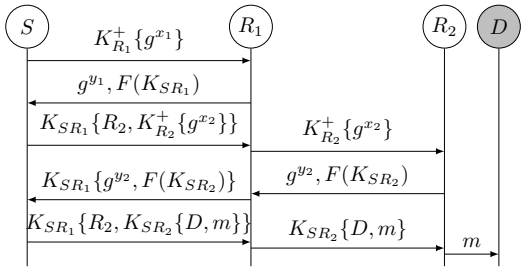

Figure 1. Tor anonymization protocol – example

C. Key Construction Based on Erasures

A basic building block of our anonymization protocol is the key-agreement procedure. Due to the inherent channel variability and noise in the wireless, Eve is prone to packet erasures, enabling, thus, the legitimate nodes to use their shared, correctly received packets (unknown to Eve) as a source of secrecy. The idea of linearly combining shared packets for constructing information-theoretically secure keys, has been presented in [2]. As an example, suppose that two nodes, Alice and Bob, share three *random* packets, i.e., packets with randomly produced payload bits, X_1, X_2 and X_3, and assume they also know that there is at least one packet out of these three that Eve does not have. Then they can both compute $K = X_1 \oplus X_2 \oplus X_3$, which serves as a pairwise key, certainly secure from Eve. Note that Alice and Bob do not need to know *which* packets Eve has, they only need to know *how many* she has. In general, using Maximum Distance Separable (MDS) codes, we can securely create as many linear combinations as the number of packets Eve does not have [2].

We built on this idea to propose a concrete secret-key agreement protocol in [3], and we were able to produce secure keys at a rate of Kbps, in an 1-hop test-bed. We further extended our protocols for multi-hop networks in [4], where the existence of interference and multi-path provides additional sources of packet erasures, and we simulated the key-agreement in multi-hop setups up to 5 hops and 500 nodes. The observed high secrecy rates (at the order of Kbps) suggested the feasibility of using the produced keys as one-time pad encryption keys.

The anonymization protocol presented in this paper is not, however, bounded to this specific key-generation technique. Any key-agreement procedure, that enables nodes in a multi-hop wireless network to establish secure pairwise and group keys, under the presence of Eve, would serve as the base of our traffic anonymization protocol.

D. Goals and Performance Metrics

The goal of our traffic anonymization protocol is to create uncertainty to Eve about the sender and the receiver of a given message m. Let $\mathcal{S}, \mathcal{D}$ denote the random variables that describes who the actual sender and receiver is, and E Eve's

knowledge on the protocol and the produced traffic.

• The *sender uncertainty* $U_{\mathcal{S}}$ and *destination uncertainty* $U_{\mathcal{D}}$ are measured as the conditional entropies:

$$U_{\mathcal{S}} = H(\mathcal{S}|E) \text{ and } U_{\mathcal{D}} = H(\mathcal{D}|E).$$

• The *sender-receiver uncertainty* expresses the uncertainty about the communication pair and equals

$$U_{\mathcal{S}-\mathcal{D}} = H(\mathcal{S}|E) + H(\mathcal{D}|E).$$

$U_{\mathcal{S}-\mathcal{D}}$ gives the entropy of the joint distribution of $(\mathcal{S}, \mathcal{D})$ in case the two random variables are independent from Eve's perspective. The maximum source uncertainty within a group would be achieved if Eve believes each group member to be the source with equal probability.

III. Traffic anonymization

We here describe a communication scheme aiming to provide a level of anonymity that is comparable with the anonymity level of the Tor system [1], albeit also secure against computationally unbounded, but presence-limited, adversaries. The design of our protocol aims in satisfying the two fundamental Properties 1 and 2 of Tor, that we described in Section II-B.

A. Main Ideas and Examples

The steps of negotiating the symmetric keys in Tor, are essentially replaced by the key-generation protocol in [4]: In an initial step, legitimate nodes produce and transmit *random* packets; next, they publicly announce to each other which packets they correctly received. In a second step, the nodes linearly combine their common packets to create keys, on request. We now describe how they can use these resources for anonymous communication, as depicted in Fig. 2.

Example: S wants to communicate message m to D:
1) S randomly selects two relay nodes R_1 and R_2.
2) S uses one-time pad encryption to send to R_1 the message m and the identities D and R_2 through the packet:

$$K_{SR_1}\{R_2, K_{G_2}\{D, m\}\} = K_{SR_1} \oplus \{R_2, K_{G_2} \oplus \{D, m\}\},$$

where K_{SR_1} is a secure pairwise key between S and R_1 (we will call this link encryption), and K_{G_2} is a random packet that all nodes in a group G_2 have successfully received, with $\{S, R_2\} \subset G_2$ but $R_1 \notin G_2$, i.e., this packet is secret from R_1 (we will call this group encryption).
3) R_1, that has the pairwise key K_{SR_1}, removes it to find out that it needs to forward to R_2; it then re-encrypts using the pairwise key $K_{R_1R_2}$ and sends the packet:

$$K_{R_1R_2}\{K_{G_2}\{D, m\}\} = K_{R_1R_2} \oplus K_{G_2} \oplus \{D, m\}.$$

R_1 does not possess K_{G_2} and thus does not learn D and m.
4) R_2 removes both $K_{R_1R_2}$ and K_{G_2}, and sends m to D; R_2 does not know that S originated message m.

The link and the group keys serve complimentary roles in ensuring anonymity. The role of the link keys, K_{SR_1} and $K_{R_1R_2}$, is to hide R_2, D, and m from intermediate relays as well as external eavesdroppers, similarly to the symmetric encryption in Tor. The role of the group key K_{G_2} is threefold. First, it hides the identity of the destination from R_1, who only learns the identity of the next relay R_2. Second, because it also hides the message m from R_1, even if R_1 overhears the unencrypted message m that R_2 transmits, it cannot link m to packet $K_{G_2}\{D, m\}$ and thus again will not learn the destination. Third, it hides the identity of the sender within the group G_2 for R_2, who only knows that $S \in G_2$. In other words, the role of the group keys is to provide the basic anonymity property: each relay knows only its predecessor and its successor in the communication path; similarly to Tor.

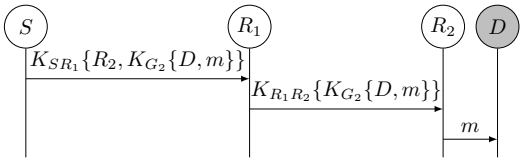

Figure 2. Traffic anonymization protocol – example

How we have created keys has significant implications on the anonymity protocol we have designed. Our protocol essentially combines the layered (onion) encryption of Tor with one-time pad encryption. We can afford to use one-time encryption, exploiting the high key-generation rates in [4]; S can randomly select R_1 and R_2 because we can create keys between all pairs of nodes; and because we distribute random packets to create shared randomness, we can easily find large sets G_2 that share common random packets (see Section IV). The size of G_2 is important as it determines the amount of anonymity: the larger it is, the harder it is for the adversary to guess correctly the originator of a packet.

B. Traffic Anonymization Protocol

The protocol we described in the previous example naturally extends to multiple relays, as described next.

1) S selects randomly t relays $R_1, \ldots, R_t$.
2) S creates each group key K_{G_i} by randomly selecting a packet from the packet dissemination phase among the ones that (a) are not known by R_{i-1}, (b) are known by R_i, (c) are known by at least σ other nodes, where the parameter σ defines the minimum size of G_i.
3) S sends to R_1 a packet of the form:

$$K_{SR_1}\{R_2, K_{G_2}\{R_3, K_{G_3}\{\ldots K_{G_t}\{D, m\}\}\}\}$$

such that $\{S, R_i\} \subset G_i$, $R_{i-1} \notin G_i$.
4) The first relay R_1 decrypts the packet using the link key K_{SR_1} and encrypts the encapsulated packet destined for R_2, using the link key $K_{R_1R_2}$, and sends the packet:

$$K_{R_1R_2}\{K_{G_2}\{R_3, K_{G_3}\{\ldots K_{G_t}\{D, m\}\}\}\}.$$

5) The relay R_i sends to R_{i+1} the packet:

$$K_{R_iR_{i+1}}\{K_{G_{i+1}}\{R_{i+2}, K_{G_{i+2}}\{\ldots K_{G_t}\{D, m\}\}\}\},$$

which is produced as follows: 1) After removing the two outermost encryption layers (first with a link key

$K_{R_{i-1}R_i}$ and then with a group key K_{G_i}), the received packet reveals the next relay R_{i+1} on the path and an encapsulated packet that is encrypted with $K_{G_{i+1}}$. 2) R_i encrypts the encapsulated packet with $K_{R_i,R_{i+1}}$.

6) The last relay R_t simply forwards m to D, after removing the two remaining encryption layers.

Note that this protocol can be used to also support two-way communication: since every relay knows the preceding relay along a path, they can forward a response from D by applying the same type of encryptions but now in the reverse direction.

C. Privacy Analysis

We use the term *flow* to describe the set of all the packets that are exchanged to support the communication of a specific S-D pair. We are interested in four forms of *unlinkability*:

• *Unlinkability of packets:* Eve is not able to tell whether two (or more) overheard packets belong to the same flow.

• *Unlinkability with the destination:* Eve is not able to tell which is the destination of an overheard packet. We measure this with the metric $U_{\mathcal{D}} = H(D|E)$.

• *Unlinkability with the source:* Eve is not able to tell which is the source of an overheard packet. We measure this with the metric $U_{\mathcal{S}} = H(S|E)$.

• *Source-Destination unlinkability:* Eve does no learn which source communicates with which destination.

Recall that for us Eve may be a passive external eavesdropper, or an honest-but-curious node in our network.

1) Unlinkability of Packets: Clearly, we need to have more than one flows in our network, as the uncertainty, to which flow a packet belongs, is constrained by the number of flows. We will next assume that a large number of flows share the network; this is also a basic premise of Tor.

If Eve overhears a packet, she could learn which node transmitted it and which node received it (she could learn one link of the path); if she could overhear multiple packets that she found were part of the same flow, she could piece together parts of the path, and thus her uncertainty about the communicating parties would reduce. Packet unlinkability is essential to avoid giving such side information to Eve.

In our protocol, the link keys together with the group keys, ensure that all transmitted packets are statistically independent and thus, even if Eve observes multiple of them, she cannot correlate them. The use of link keys ensure that packets appear statistically independent of each other whether or not they belong to the same flow. This property holds also against a relay: the content of a packet a relay can see, by knowing its own link key, is independent of the same packet encrypted with a different link key. It means relays cannot recognize packets that they themselves forwarded earlier along the path.

The only packet that is not protected with a link key is the last packet of the flow. Hence, it remains to protect the last message from a relay, who knows also a link key. The group key plays a role here: it encrypts the message from relays, which makes also the last packet independent and thus unlinkable with its previously seen encrypted version. It follows that for all nodes (including Eve) packets remain unlinkable with each other in the network.

2) Unlinkability with the Destination: If Eve overhears the transmission of R_t (of the last relay on the path), then she learns the destination of the packet, and thus $U_{\mathcal{D}} = 0$; trivially, this is the case if Eve is the node R_t. The leakage of this information is unavoidable, since D is outside the network. This is also the case in Tor.

If Eve overhears the transmission of any other packet, the packet remains unlinkable with its destination. Indeed, link keys protect the identity of the destination from any node who is not a relay on the path; and group keys protect the identity of the destination from the nodes that are relays. It follows that for any node, including relays, the destination remains unlinkable with any packet of the flow except for the last-hop unencrypted packet.

3) Unlinkability with the Sender: We here need to distinguish cases depending on which node Eve is. First, assume Eve is not one of the R_i relays on the path; then the link keys make the different packets of a flow indistinguishable, i.e., Eve cannot tell if an overheard packet is the first packet of the flow, and cannot learn anything about the sender. Next, assume Eve is R_1. Then Eve knows that S is the source, and thus for the first packet $U_{\mathcal{S}} = 0$. This is also the case in Tor, if the adversary manages to compromise the first onion-router, to which the user's onion-proxy connects.

Assume now Eve is a relay R_i on the path. R_i knows that $S \in G_i \cap G_{i+1}$, since the source has to be a member of both groups. Thus it can link S with the group $G_i \cap G_{i+1}$. In the example of Fig. 2, R_2 learns that $S \in G_2$. Ideally, any node in $G_i \cap G_{i+1}$ would appear equally likely to be the actual source, i.e., Eve would infer a uniform distribution over these nodes. However, the selection of the groups G_i does not guarantee this property; the distribution will be skewed from the uniform. We numerically evaluate the uncertainty $U_{\mathcal{S}}$ in the evaluation Section IV, and find that it is very close to uniform.

4) Sender-Receiver Unlinkability: From the previous arguments it follows that the uncertainty about the communicating pair $U_{\mathcal{S}-\mathcal{D}} = U_{\mathcal{S}} + U_{\mathcal{D}}$ is never 0. Moreover, $U_{\mathcal{S}-\mathcal{D}}$ is the largest possible when Eve is not one of the relays R_i and she does not overhear the last packet of the flow. It is reasonable to assume that the uncertainty about the destination $H(\mathcal{D})$ is larger than about the source $H(\mathcal{S})$, since the destination could be any server on the Internet. Thus $U_{\mathcal{S}-\mathcal{D}}$ takes its smallest value if Eve is the last relay R_t on the path. In our evaluation we assume this worst-case situation and numerically evaluate the sender-receiver uncertainty under this condition.

5) Side Information Attacks: When analyzing the unlinkability properties that our protocol provides, we only considered the information that the content of the transmitted packets can reveal to an adversary. However, an adversary may also observe additional side-information; the amount and type of this information depends on the actual implementation of the protocol and also on its interplay with other protocols (e.g. the routing protocol used). Such side information is present irrespective of the applied anonymizer solution; indeed most

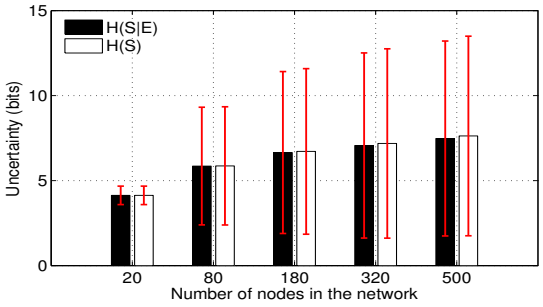

Figure 3. Anonymity for density $d = 20$ and $k \in \{1, 2, 3, 4, 5\}$

known attacks against Tor are of this kind (e.g. [5], [6]). The possible sources of side information include traffic analysis (timing information, number of sent/received packets), topology (routing and location information), and application level analysis. For an overview of side-channel attacks we refer to [7]. Although it is not possible to conceal all the side information, by its design, our protocol offers a level of anonymity comparable to that of Tor.

IV. Performance Evaluation

We use the Java-based, discrete event-driven simulator JiST [8] together with the SWANS library [9], for ad-hoc wireless networks. In Table I we summarize the configuration parameters of the simulation setup. We use an IEEE 802.11b/g compliant MAC configuration. We simulate a wireless ad-hoc network as a set of n nodes uniformly at random placed on a square area of dimension x meters. All nodes have the same communication capabilities that yield a transmission range of r meters. Under the configuration parameters in Table I, the range is $r \approx 200m$. Therefore, for a k-hop network we set $x = k * r/\sqrt{2}$. We define the *unit area* as an 1-hop area. We consider networks with fixed network density per unit area, that is, for a k-hop area and a given density d (nodes per unit area) we have in total $n = k^2 * d$ nodes. We first run the key-agreement in [4] and then our anonymization protocol.

Fig. 3 numerically evaluates the sender-receiver unlinkability that our protocol achieves. We assume that Eve is the worst-case node for us relay R_t (as we explain in Section III-C); in this case, the sender-receiver uncertainty equals the source uncertainty $U_S = H(S|E)$. We compare this to the ideal uncertainty $H(S)$ Eve would have, if each group member would be the source with equal probability (note that $H(S) = 5$ amounts to uniform probability within a group of size 2^5). We find that with our protocol we can restrict Eve to only learn that the source belongs in a set of size approximately half the network population; moreover, Eve perceives each node in the group to be the source with probability close to uniform.

V. Related Work

Our anonymity protocol combines the onion routing of Tor [1] with one-time pad encryptions, to provide protection against computationally unbounded adversaries. There also exist alternative anonymous routing protocols specially designed for ad-hoc networks (e.g. [10], [11]) but they all build on

MAC Layer	Slot Time	$20\mu s$
	W_{min}	31 slots
	W_{max}	1023 slots
	SIFS	$10\mu s$
	DIFS	50 μs
	PHY header	192 bits
	MAC header	272 bits
	DATA frame header	464 bits
	ACK frame	304 bits
PHY Layer	Frequency	2.4 GHz
	Basic Rate	1 Mbps
	Data Rate	36 Mbps
	Tx Power	15 dBm
	Sensitivity Threshold	-81 dBm
	Reception Threshold	-71 dBm
	Reception Model	SNR
	SNR Threshold	15 dB
Channel Model	Propagation Model	TwoRay
	Fading Model	Rayleigh
	Interference Model	AdditiveNoise

Table I
Configuration of simulation setup

computational limitations. In this paper, we considered privacy in the presence of a passive adversary; an active adversary might for instance intentionally introduce timing patterns that she can later identify [5]. Introducing latency and mixing [12] can make timing attacks more difficult but at the same time decreases throughput.

VI. Conclusion

We presented the design and the analysis of an anonymization protocol that leverages the key-generation in [4], in order to offer a Tor-like level of anonymity, yet without relying on the computational limitations of Eve. Our experimental evaluation shows that, with our protocol, we can achieve almost perfect anonymity within a group of roughly half the network size.

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th Usenix Security Symposium*, 2004.
[2] M. Jafari Siavoshani, S. Diggavi, C. Fragouli, U. K. Pulleti, and K. Argyraki, "Group secret key generation over broadcast erasure channels," in *Asilomar Conference on Signals, Systems, and Computers*, 2010.
[3] I. Safaka, C. Fragouli, K. Argyraki, and S. Diggavi, "Exchanging pairwise secrets efficiently," in *INFOCOM*, 2013.
[4] I. Safaka, L. Czap, K. Argyraki, and C. Fragouli, "Efficient key exchange for wireless multi-hop networks," Tech. Rep., 2015. [Online]. Available: http://infoscience.epfl.ch/record/207888
[5] S. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *IEEE Symposium on Security and Privacy*, 2005, pp. 195, 183.
[6] T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price, "Browser-based attacks on Tor," in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science. Springer, 2007, pp. 184–199.
[7] S. J. Murdoch, "Covert channel vulnerabilities in anonymity systems," Ph.D. dissertation, University of Cambridge, 2007.
[8] R. Barr, Z. J. Haas, and R. van Renesse, "JiST: An efficient approach to simulation using virtual machines," *Software: Practice and Experience*, vol. 35, no. 6, pp. 539–576, 2005.
[9] R. Barr, Z. J. Haas, and R. Van Renesse, "Scalable wireless ad hoc network simulation," *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*, 2005.
[10] B. Zhu, Z. Wan, M. Kankanhalli, F. Bao, and R.-H. Deng, "Anonymous secure routing in mobile ad-hoc networks," in *IEEE International Conference on Local Computer Networks*, 2004, pp. 102–108.
[11] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *MobiHoc*, 2003.
[12] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, 1981.