Author Response to Reviews of

DAENet: Making Strong Anonymity Scale in a Fully Decentralized Network

Tianxiang Shen, Jianyu Jiang, Yunpeng Jiang, Xusheng Chen, Ji Qi, Shixiong Zhao, Fengwei Zhang, Xiapu Luo, Heming Cui
IEEE Transactions on Dependable and Secure Computing,
`Manuscript ID:TDSC-2019-11-0652`

Dear Referees,

Thank you very much for your valuable comments. We are profoundly grateful for the feedback, which greatly helped us improve the quality of the original manuscript. Accordingly, we have modified the manuscript by taking into consideration both reviewers' requests. We take it as a positive gesture that both referees have submitted positive remarks about the manuscript. The revised manuscript has been re-written by taking into account all these recommendations. The changes in the revised manuscript are cataloged here. We have, thus, replied to the Reviewers' comments, and indicated clearly what we have changed.

RC:*Reviewer Comment*, AR:*Author Response*, □ Manuscript text

1 Associate Editor

RC: Please carefully address every comment from each reviewer.

AR: Thank you for allowing us the chance to improve our work through the revised manuscript.We have tried to include all the suggested changes in the updated manuscript. Individual reviewer-specific comments, which have been incorporated in the updated manuscript, are specifically cataloged in this document.

2 Reviewer #1

RC: I like the idea of the paper, but I just don't think it's publishable in the current format. The writing, organization, and evaluation need to be significantly improved. It looks like you used the Loopix paper as a guideline for your evaluation, but that's a very different network, and so your eval is going to be different.

AR: Thank you for your recognition of our ideas and pointing out the shortcomings of our original manuscript. For writing, we have corrected all mistakes, and given the revised work a thorough read-through to reduce writing errors further. For paper organization, we have re-written the main part of this paper to highlight the contribution of this work better and shown our effort by providing a latex-diff manuscript (attached). For evaluation, we have re-written the evaluation setup to clearly introduce our motivation of our baseline systems such as Loopix, the details can be found in section 6.

RC: The discussion of Tor vulnerabilities in the introduction is dated (based on a paper from 2005), and does not take into account more recent Tor improvements, such as cover traffic, guard nodes, and etc.

AR: Thank you for pointing out this issue. We have added discussion about recent attacks and defending techniques on Tor (e.g., cover traffic and guard node selection), and moved the detailed description of Tor to the related work. We have included the following information in section 8 of the revised manuscript.

> The most popular relay-based anonymous communication system is Tor [15]. Due to its popularity and transparent development processes [61], many researchers have explored attacks that can de-anonymize Tor users and hidden-service providers by monitoring the network traffic. Recent attack vectors for Tor include BGP-based attacks [62], [63], website fingerprinting [62], [63], [64], [65], traffic correlation [10], [11], [12], [66], congestion attack [67], [68] and targeted DoS [39], [69], [70]. Meanwhile, researchers also propose methods to enhance Tor's security by optimizing the bandwidth report for selecting guard nodes [71] and monitoring circuit construction [72]. Also, some recent Tor improvements consider generating cover traffic within middle routers of circuits, such that the middle routers can hide any relationship between compromised entry and exit nodes [73], [74].

RC: How/who assigns ID's? (section 2.2) You discuss this later, but not here.

AR: Thank you for your comments. We have given more details of DAENet's underlying communication network in the revised manuscript in section 2.2. Specifically, in the underlying network, each participant joins the network by sending a *join* request to a known DAENet node. The node will assign an identifier to the participant by using the consistent hashing technique and help the participant set up its routing table by several rounds of consultations to its neighbors. In DAENet, the node is called the *guarder node*. The guarder node also serves as an attestation server to verify whether an unmodified DAENet's program is executed inside a real SGX host.

RC: DAE v.s. SGX-Tor – I think this part is in the wrong place. The comparison vs SGX-Tor makes a lot of claims that haven't been supported or even explained. Possibly move this later in the paper.

AR: Thank you for pointing out the organization problem. We have moved this comparison to the related work in section 8.

RC: I don't think it's so easy to just say "SGX is trusted" when there are several known vulnerabilities that compromise the security of the system. For SGX-Tor, if SGX fails, you're left with Tor, which is somewhat secure. With your system, if SGX is compromised, there is no security left at all. This is a major difference, and I think it merits at least a discussion of the impact of known SGX flaws on your system.

AR: Thank you for your valuable insights. The main usage of SGX in DAENet is to securely and efficiently select a set of distributed trustworthy shuffling nodes, so that our system can scale to a large number of users while maintaining low end-to-end latency and constant-size bandwidth cost. However, we have added the following information to discuss the SGX vulnerabilities that could happen and how DAENet defends against them in section 4.1.

We notice that an SGX may be compromised because of SGX vulnerabilities [48], further compromising the anonymity provided by DAENet. DAENet solves this problem by using two approaches. First, such vulnerabilities can usually be fixed through CPU microcode updates [48], and such updates increase the Security Version Number (SVN) used for attestations. DAENet's guarder node checks the latest SVN within the network and rejects nodes with SVN that is smaller than this value during attestations, so nodes with out-of-date microcode (i.e., contain potentially compromised SGX) cannot join the network. Second, for vulnerabilities that cannot be fixed through CPU microcode updates, Intel returns a revocation certificate list during attestations. DAENet rejects attestation reports signed by these certificates and avoids the admissions of nodes with SGX vulnerabilities that cannot be fixed.

RC: You discuss the different attacks you will protect against, but you don't specify *how* the system will protect against them. Contrast this with Section 3.3 of the SGX-Tor paper to see what I'm talking about.

AR: Thank you for your comments. Following your advice, in section 3.3 where we talked about the security goals before, we have added corresponding defending techniques for clearer elaboration. Specifically, there are three security goals and corresponding privacy approaches. First, to defend against targeted DoS attacks, we run the anonymous protocol in a fully decentralized network where participants are equal. Second, to defend against passive traffic analysis attack, we shuffle real messages with indistinguishable dummy messages inside SGX to mess up the orders of input and output packets, and disseminate messages to each participant's neighbors with full randomness. Third, to resist disclosure attacks, we use randomly generated dead drop nodes to build up different and unpredictable routing circuits in each communication round, preventing active attackers from tracking and revealing sender identities.

RC: The theorems in this section aren't actually theorems. They're just statements of equations that come directly from your formulations. Also, you refer to Theorem 2 as a "lemma" in the text preceding it, which is not a theorem. You also say it's an "upper bound" but there is no inequality, which an upper bound would imply. You later state that 1/y is the is "the upper bound for an adversary to correctly link the input message x and the corresponding output message x1." which might be true, but this is not derived or demonstrated in your paper. This looks like what they did in the Loopix paper, but they provided proofs in an appendix, which you do not.

AR: Thank you for pointing out the writing mistake in this paper. The previous Theorem 1 that described the process of pulling messages from shuffle pools has been refactored to be a statement in the revised manuscript. The statement is included in section 4.3. We give an upper bound probability 1/y because all outgoing messages are from the host's shuffle pool, hence the linking probability is limited to the total number of existing messages in current host. As there are totally y messages as we defined, the upper bound on the probability that adversaries can correctly do the traffic correlation is thus 1/y. This inference applies to other shuffled-based systems that defends against traffic correlation attacks as well. Loopix handles traffic correlation attack and also gives an upper bound on the attack probability (i.e., 1/(l+k)) because Loopix assumes a shuffle pool is mixed l+k messages presently.

RC: Question about detecting malicious packet drops in DAENet.

AR: In this revised manuscript, we have removed the detection protocol which was used to defend against eclipse attacks in the original manuscript, and our decision is based on the observation that the consequence of an eclipse attack in our system can be treated equally as a targeted DoS attack or node failure, thus a node under eclipse attacks can simply rejoin the network for the anonymous service. Specifically, the adversaries can arbitrarily drop packets in the network, including some underlying P2P control messages that are used for maintaining the membership, causing eclipse attacks that partition some nodes from the network. In that case, some nodes in the network will lose connection with these attacked nodes and will remove these attacked nodes from their routing tables, which is the same as the case where nodes are under targeted DoS attacks or failed. As a result, the partitioned nodes can simply wait for a short time and then rejoin the network. We have added this discussion in section 5.2.2.

RC: One malicious dead drop node won't compromise sender/receiver confidentiality, but what about multiple nodes. Specifically, what fraction of the dead drop nodes need to be non-malicious? This is a key number because it's the easiest place to compromise the network.

AR: Thank you for your valuable comments. We have added corresponding discussion in section 4.4. Specifically, compromised dead drop nodes might hurt the network from two aspects: liveness of communication and anonymity of users. For liveness, if multiple dead drop nodes are malicious, they might refuse to exchange message payload and forward messages. If so, the session nodes who are communicating with each other can safely switch to other unused dead drop nodes for message exchange in next communication rounds without compromising anonymity.

For anonymity, DAENet's anonymity will still hold as long as there is one honest node in a circuit. We provide the following information in the revised manuscript.

Conversation with Compromised Nodes: Even with some fully-compromised dead drop nodes, DAENet can still preserve anonymity due to the following reasons. First, adversaries cannot determine which nodes are selected as dead drop nodes in current conversation, and further compromise these nodes. This is because the *DeadDrop_keys* are generated inside SGX enclaves without involving an untrusted third-party, the locations of dead drop nodes used in the communication are kept confidential to other participants except for the session nodes, making the communication circuit unpredictable.

Second, even if the adversaries can control a fraction of nodes in the network, and these compromised nodes are happened to be selected as the dead drop nodes for a conversation, the anonymity guarantee still holds as long as one node in the circuit is honest. This is because our distributed shuffling protocol guarantees oblivious traffic pattern, and such oblivious traffic pattern offers strong anonymity against traffic analysis: a single honest participant in a circuit that correctly executes message shuffles is enough to ensure anonymity. Thus, even if all dead drop nodes are compromised, these dead drop nodes still cannot determine who is communicating with whom. In addition, in section §5.2.1, we prove that the adversaries have low attack ability (i.e., small probability) to control all relays in a circuit when DAENet scales up.

RC: epsilon for your network is quite high, even for large numbers of nodes. (Compare this with Loopix.) You also only analyze this for two parameters of your network, and

it seems like you combine shuffle rate and emission rate? Shouldn't you explore each parameter independently?

AR: Compare DAENet's likelihood difference with Loopix, DAENet's likelihood is smaller than Loopix's in the worst case, while in the best case Loopix's likelihood is smaller than DAENet's. However, Loopix achieves this by sacrificing communication latency with increasing delays. We have added the likelihood comparison with Loopix in the revised manuscript in section 5.1.2.

> Loopix also uses likelihood to evaluate its defending capability against global traffic attacks. Even if Loopix's likelihood can be smaller than DAENet in some cases, it incurs additional message delay in each mix node. Specifically, in Loopix's likelihood evaluation setup (i.e., a topology of 3 layers with 3 mix nodes per layer), when Loopix achieves comparable likelihood as DAENet (0.25), it incurs additional 1s delay in each mix node. Thus, Loopix sacrifices at least 3s latency throughout all three layers of shuffles which is larger than DAENet's end-to-end communication latency (see §6).

We analyze the likelihood from two aspects: the scale of the network and the shuffle rate in each host. We do not study the likelihood difference with different message emission rate because DAENet enforces participants to have the same emission rate which can prevent adversaries from learning sensitive information by observing whether a participant is idle or busy. Thus, changing the emission rate will not affect anonymity.

RC: 6.1 Line 51-52 – I'm not sure what you mean when you say "DAENet can identifies them". There is no explanation here for the lack in increase in latency. This also raises the question of what happens with anonymity when almost every packet is real and not cover traffic.

AR: We have given more concrete explanations about the lack of latency increase in section 6. For anonymity, since real traffic and dummy traffic are equally shuffled, when almost every packet is real and not cover traffic, the anonymity guarantee still holds: traffic analyzers still cannot correlate an input and output message because each participant shuffles messages to mess up the order of messages, and all messages are mixed regardless of whether a message is real or not.

RC: Bandwidth usage – I don't see how you avoid network bandwidth for dummy messages. It says here they replace dummy messages with control messages, but how many control messages are there? And if you limit dummy traffic, don't you either limit throughput or limit the anonymity provided by cover traffic?

AR: Thank you for your comments. We have updated the information of bandwidth usage in section 6.

RC: The related works section is extremely small and limited. There is no real comparison between existing works and the presented work. It's just a list of papers and names. This needs to be significantly improved, and a design comparison between your system and existing systems needs to be added.

AR: Thank you for your valuable comments. We have significantly improved the related work chapter in the revised manuscript. Currently, in our related work, we not only compare the design points of our system with TEE-based anonymous network and other shuffle-based mix networks, but also outlook alternative approaches that might be used to do verifiable shuffles.

3 Reviewer #2

RC: The presentation of this paper is satisfactory. Nonetheless, the following issues should be addressed for better readability:

AR: Thank you for your encouraging comments.

RC: Question about detecting malicious packet drops in DAENet.

AR: In this revised manuscript, we have removed the detection protocol which was used to defend against eclipse attacks in the original manuscript, and our decision is based on the observation that the consequence of an eclipse attack in our system can be treated equally as a targeted DoS attack or node failure, thus a node under eclipse attacks can simply rejoin the network for the anonymous service. Specifically, the adversaries can arbitrarily drop packets in the network, including some underlying P2P control messages that are used for maintaining the membership, causing eclipse attacks that partition some nodes from the network. In that case, some nodes in the network will lose connection with these attacked nodes and will remove these attacked nodes from their routing tables, which is the same as the case where nodes are under targeted DoS attacks or failed. As a result, the partitioned nodes can simply wait for a short time and then rejoin the network. We have added this discussion in section 5.2.2.

RC: It is questionable that the use of SGX provides enhanced security as it relies on an SGX assumption (i.e., it is not subject to side channel attacks) and periodic detection of active attacks. Is it better to emphasize efficiency and scalability (even robustness)?

AR: Thank you for your valuable insights. Following your advice, to better emphasize our strengths, we have highlighted the importance of using SGX to efficiently preserve the integrity of operations, such that our system can scale to a large number of users with moderate increasing latency. For the SGX assumption, we have added a discussion of side-channel attacks in section 7. Also, we have added the following discussion about SGX vulnerabilities on DAENet and corresponding defenses at the end of section 4.1:

> We notice that an SGX may be compromised because of SGX vulnerabilities [48], further compromising the anonymity provided by DAENet. DAENet solves this problem using two approaches. First, such vulnerabilities can usually be fixed through CPU microcode updates [49], and such updates increase the Security Version Number (SVN) used for attestations. DAENet's guarder node checks the latest SVN within the network and rejects nodes with SVN less than this value during attestations, so nodes with out-of-date microcode (i.e., contain potentially compromised SGX) cannot join the network. Second, for vulnerabilities that cannot be fixed through CPU microcode updates, Intel returns a revocation certificate list during attestations. DAENet rejects attestation reports signed by these certificates and avoids the admissions of nodes with SGX vulnerabilities that cannot be fixed.

RC: At the end of section 2.2, please clarify that N is the total number of participants.

RC: In section 3.1.1, it says, "DAENet allows arbitrary number of compromised participants in the network, but requires an adequate number of honest participants to

ensure message deliveries." Here "arbitrary number" is contradictory to "adequate number". Please revise.

RC: It says, "DAENet requires at least O(logN) out of N honest participants". It should be revised to "DAENet requires at least O(logN) honest participants out of total N participants."

AR: Thank you for pointing out these writing mistakes in our paper. We have incorporated all your suggestions in the revised manuscript, and given a thorough read-through to reduce writing errors further.

RC: At the end of section 4.1, it says "a malicious joiner node will fail to pass the attestation". Please clarify why.

AR: A malicious joiner node (i.e., guarder node in the revised version) might refuse to admit benign nodes or try to admit specific nodes. Thus, the malicious node has broken code integrity, which will be detected and revealed in the remote attestation report. We have updated the discussion in section 4.1.

RC: For theorem 2, what is the relationship between y and t,k in extreme cases? Does this theorem hold in extreme cases even if Alice is compromised?

AR: The value of y, t and k is independent. If Alice receives an input message, the message's next hop has equal probability of being any of Alice's neighbors. Thus, it is likely for Alice to hold none messages in some of its shuffle pools. In other word, whatever the value of y is, k - the number of empty shuffle pools is independent of y and can be any non-negative value while k $<\log \mathcal{N}$ still holds (N is the total number of participants in the network). This can be applied to t as well. If Alice is compromised, Alice can arbitrarily drop messages but this theorem still holds because Alice cannot intervene the shuffling process within SGX and correlate pairs of input and output messages.

RC: In section 4.5, is the existence of service providers and brokers contradictory to the P2P feature of DAENet and its security against DoS attack?

AR: The existence of service providers and broker nodes for handling registration requests is not contradictory to the P2P feature of DAENet due to two reasons. First, a service provider can assign different broker nodes to serve it's registration requests, and these broker nodes are randomly distributed in the fully decentralized network. Second, the broker nodes are stealthy to the adversaries. This is because the only information the adversaries can get is the identities of broker nodes. As our stealthy P2P network hides nodes' identities, the adversaries cannot locate the broker nodes in the network.

With gratitude,
Tianxiang Shen
Author of the manuscript