Optimistic fair exchange for secure forwarding

Melek Önen, Abdullatif Shikfa, Refik Molva
Institut Eurécom
Sopiha-Antipolis, France
{Melek.Onen, Abdullatif.Shikfa, Refik.Molva}@eurecom.fr

***Abstract*— Several cooperation enforcement schemes based on rewarding mechanisms such as electronic cash or online credits have lately been proposed to prevent selfish behavior in ad-hoc networks. However, these schemes suffer from the lack of fairness guarantees or the reliance on costly mechanisms such as tamper-proof hardware or the requirement for Trusted Third Parties (TTPs) that are not suitable for ad-hoc networks.**

In this paper, we present a new cooperation-enforcement scheme that is perfectly suitable for ad-hoc delay-tolerant networks. The protocol is based on a simple technique called hot-potato forwarding whereby in order to receive a packet, potential recipients must first deliver an advance reward to the sender prior to the transmission of the packet. Thanks to this technique cooperation among nodes becomes mandatory and poisoning attacks and cheating actions are inherently prevented. The second contribution in our scheme is an optimistic fair exchange protocol that solves the fairness problem that is inherent to peer rewarding schemes. The protocol achieves total fairness with the help of a TTP and is optimistic in that the TTP is only involved in case of conflict between peer nodes. Correct execution of the protocol does not require any access to the TTP, so fairness is achieved without any impact on well-behaving nodes. The fairness of the protocol is validated through the exhaustive analysis of all possible protocol traces.

I. Introduction

In self-organizing networks, nodes form a temporary network without the help of any infrastructure and are assumed to have limited resources (eg. memory, battery). All available nodes are expected to perform basic network operations such as packet forwarding. Scarcity of resources would inherently foster nodes to selfishly behave in order to optimize the usage of their resources. Yet, such behavior can have a strong impact on the performance of the network [7]. In order to reduce the effect of selfishness, collaboration among parties must be encouraged by incentive mechanisms. Such cooperation enforcement schemes would guarantee fair and efficient networking operations.

Existing cooperation enforcement mechanisms are either based on some reputation mechanisms or on the use of virtual currencies. Reputations mechanisms [6], [2] ensure that each node accepts to cooperate with its neighbors based on the past behavior of the latters. Credit-based schemes [3], [10] enforce node collaboration by rewarding cooperating nodes with a certain amount of credits that they further can use for their own benefit. Yet, credit-based solutions suffer from lack of fairness. A node may indeed not forward a packet although it has received a certain amount of rewards. Existing solutions either require tamper-proof hardware at each node or an online trusted third party.

In this paper, we present a new cooperation-enforcement scheme that is perfectly suitable for ad-hoc delay-tolerant networks. The protocol is based on a simple technique called hot-potato forwarding whereby in order to receive a packet, potential recipients must first pay the sender a reward without prior knowledge about the packet. The second contribution in our scheme is an optimistic fair exchange protocol that solves the fairness problem that is inherent to peer rewarding schemes.

Section II introduces the new approach and analyzes the specific security requirements. We then describe the global forwarding mechanism and the underlying rewarding mechanism. We then investigate the exchange protocol between two nodes. Finally, we analyze and validate the security of the proposed protocol and discuss related work.

II. Overview of the proposed scheme

A. An innovative approach: Secure hot potato forwarding

As explained in the introduction, in order to reduce the impact of selfishness on the performance of the network, new incentive mechanisms have been proposed. Existing solutions can be grouped in two classes:

- reputation mechanisms [6], [2] whereby each node agrees to cooperate with some other node based on the latter's past behavior with respect to the collaborative operation as monitored by other nodes.
- rewarding mechanisms [3], [10] whereby in return for each contribution, collaborating nodes receive a certain amount of reward that they further can use for their own benefit.

Existing solutions that are based on some rewarding mechanism suffer from lack of fairness assurance and reliance on costly tamper-proof hardware or on-line trusted third parties that are not suitable for self-organizing networks.

We propose a new forwarding technique whereby in order to receive a packet, each node must first deliver an advance reward to the sender without any prior knowledge about the packet. Upon receipt of the packet, if the recipient is not the destination, the recipient forwards the packet to the next hop enroute to the destination in order to recover its rewarding amount. The motivating idea behind this approach is quite similar to hot potato routing [9] whereby packets must keep on moving until they reach their destination.

We illustrate this new approach by a basic protocol that is depicted in figure 1. In this figure, node A first asks node B if

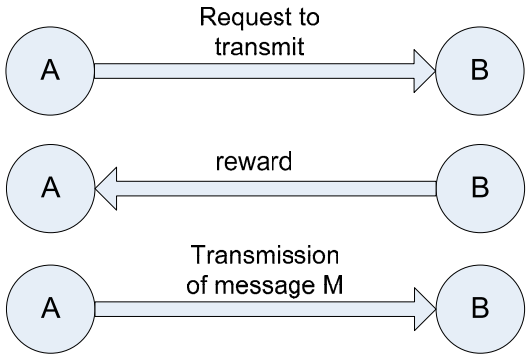

Fig. 1. The basic hot potato forwarding mechanism

it is interested in receiving a packet M. Having no knowledge about the packet and assuming that M can be destined to itself, node B accepts to receive the packet by sending a certain amount of reward to node A. A then forwards the packet to B. Further, when B realizes that the packet is not destined to itself, it proceeds similarly by trying to send the packet to the next node enroute upon receiving the reward.

In the next section, we analyze this approach with respect to cooperation enforcement and security requirements. We then come up with a complete solution whereby all security requirements are addressed.

B. Security issues

First of all, the hot potato approach enforces cooperation among the nodes of an ad hoc network with respect to packet forwarding based on two simple facts:

- a potential recipient node is motivated to deliver the reward for a packet based on the fact that this is the only way it can receive traffic destined to itself;
- in case packet is destined to another node the rational recipient is also motivated to forward it in order to recover the reward.

.

The protocol can be a potential target for Denial of Service attacks. For example, an attacker may generate some bogus messages in order to earn some rewards and use the resources of other legitimate nodes. Moreover, an intermediate node may also forward the same message to several nodes in order to earn more rewards than it actually deserves.

Furthermore, as depicted in the previous section, while defining the new scheme, the problem of fairness must also be considered. Once a certain node receives the reward, it may refrain from forwarding the message and end the exchange protocol. Existing cooperation enforcement schemes based on rewarding mechanisms either rely on tamperproof hardware or on the existence of an online trusted third party that acts as a mediator in case of possible litigation. Such mechanisms are not suitable to self-organizing networks since they either are expensive or require the availability of an online trusted third party. A concept that nicely fits with the underlying opportunistic networking model is offered by optimistic security protocols [1] whereby some communications with trusted third parties might be required but the correctness of the security protocol does not require on-line connectivity.

Based on these observations, four specific security requirements are defined:

- **cooperation enforcement**: when a node paid for a packet and received it, it must forward it;
- **protection against poisoning attacks**: nodes must be prevented from sending bogus packets;
- **protection against cheating actions**: nodes must be prevented from unduly earning rewards by sending the same packet several times;
- **fairness**: transmission of a packet subsequently to the receipt of the corresponding reward must be assured.

In the next section, we present the forwarding protocol, describe the underlying rewarding mechanism and detail the fair exchange protocol between two nodes. We then analyze the protocol with respect to the security requirements above and validate the security and robustness of the protocol.

III. The proposed scheme

In order to introduce the proposed scheme, we first briefly sketch the environment and describe the underlying rewarding mechanism. The solution is then presented as an exchange protocol to be executed by each pair of nodes on the route between the source and destination of a packet.

A. Environment

We consider the scenario whereby a source node S wishes to send a packet M to a destination node D. We assume that node S as well as all intermediate nodes can find out the next hop enroute to the destination node D.

The proposed protocol relies on the existence of a trusted third party (TTP). However, as depicted in section II-B, the TTP is only required when a node crashes or attempts to cheat. The TTP does not initiate any communication. In addition to resolving conflicts, the TTP is also responsible of crediting and debiting nodes with respect to their rewards.

Moreover, each node generates a pair of public and private keys that are required for digital signature operations during the exchange protocol. In addition to their own public and private keys, all nodes store the TTP's public key in their memory.

B. The rewarding mechanism

The rewarding mechanism is a key element of the proposed protocol since it stimulates nodes to cooperate and prevents them from cheating. Comparing with E-cash [4], [5], in this specific scheme, rewards exhibit different properties that are enumerated as follows:

- **no anonymity**: a reward is tightly bound with the payer's and payee's identities and the hash value of the corresponding message.
- **no double-spending and non-forgeability**: rewards resulting from duplication or copying of valid rewards or

forging will be detected and funding of such rewards by the TTP will be prevented.

- **no re-usability**: each reward is tightly bound to a single message exchange. Nodes periodically contact the TTP in order to transform earned rewards to usable rewards. Only the TTP can perform such operations.

We now reconsider the scenario whereby a source S wishes to send a packet M to destination node D and describe the rewarding mechanism. The first intermediate node I_1 pays S for a certain amount r. Then, in order to get a compensation, when I_1 contacts the next intermediate node I_2, it asks I_2 for a larger amount $r+c$. When the packet reaches its destination D, D recovers its rewards by further contacting the TTP. The TTP then credits D and debits the source S for the corresponding amount.

Thanks to the proposed rewarding mechanism, intermediate nodes that forward packets are rewarded to compensate the energy they deploy to do so. The forwarding mechanism ends when the destination pays some rewards and further receives the packet. The destination node D recovers its rewards by further contacting the TTP that then debits the source of the packet. If for some reason (eg lack of battery), D does not accept to receive the packet without having knowledge about it, the last node contacts the TTP with the proof of rejection generated by D in order to recover its rewards. In summary, only the source pays for the forwarding of the packet, which is quite a common strategy as in [3], [10]. Thanks to this approach, a node cannot earn rewards by just sending bogus packets to some other nodes and therefore poisoning attacks are inherently prevented.

C. The exchange protocol between two nodes

Given the global forwarding protocol and the underlying rewarding mechanism, we now turn to the exchange protocol between a pair of adjacent nodes on a forwarding path.

In the sequel of this paper, we use the following notation:

- M denotes a message;
- h denotes a cryptographic hash function;
- $E_K(M)$ denotes the symmetric encryption of message M with key K;
- $PKE_{TTP}(M)$ denotes the asymmetric encryption of message M with the public key of the TTP;
- $\sigma_A(M)$ denotes the digital signature of message M with the private key of node A.
- $reward(A, B, h(M), r)$ denotes the reward paid by node A to node B for receiving M for r amounts of reward;

Based on the security requirements defined in section II-B, we propose a new exchange protocol that is accomplished in five steps. This protocol is illustrated in figure 2. We remind that an interaction between the peer nodes and the off-line trusted third party (TTP) is only required either when a node attempts to cheat or when a node reclaims some reward.

We now describe each step of the protocol:

- **Step 0:** node A first sends the hash value of M with the requested rewarding amount. These two elements are signed with the private key of node A. Upon receipt of the request, B first checks if this request is a replay (B keeps a temporary history of previous hash values). It then verifies the signature of A. If the signature is not valid, B ends the communication. If, on the other hand the signature is valid and B has enough rewards and resources to receive the message, then node B proceeds to **Step1**.
- **Step 1:** Node B agrees to receive M and sends its signed agreement. Upon reception, node A verifies B's signature and if the latter is not valid, the communication ends. Otherwise, node A proceeds to **Step 2**.
- **Step 2:** Node A selects a random secret key K_A and encrypts M with this key using a symmetric encryption algorithm such as AES [8]. It then encrypts K_A with the public key of the TTP and signs all these data pieces with its private key. Once B receives the message, it first verifies the signature of A. If the latter is not valid then the communication ends. If, on the other hand the signature is valid then node B proceeds to **Step 3**.
- **Step 3:** Node B sends node A the requested reward. When A receives the reward corresponding to M, it verifies the validity of the reward. If the reward is not valid, then node A contacts the TTP to resolve this conflict. Otherwise, node A proceeds to **Step 4**.
- **Step 4:** Node A sends K_A to node B in order to allow node B to decrypt the message. When B receives the key K_A, it first decrypts M, verifies the integrity of M with respect to the reward using the digest value $h(M)$ included in the reward. If the message resulting from decryption does not match the reward, then B contacts the TTP to resolve the conflict. Moreover, if B does not receive the key K_A, then it can contact the TTP in order to receive K_A from the TTP. If on the other hand, the decrypted messages and the reward match, the communication ends with success.

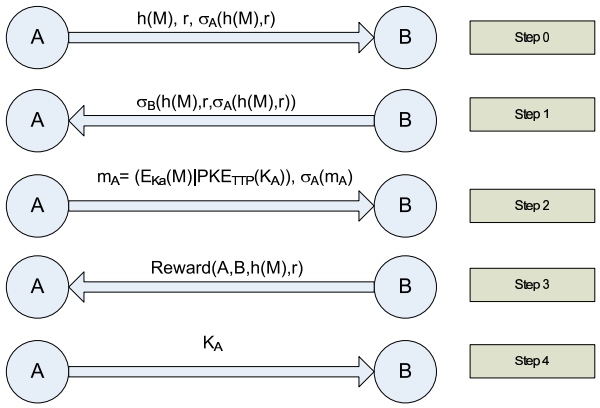

Fig. 2. The exchange protocol between nodes A and B

IV. Evaluation

A. Security analysis

We analyze the global scheme with respect to the security requirements defined in section II-B. We assume that nodes

are uniquely identified by a certificate issued by a TTP and that there exists an underlying authentication mechanism.

1) Cooperation enforcement: First of all, we showed that thanks to the proposed protocol, cooperation is mandatory because of the underlying rewarding mechanism. Nodes have no choice but to receive all incoming packets if they want to be sure to receive packets that are intended to them. Once they received packets, they must forward those that are not intended to them in order to recover rewards spent before the reception. If packets are not forwarded, then they simply lose some rewards and thus are immediately punishing themselves. Moreover, intermediate nodes that participate in the forwarding of packets are rewarded more than they are charged and therefore compensate the energy they deploy to perform such operations.

2) Protection against poisoning attacks: Thanks to the rewarding mechanism whereby only the source is charged for sending the packet, a node will not have incentive to send bogus messages for the purpose of poisoning.

3) Protection against cheating actions: In order to maximize their payoff, nodes might forward a packet to several other nodes in order to receive multiple rewards. Such an attack could strongly affect intermediate nodes accepting to forward duplicate packets since the destination will ultimately reject duplicates. However, duplicated transmissions would also be detected by the TTP due to the strong bounding between each message and the corresponding reward. If a node requires to transform two rewards that both depend on the same message, the TTP will not credit this particular node and will punish it by debiting it for an amount that is proportional to the replay frequency. Therefore, a cheating node will lose both in terms of resources that are consumed by forwarding the packets several times and in terms of losses due to the punishment.

4) Fairness: We now evaluate the fairness of the exchange protocol. The exchanged is defined to be fair if at the end of the protocol, node A receives its reward and node B receives the message. As described in section III-C, nodes A or B contact the TTP only if a problem occurs during the exchange. They also might cheat by contacting the TTP to reclaim the resolution of an inexistent conflict. In order to evaluate the fairness of the protocol, we consider all possible scenarios: in the basic protocol, the TTP might be contacted at the end of each step. We thus analyze all possible communications between the TTP and each of the nodes at the end of each step as depicted in figure 3. We do not consider a possible communication with the TTP at the end of **Step 0** since the communication ends in that case. In all other cases, the TTP resolves the conflict and ensures at the end that node A receives its reward and node B receives the message. We denote m_1 and m_2 messages that are exchanged at **Step 1** and **Step 2**, respectively.

At the end of **Step 1** or **Step 2**, node A may contact the TTP by claiming that it has already sent the message m_A to node B and did not receive the reward. In this case, in order to resolve this conflict, node A must send the correct message

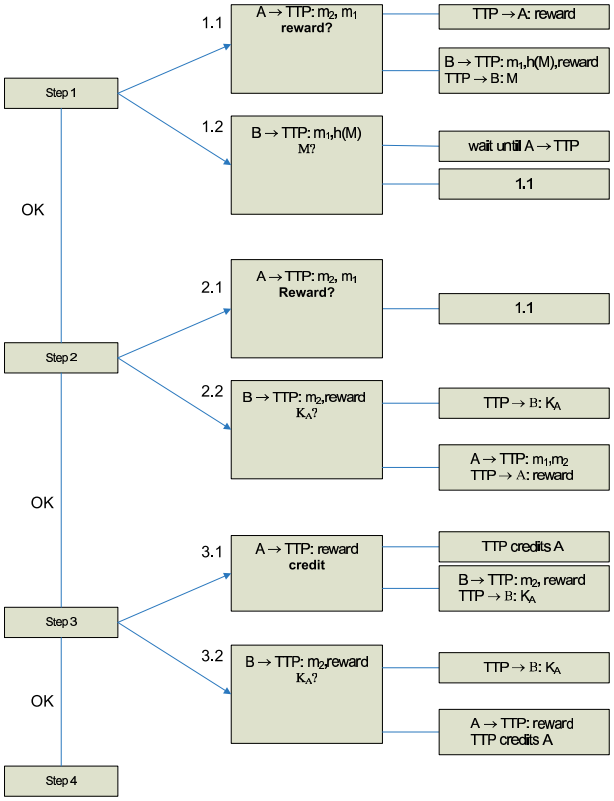

Fig. 3. Fairness validation

M to the TTP. If the message is correct, the TTP credits A and keeps M in memory until B sends a request to get M.

Moreover, at the end of **Step 1**, node B may also contact the TTP by claiming that node A did not send m_A. In this case the TTP must wait node A and proceed as in the previous case. If on the other hand, without sending any reward to node A, node B contacts the TTP by sending the message m_2 received at **Step 2** and m_1, its agreement, and by claiming that it has already sent the reward but did not receive K_A, B must automatically send the reward to the TTP. Then, the TTP resolves this conflict by first verifying the reward and then decrypting K_A. When A further contacts the TTP, the latter sends the corresponding reward.

At the end of **Step 3**, B may claim K_A although it has sent the corresponding reward. Similarly, the TTP resolves the conflict by sending K_A to node B and by crediting A if this latter contacts him.

B. Performance evaluation

In this section, we evaluate the performance of the scheme in terms of memory storage, computational cost and communication overhead. The computational cost and communication overhead have a direct impact on the battery usage.

First of all, the computational activity of each node includes a signature generation and verification at each step. Moreover, the forwarding node must perform a symmetric encryption over the message and an asymmetric encryption on the key with the public key of the TTP. The memory cost is related to

the number of rewards received for each message. This cost will also have an impact on the frequency of the communication between the node and the TTP.

The exchange protocol is performed in five steps. The message is sent at **Step 3** with its signature and the encryption key. Additional steps only involve some hash values or signatures and thus are not considered as costly.

Furthermore, the proposed mechanism does not require any tamper-proof hardware. As opposed to existing incentive mechanisms, the correctness of the protocol does not require the online-connectivity of the TTP. Moreover, since cheating actions or selfishness incur additional costs such as those due to communication with the TTP, there is an additional deterrent against cheating based on resource optimization.

V. Related work

In this section, we compare the proposed protocol with two credit-based solutions.

In [3], authors define two different payment models based on a new virtual currency named "nuglets". In the first model the source of the packet pays all intermediate nodes that are forwarding the packet by loading nuglets within the packet. Intermediate nodes acquire some nuglets from the packet when they forward it and if the packet runs out of nuglets then it is dropped. In the second model, as in our proposed scheme, intermediate nodes buy packets from previous intermediate nodes and the total cost of forwarding the packet is covered by the destination. Both models require tamper-proof hardware at each node. As opposed to our scheme, fairness is not an issue in these nodes since the tamper-proof hardware is assumed to represent a fully trusted, therefore fair authority.

In [10], authors propose a credit-based system denoted by Sprite that relies on the existence of a third party named Credit Clearance Service (CCS). In this solution, the source pays all intermediate nodes. As opposed to our scheme, Sprite requires an online Credit Clearance Service, whereby each intermediate node must contact the CCS whenever they forward the message in order to receive their rewards from the source. Sprite requires an immediate reachability of the TTP which is not a reasonable assumption for self-organizing networks.

We can notice a key difference between our scheme and existing solutions. Indeed, all existing solutions, whether credit or reputation based, encourage nodes to cooperate by rewarding them for providing a service. Thus, nodes are simply motivated to cooperate. On the contrary, in our protocol, cooperation is mandatory since the only way a node can receive traffic destined to itself is through cooperation.

VI. Conclusion

In order to reduce the impact of selfishness on the performance of self-organizing networks, we proposed a new credit based mechanism that enforces nodes to cooperate. Cooperation becomes mandatory since this scheme inherently forces nodes to place an advance payment of rewards in order to receive traffic destined to themselves. Once a node realizes that a packet is not intended to it, there is then a high incentive for the node to forward the packet in order to recover the rewarded amount. This new mechanism relies on a specific rewarding mechanism whereby only the source pays for the packet and intermediate nodes receive some rewards in order to compensate their resource usage. We described the exchange protocol between two nodes in detail and analyzed its security. The reliance of this protocol on a trusted third party (TTP) is alleviated by the fact that the TTP is only involved when a conflict occurs between communicating nodes. The fairness of the protocol is analyzed using a logical chart enumerating all cases.

As part of future work, we plan to investigate the rewarding model and the underlying pricing mechanism.

References

[1] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4), April 2000.

[2] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness and robustness in mobile ad hoc networks. In *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, 2002.

[3] L. Buttyan and J. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad Hoc Networks*, 8(5), October 2003.

[4] D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R. Rivest, and A. Sherman, editors, *Advances in Cryptology - CRYPTO'82*, pages 199–203. Plenum Press, 1982.

[5] D. Chaum. Blind signature schemes. In D. Chaum, editor, *Advances in Cryptology - CRYPTO'83*, page 153. Plenum Press, 1983.

[6] P. Michiardi and R. Molva. Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In *Proceedings of IFIP Communication and Multimedia Security Conference (CMS)*, 2002.

[7] P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *European Wireless Conference*, 2002.

[8] N. I. of Standards and Technology. Advanced Encryption Standard, 2001.

[9] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of Hot-Potato Routing in IP networks. In *Proceedings of ACM SIGMETRICS*, June 2004.

[10] S. Zhong, J. Chen, and Y. Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *Proceedings of Infocom*, 2003.