SecureRoutingDHT: A Protocol for Reliable Routing in P2P DHT-based Systems

Ricardo Villanueva[*][†], María-del-Pilar Villamil[*]
[*]Department of Computer and Systems Engineering
University of Los Andes, Bogotá, Colombia
Email: rvillanueva@egresados.uniandes.edu.co,mavillam@uniandes.edu.co
[†]University Simon Bolivar, Barranquilla, Colombia
Email: rvillanueva1@unisimonbolivar.edu.co

***Abstract*—Secure routing in P2P distributed hash table based systems has been an open subject for several years due to the importance of the routing protocol in these systems. Providing security at the routing level is a hard task because of the open nature of these systems. This article presents a protocol for reliable routing in P2P DHT-based systems, which mitigates routing attacks. It makes use of a quorum topology and a reputation system to provide security at the routing level. It is shown, theoretically and by simulations, that proposed protocol keeps stable the number of involved messages in the forwarding process, as well as it tolerates up to 30% of malicious nodes.**

***Keywords*-P2P DHT-based systems; security, threats; routing; quorum; reputation; Bayesian-systems.**

I. Introduction

Distributed hash table based systems are a special class of distributed system with interesting properties such as scalability, decentralization and self-organization. On top of these systems have been built a plenty applications such as distributed storage, application layer multicast, and so forth [5]. Nevertheless, on building of these applications has not been considered security as a main quality attribute, for that exists several threats to be taken into account.

Providing security to these systems is rather challenging due to DHT inherent features. According to Sit and Morris [28], threats against these systems comes from anywhere. In fact, they identified several attacks, and further classified them into routing, storage/retrieval and miscellaneous attacks. Particularly those threats against routing mechanism are extremely important, since they could compromise the proper functioning of the whole system.

The routing process is composed of two main subprocesses: routing table maintenance and message forwarding. Therefore, a malicious peer could misroute or drop messages along the path -*incorrect lookup*-, attempt to corrupt routing table entries of other nodes - *eclipse* attack [27][6])-, fool any peer through joining process in order to induced it into an incorrect network -*overlay partition*-, send unused messages or frequently joining/leaving the overlay network.

Although there are several works that have addressed this problem [33], these works are isolated, namely, only focusing on a specific system or attack; they even do not consider performance issues as number of messages. This paper presents a protocol that extends the underlying DHT to a redundant topology and makes use of one reputation mechanism in order to harden the DHT, but mantaining the number of sent messages stable and being easily coupled to other mechanisms.

This paper is organized as follows: Section II presents models and assumptions, which are used throughout all this paper. Section III presents strategies proposed to mitigate the impact of the *routing* attack. Section IV presents the reliable routing protocol SecureRoutingDHT. Section V presents the theoretical and practical (through simulations) analysis. Finally, Section VI concludes and gives some perspectives about future works.

II. Assumptions and definitions

Each DHT system is defined over an identifier space K, where peers and resources are mapped into. A closeness metric ρ is used for matching resources to peers. Commonly, this is achieved by using a proper hash function h, defined from the peers/resources set to K [5].

Furthermore, each node p within the system has at least two different types of links to other nodes, namely, p maintains links to specific close and distant nodes. Those links form the so-called *routing table*, which is used in the forwarding process. This process uses a greedy algorithm that has been implemented in three different ways: recursive, iterative and trace [33].

In *recursive* routing, a initiator p requests for a resource and *consequently this request* is forwarded by each intermediate peer to a next one independently. Whenever this request has reached to the responsible node, the *reply* is sent directly to the initiator or forwarded back by peers on the reverse path. Unlike previous, in*iterative* routing each intermediate peer sends contact information of next peer back to the initiator, hence p will be able to contact directly to the next peer. As a consequence, p is able to detect misbehaviour peers through techniques based on structure of DHTs, nonetheless the latency of the forwarding process is augmented.

Finally, *tracer* routing is a combination from both previously introduced techniques [35]. In this mechanism, each intermediate node must forward the message to next peer but also sends contact information of next peer to the initiator. Therefore, p knows about the entire process but routing latency is not affected.

In connection with the security model to be considered, it is supposed that there is a mechanism that randomly assigns a $nodeid$ to each new peer. In fact, this can be accomplished by coupling our proposal to other ones whose goal is to mitigate the *Sybil Attack* [1][17]. As a consequence, only a fraction of malicious peers exist during a period of time, as well as peers are uniformly distributed over the DHT. Moreover, a malicious peer can discard, generate or incorrectly forward any message. This model is widely known as *random fault model* [4], where a peer is malicious with probability at most f.

III. Secure Routing in DHT-based Systems

There are several proposals that try to mitigate *routing attacks*. In previous work [33] , we classified these strategies based on how attacks are addressed, namely, we identified three styles of solutions: those based on message redundancy, those based on malicious node detection and those trying to avoid malicious nodes by computing trust profiles of other peers.

As far as redundancy-based strategies are concerned, the requester sends several messages in order to increase the probability of reaching the responsible peers. In this style of solution, two approaches were identified: multi-path routing, where the requester peer sends a message among its neighbours, hence it is being forwarded to the responsible peers through multiple paths [4][11][15][23]. On the other hand, in wide-path routing strategies, peers are re-arranged in groups (*quorums*), hence the initiator peer broadcasts the request to each peer within its group, afterwards the message is broadcasted at same way by other peers until it reaches the destination quorum [19][24][36].

Concerning malicious node detection techniques, the sender detects a malicious node by verifying whether an invariant of the system is fulfilled - one of the most used invariant is that each hop is closer to the target during *lookup* process-. Otherwise, the sender requests to a previous considered-good node for another peer, in order to continue with the search [20][21][29][34].

Incidentally, strategies based on trust profile attempt to measure, under a well defined mechanism, which peers are the more suitable in the forwarding process. These mechanisms have been implemented using reputation systems and social networks. The former allows each peer to construct the profile of other peers using historical data and recommendations [9][21][24][25][26].

Conversely, those using social network build the trust profile of peers based on features of the social network graph. For instance, Sprout [18] relies on the fact that friends are expected to have a more reliable behavior than other ones. On the other hand, the technique introduced by Danezis et al. [7], is based on sending requests to peers which have appeared a few times in social graph paths, therefore requests are balaced over the system.

IV. SecureRoutingDHT

This section introduces SecureRoutingDHT, a protocol to provide reliability in the lookup process. In essence, this section presents decisions that were taken into account for constructing this protocol.

A. Routing Protocol Construction

The routing protocol is defined over a redundant structured topology that is organized into several groups of peers called *quorums*. These groups are connected among them and are constructed by augmenting the routing table. A *quorum* provides flexibility and diversity for selecting a peer during the lookup process, as well as storing multiples replicas of an object and cooperating among peers.

Each peer p in a redundant topology, maintains three levels for routing information:

1) Peers within its *quorum*: the peer p has links to all peers in its quorum.
2) Peers in other *quorums*: For each contact peer, q, of p, it maintains links to all peers within the quorum of q.
3) Backpointers: p maintains pointers to the peers pointing to it.

The aforementioned construction suggests that the overlay structure is strongly connected (redundant). Hence, some of the attacks previously introduced are hardly to lunch. In fact, with this structure, each node can verify their links so as to detect lunched routing poisoning and unsolicited message attacks.

1) Protocol: SecureRoutingDHT makes usage of the *recursive* style routing; but, during the process, each peer is provisioned of a selection function (Reputation mechanism) that chooses the most reliable peer within the next *quorum*, to send the request. Finally, at penultimate node, the request is broadcasted to a subset of peers within the last *quorum*, thus resistance to storage and retrieval attacks is provided. Figure 1 illustrates, in a general fashion, how the routing process is performed by SecureRoutingDHT.

Let Q_k be a quorum at $k - th$ step of the routing process, R_{qp} be the reputation of peer q maintained by p and h be the average number of steps to reach the

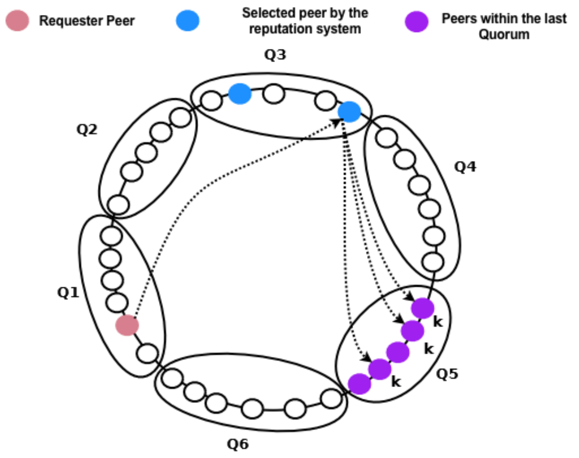
Requester Peer
Selected peer by the reputation system
Peers within the last Quorum
Q3
Q2
Q4
Q1
Q5
Q6
k
k
k

Figure 1. Routing Process

target. Now, let us suppose a peer $p \in Q_1$ (initiator) requests for key k, so the protocol works as follows:

- p constructs a set of peers, S , whose reputation value exceeds a threshold u (configuration parameter). Formally, $S = \{q \in Q_2 | R_{qp} \geq u\}$ and for completeness $S = Q_2$ if $S = \oslash$. Now, peer p randomly chooses a peer q_2 from S and sends the request. This random selection allows feeding the reputation mechanism.
- Let $p = q_2$ and repeat step 1 until reaching *quorum* Q_{h-1}.
- Finally, q_{h-1} sends the request to each node within a subset $D \subseteq Q_h$, which is responsible for storing k.

2) Routing Protocol Maintenance: Each overlay network needs a process to keep up to date the organization of the peers within it. This process is time consuming since it is performed frequently, but it is even more consuming in quorum based overlay networks because of its higher number of links among peers. However, there is a trade-off between security and performance.

Whenever a peer p is joining to the network, it contacts another peer q by sending its corresponding membership *token*. At that point, q validates the *token* sent by p. Afterwards q forwards a *join* message with identifier id_p. As soon as q receives the responsible *quorum*, it sends back to p. At that stage, p is able to notify to each neighbour, allowing them to update their routing information. Finally, p performs a similar process by sending multiple queries, which depends on the underlying DHT, in order to build of other *quorums*. Moreover, for each formed *quorum*, p notifies them, so as that they can update their *backpointer* information.

B. Reputation Mechanism

The proposed reputation mechanism was designed by taking into account three components suggested by Hoffman et al. [12]. These components are: (1) Formulation, (2) Calculation, and (3) Dissemination. The first one defines the reputation metric foundations and the information sources. In turn, the second one describes the formulae to compute the reputation of a peer, and finally, the third one defines the interaction mechanisms among peers.

1) ***Formulation****:* Since the routing process is *recursive*, a peer is only able to compute ratings in accordance with the success or failure of sent messages. Thus the reputation-updating process is realized by asking recommendations or through own interactions. Accordingly a 4-tuple (E_e, E_f, R_e, R_f) is defined, where variables E_e and E_f are two events, reprsenting if a message is forwarded successfully or not respectively. In turn, R_f and R_e are events representing whether a recommendation is considered as biased or non-biased respectively. As it can be noted, a peer only assigns reputation values to peers within its routing table.

2) ***Calculation****:* There are several ways to compute the reputation of a peer, namely, average ratings, trust models or Bayesian models and so forth [13]. On the one hand, models based on simple average are not appropriate since they do not allow representing the context adequately. On the other hand, trust models and Bayesian systems, which have been extensively studied and proved as equivalent, are more adequate because of their properties such as context evaluation, easy computation, extensibility in terms of number of variables and aging [14]. Moreover, Bayesian reputation systems are those based on the Dirichlet function, for that allowing the definition of several variables [14].

Let $X = \{X_1, X_2, \ldots, X_k\}$ be the set of k random variables, which represent the events of the observations and p_i be the probability function for X_i which satisfies $\sum_{i=1}^{k} p_i = 1$.

The computation Dirichlet function is not practical; as a consequence this value is calculated as [14]:

$$\wp = \Sigma_{j=1}^{k} \tau_j \vec{S}(X_i) \qquad (1)$$

where $\vec{S}(X_i) = E(\vec{p}(X_i)|\vec{r},\vec{a}) = \frac{\vec{r}(X_i)+W\cdot\vec{a}(X_i)}{W+\Sigma_{j=1}^{k}\vec{r}(X_j)}$ is the expected value of X_i and values $1 \leq \tau_j \leq k$ are weights.

Note that, if $\tau_j = 1$ for all $1 \leq j \leq k$, $\wp$ will be equal one. In addtion, $\vec{a}$ is the base rate vector over the state space and W is a weight, which is typically set to 2, but W could be chosen higher in order to reduce the influence of new evidence over the base rate [14].

Observations are accumulated as a vector $\vec{R} = (R_1, R_2, \cdots, R_i, \cdots, R_k)$ by each peer. If an event which affects to variable X_i is detected, $\vec{R}$ will be updated by performing $\vec{R} = \vec{R} + I_i$, where I_i is the identity vector.

For aging observations, let $M_{y,t}$ be the set of peers that collect observations during a interval time *t* for an agent y, $\vec{r_y^x}$ be the vector of observations collected by x for y in the same interval. Now let $\vec{r_{y,t}}$ be the set of the total observations in the interval t for agent y, hence $\vec{r_{y,t}} = \Sigma_{x \in M_{y,t}} \vec{r_y^x}$. Furthermore, vector $\vec{R}$ can be updated by computing $\vec{R_{y,t}} = \lambda \vec{R_{y,t-1}} + \vec{r_{y,t}}$, where $0 \leq \lambda \leq 1$. As it can be noted, a higher value of λ gives more priority to historical data.

Finally, the reputation of a peer is calculated as follows

$$\wp = \tau_{E_e} \vec{S}(E_e) + \tau_{R_e} \vec{S}(R_e) + \tau_{E_f} \vec{S}(E_f) + \tau_{R_f} \vec{S}(R_f) \qquad (2)$$

where $\tau_{E_e} = 1$, $\tau_{R_e} = 0.5$ and $\tau_{E_f} = \tau_{R_f} = 0$. These assignments give a higher priority to successful messages, because they are performed more frequently.

*3) **Dissemination**:* There are two sources of information: direct and indirect. The former encloses the interactions which a peer has with other peers, and the latter comprises the provided recommendations from a peer.

The recommendation process builds a set of peers built from known *quorums* and asks for a recommendation to each peer within it of another peer. Each provided recommendation is sent back by using the *Piggyback* protocol. As soon as recommendations are received, those are classified as biased or non-biased by performing the following classification algorithm:

Suppose that p asks for recommendations for a peer r to a group of peers C_r. At that point, there is expected that each peer c within C_r sends to p the corresponding recommendation as a vector $\vec{R_{r,t}^c}$. As soon as a defined time has elapsed, p computes the local reputation value of r, $\wp_r^p$, as well as $\wp_r^c$ for each received recommendation from $c \in C_r$. These values are computed with $\vec{R_{r,t}^c}$ and the local base vector $\vec{a}$. Afterwards, p computes the following

$$\sigma = \sqrt{\frac{\Sigma_{c \in C_r} (\wp_r^p - \wp_r^c)^2}{|C_r|}} \qquad (3)$$

Now, let us consider the interval $I = [\wp_r^p - k \cdot \sigma, \wp_r^p + k \cdot \sigma]$, where k is a positive constant that generally is setted to 1. The classification method arranges each received-recommendation $\wp_r^c$, as biased or non-biased, if $\wp_r^c$ is within the interval or not respectively. For those peers which sent a considered-biased recommendation, the corresponding variable R_f is incremented by 1, otherwise the corresponding variable R_e is incremented by 1.

Furthermore, a new set of recommendations, E, is formed with each one of received recommendation considered as non-biased- these recommendations are represented as a vector-. From the set E, p only chooses a few recommendations in order to avoid that colluding peers try to overstimate/understimate the reputation value of another peer; and updates the corresponding reputation value by computing $\vec{R_{r,t}^p} = \vec{R_{r,t}^p} + \Sigma_{c \in H_r} (\wp_c^p \cdot \vec{R_{r,t}^c})$.

Besides of mentioned components, it is important to define a mechanism to reduce the impact of churn to the reputation system. In fact, a peer can take of advantage of the joining/leaving process to gain a new reputation value [25]. Therefore, a mechanism that alleviates this threat must be implemented.

A possible solution is to use the same system to store these values, even though this would imply an increment of the number of messages, as well as adressing new concerns - those related to data availability, integrity, privacy and access controls [22]. Therefore, this sort of solution is not appropiate.

As a consequence, another strategy is implemented which takes advantage of the fact that several Sybil attack solutions assign a fixed identifier to each peer [1][4][16][17]. Following this, it is likely that the set of backpointers of the joining peer would be the same, consequently a local cache is proposed in order to store calculated reputation values of the off-line neighbours.

Since cache size is finite, the implemented replacement policy only maintains reputation values of those peers which are likely to rejoin to the system (LRU-based). Each peer is assigned a default estimated off-line period, called PER_0 at first time. In this way, whenever a peer leaves/joins the system, its backpointers peers calculate a off-line period PF and update the correspoding PER by computing PER_{i+1}=PER_i $\times$ α+$PF \times \beta$, where α, β are weights which tipically are set to 0.2 and 0.8, respectively [3].

V. Evaluation

On this section is presented an analysis of our protocol regarding the number of messages during its operations, as well as the probability of success whenever messages are forwarded.

A. Theoretical Analysis

Theoretical analysis is presented regarding number of involved messages in the forwarding and maintanance proccess, as well as the expected probability for that a message reaches to responsible peers.

1) Number of messages: Suppose that $q_1 \in Q_1$ is searching the responsible nodes of one resource with id k. Let $Q_1, Q_2, \cdots, Q_h$ involved *quorums* during the routing process. Note that h depends on the underlying P2P DHT-based system. Moreover, let $D \subseteq Q_h$ be the set of peers storing key k. Hence, the expected number of messages to reach D is equal to $h - 1 + |D|$. Particularly, If Chord is the underlying system, there holds that $h = O(\log_2 n)$.

TABLE I
EXPECTED NUMBER OF MESSAGES

Strategy	Number of expected messages	Error
Chord [30]	$\log_2 n$	0
Replica [11]	$2^{\lvert D\rvert-1}\log_2 n$	$\log_2 n\cdot(2^{\lvert D\rvert-1}-1)$
S-Chord [10]	$\lvert D\rvert^2\log_2 n$	$\log_2 n(\lvert D\rvert^2-1)$
QuorumP1 [36]	$2\cdot\lvert D\rvert\log_2 n$	$\log_2 n(2\cdot\lvert D\rvert-1)$
QuorumP2 [36]	$\frac{(\log_2 n-1)}{1-f}+2\cdot\lvert D\rvert$	$\frac{f\cdot\log_2 n-1}{1-f}+2\cdot\lvert D\rvert$
Secure Routing DHT	$\log_2 n-1+\lvert D\rvert$	$\lvert D\rvert-1$

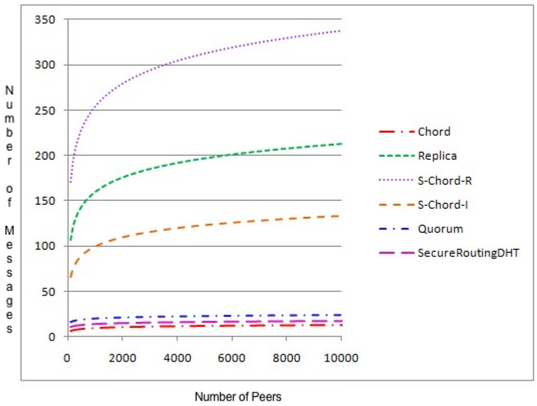

Figure 2. Expected number of messages

Table I presents formulae for strategies analyzed and its corresponding error. This value is defined to be equal to the difference between the strategy number of messages and Chord number of messages. Moreover, Figure 2 shows how the number of messages is reduced by SecureRoutingDHT due to usage of the reputation mechanism. Note that results are roughly equivalent to Chord, when number of peers grows.

2) Maintenance: The expected number of required messages during the maintenance process is derived from sum up each message involved in the update of the routing table. On the one hand, the number of expected messages to obtain the corresponding *quorum* is $(h-1)+(2r+1)$. On the other hand, the number of expected routing contacts depends on the underlying DHT, say, C_f, hence the expected amount of messages is:

$$C_f\cdot(h+2r) \tag{4}$$

When Chord is the underlying DHT, roughly $C_f = h = \log_2 n$. Moreover, if $2r = \log_2 n$, as in Myrmic [34], the expected number of messages is $2(\log_2 n)^2$, namely, the complexity is $O((\log_2 n)^2)$. It is important to notice that $2r = \log 2n$ is a value that increments failure tolerance, so it is an acceptable value.

3) Tolerance to malicious peers: This subsection compares the proposed protocol with other approaches regarding the probability of success when a message is forwarded, namely, the measure of its reliability.

From threat model, it can be seen that a peer is malicious with a probability at most f. Hence that probability of E_1, the event representing a path with h hops and not containing malicious nodes, is

$$Pr(E_1)=(1-f)^h \tag{5}$$

Let us consider a multi-path based strategy in where a message is sent through d independent paths. Furthermore, let X_2 be a random variable that represents the number of paths that does not contain any malicious nodes. Therefore, the failure probability of a multi-path based strategy is given by *Pr(fail)≤Pr(*X_2*=0)*.

As it is known that a free-malicious path has probability $Pr(E_1)=(1-f)^h$, then the probability that a path contains at least a malicious node is $1-Pr(E_1)$. Therefore, the probability of each path would be non-free-malicious is given by $(1-Pr(E_1))^d$. Finally, the probability that at least a path is free-malicious, $Pr(E_2)$, is given by :

$$Pr(E_2)=Pr(0\leq X_2)=1-(1-(1-f)^h)^d \tag{6}$$

Conversely in wide-path-based strategies, a message is successfully forwarded if at least one peer within each intermediate *quorum* is not malicious. Let E_3 be the event that one message has been forwarded successfully. It is clear that the probability that, in a *quorum* of size d, will be there at least one non-malicious peer is $1-f^d$. Therefore,

$$Pr(E_3)=(1-f^d)^h \tag{7}$$

For our protocol, analysis is based on that introduced in [26]. Let α be the probability that the reputation mechanism excludes an honest peer and β be the probability that the reputation mechanism chooses a malicious peer. Furthermore, let D_i be a set of size d and E_4 be the event that a malicious peers is selected from a set D_i of size d. Finally, let E_5 be the event that a peer is selected from a quorum by the reputation systems.

Evidently $Pr(E_4)=\frac{f\cdot d}{d}=f$ and $Pr(E_5)=(1-\alpha)(1-f)+f\beta$, where $(1-\alpha)(1-f)$ and $f\beta$ are the probabilities of choosing a honest and malicious peer by the reputation system respectively. Consequently, the probability of the event of choosing a malicious peer in the set D_i given that the reputation system has already chosen one, namely, $\gamma = Pr(E_4|E_5)$ is:

$$\gamma=\frac{Pr(E_4\cap E_5)}{Pr(E_5)}=\frac{f\beta}{(1-\alpha)(1-f)+f\beta} \tag{8}$$

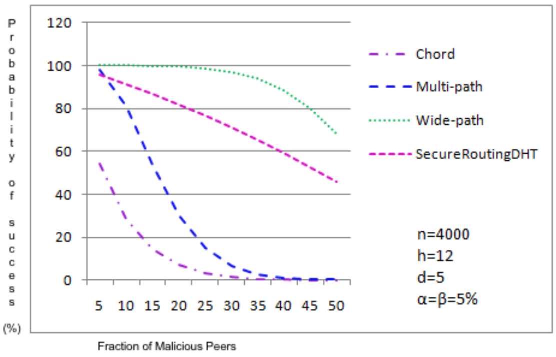

Figure 3. Probabilities of success

As it can be noted, $1-\gamma$ represents the probability of choosing a non-malicious peer within D_i. Thus, the probability of success of SecureRoutingDHT is $(1-\gamma)^{h-1}$. This new equation is similar to equation (5) and as can be seen, γ must be smaller than f in order to increment the probability of success. Suppose that $\gamma < f$ and $0 < f, \alpha, \beta < 1$, then :

$$\begin{array}{rcl} \frac{f\beta}{(1-\alpha)(1-f)+f\beta} & < & f \\ \frac{\beta}{(1-\alpha)(1-f)+f\beta} & < & 1 \\ \beta - f\beta & < & (1-\alpha)(1-f) \\ \beta(1-f) & < & (1-\alpha)(1-f) \\ \beta & < & 1-\alpha \end{array} \qquad (9)$$

The above means that whether β is sufficiently small, the probability of success of SecureRoutingDHT will increment. For instance, setting to $f = 0,25$, $n = 4000$, $h = \log_2 n$ and $\alpha = \beta = 0,05$, the probability of success of SecureRoutingDHT is 82%. Figure 3 shows the probability of success of the strategies analysed.

As it already has been shown, strategies based on wide-path have a higher probability of success. However, the number of sent messages is higher than other strategies (Figure 2). In turn, the introduced protocol provides an acceptable probability of success while using a smaller number of messages, moreover it can tolerate theoretically up to a fraction of 35% of malicious peers.

B. Simulation

Simulations were performed by using Overlay-Weaver [31]. These were carried out during a period of time, where relevant information was collected in order to measure the number of messages and the probability of success.

The deployed scenarios to evaluate the protocol are described below.

1) *Scenario 1 (scalability)*: Deploying up to 4000 nodes and issuing of requests for a selected key in order to compute the average of number of messages per query.

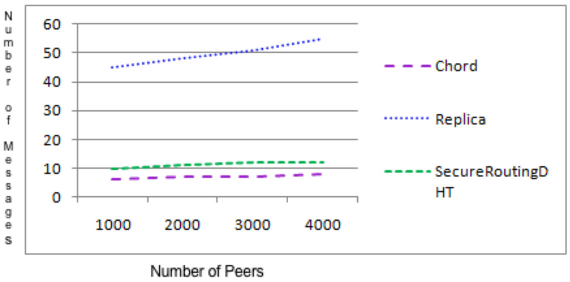

Figure 4. Number of messages in the simulation

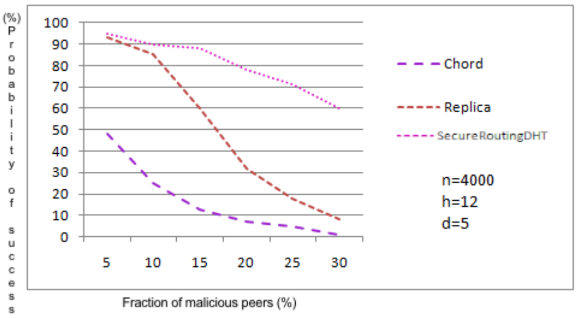

Figure 5. Probability of success in the simulation

2) *Scenario 2 (tolerance)*: Deploying 4000 nodes and uniformly distributing a fraction f of malicious peers, in order to evaluate the probability of success if f is incremented.

Next results are presented according to the scalability and tolerance of malicious peers.

Scalability: The test was performed by choosing random peers and a key k over the system. Each random chosen peer issues a request for the key k and finally the average of messages per query is computed. Figure 4 shows obtained results for Chord, Replica and SecureRoutingDHT. As can be noted, results support the scalability of SecureRoutingDHT in terms of the number of messages.

Tolerance to malicious peers: The test was performed by uniformly distributing a fraction f of malicious peers over the system, namely, $f \cdot 4000$ peers are randomly chosen and considered as malicious. In this scenario a malicious peer does not cooperate with the routing process.

Figure 5 shows results that are obtained for Chord, Replica and SecureRoutingDHT. As it can be noted, the introduced protocol exceeds the probability of success than those guaranteed by Chord and Replica and to tolerate up to 30% of malicious peers, which is an acceptable value due to the fact that solutions to the*Sybil* attack try to limit the number of misbehaivours peers.

VI. Conclusions and Future Work

P2P systems were created without any security considerations; thus, there are a lot of attacks against them, such as *sybil*, eclipse, routing, storage and retrieval attacks. As the routing process is one of the most important mechanisms within the context of P2P systems, this paper addressed threats against this process.

The paper introduces SecureRoutingDHT, a reliable routing protocol that aims to mitigate routing attacks and provide direct access to all replicas of a requested resource. This protocol is compatible with several solutions to the *sybil* attack and it is decoupled from the underlying P2P DHT-based system. As well as reduces the number of messages in comparison with those consumed in S-Chord [10], Replica [11] and Quorum [36].

Furthermore, a theoretical and practical (through simulations) analysis of the protocol are presented, concerning its scalability in terms of number of sent messages and tolerance to malicious peers. Particularly, when SecureRoutingDHT is built on top of Chord, it was theoretically shown that the expected number of messages is $\log_2 n - 1 + |D|$, as well as that the expected number of sent messages during the maintenance protocol is $\log_2 n \cdot (\log_2 n + 2r)$. The above evidences the dependency to churn rates. Finally, as for the reliability, the benefits that were obtained are significant, since that our protocol behaves fairly well up to a 30% percentage of malicious nodes.

Finally, it would be interesting to evaluate performance and probability of success of proposed protocol whenever *iterative* routing is implemented. Additionally, consider other possible mechanisms to obtain recommendations, indeed, there can be taken advantage of back-pointer information for enriching the recommendation process. There is a need for an implementation of this protocol, as well as a set of software libraries, in order that there could be built new applications that take advantage of it.

References

[1] I. Baumgart and S. Mies, ”S/Kademlia: A practicable approach towards secure key-based routing”, Proc. of the 13th Int. Conf. on Parallel and Distributed Systems, IEEE Press, 2007, pp. 1-8, doi: 10.1109/ICPADS.2007.4447808.

[2] K. Butler, S. Ryu, P. Traynor, and P. McDaniel, ”Leveraging identity-based cryptography for node ID assignment in structured P2P systems”, Transactions on Parallel and Distributed Systems, IEEE Press, 2009, pp. 1803-1815, doi: 10.1109/TPDS.2008.249.

[3] H. Cai, J. Wang , D. Li, and J. Deogun ”A novel state cache scheme in structured P2P systems”, Journal of Parallel and Distributed Computing, Academic Press, 2005, pp. 154-168, doi: 10.1016/j.jpdc.2004.09.005.

[4] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, ”Secure routing for structured peer-to-peer overlay networks”, Proc. of the 5th Symposium on Operating Systems Design and Implementation, ACM Press, 2002, pp. 299-314, doi:10.1145/844128.844156.

[5] C., Chan, S. Chan, ”Distributed Hash Tables: Design and Applications”, Handbook of Peer-to-Peer Networking, Springer Science, 2010, p. 257-280, doi:10.1007/978-0-387-09751-0_10.

[6] T. Condie, V. Kacholia, S. Sankararaman, J. Hellerstein and P. Maniatis, ”Induced Churn as Shelter from Routing-Table Poisoning”, Proc. of the 13th Symposium on Network and Distributed System Security, 2006.

[7] G. Danezis, C. Lesniewski-Laas, M. Kaashoek, and R. Anderson, ”Sybil-resistant DHT routing”, Proc. of the 10th European Symposium On Research In Computer Security, Springer, 2005, pp. 305-318, doi:10.1007/11555827_18.

[8] J. Douceur, ”The sybil attack”, Revised Papers from the 1st International Workshop on Peer-to-Peer Systems, Springer, 2002, pp. 251-260, doi: 10.1007/3-540-45748-8_24.

[9] N. Fedotova, M. Bertucci, and Veltri, ”Reputation management techniques in DHT-based peer-to-peer networks”, Proc. of the 2nd Int. Conf. on Internet and Web Applications and Services, IEEE Press, 2007, pp. 4, doi: 10.1109/ICIW.2007.53.

[10] A. Fiat, J. Saia, and M. Young, ”Making chord robust to byzantine attacks”, Proc. of the 13th Annual European Symposium on Algorithms, Springer, 2005, pp. 803-814, doi: 10.1007/11561071_71.

[11] C. Harvesf and D. Blough, ”Replica placement for route diversity in tree-based routing distributed hash tables”,Transactions on Dependable and Secure Computing, IEEE Press, 2009, doi: 10.1109/TDSC.2009.49.

[12] K. Hoffman, D. Zage, and N. Nita-Rotaru, ”A survey of attack and defense techniques for reputation systems”, Computing Surveys, ACM, 2009, pp. 1-31, doi: 10.1145/1592451.1592452.

[13] A. Jøsang, R. Ismail, and C. Boyd, ”A survey of trust and reputation systems for online service provision”, Decision Support Systems, Elsevier Science Publishers, 2007, pp. 618-644, doi: 10.1016/j.dss.2005.05.019.

[14] A. Jøsang, and W. Quattrociocchi, ”Advanced features in bayesian reputation systems”, Proc. of the 6th Int. Conf. on Trust, Privacy & Security in Digital Business, Springer, 2009, pp. 105-114, doi: 10.1007/978-3-642-03748-1_11.

[15] A. Kapadia and N. Triandopoulos, ”Halo: High-assurance locate for distributed hash tables”, Proc. of the 15th Symposium on Network and Distributed System Security, 2008.

[16] F. Lesueur, L. Mé, and V. Triem Tong, ”A sybilproof distributed identity management for P2P networks”, Proc. of Symposium on Computers and Communications, IEEE, 2008, pp. 246-253, doi: 10.1109/ISCC.2008.4625694.

[17] B. Levine, C. Shields, and B. Margolin, ”A Survey of Solutions to the Sybil Attack”, Technical Report, University of Massachusetts, 2006.

[18] S. Marti, P. Ganesan and H. Garcia-Molina, ”DHT routing using social links”, Revised Selected Papers from the 3rd International Workshop on Peer-to-Peer Systems, Springer, 2004, pp. 100-111, doi: 10.1007/978-3-540-30183-7_10.

[19] M. Naor and U. Wieder ”A simple Fault Tolerant Distributed Hash Table”, Revised Papers from the 2nd Int. Workshop on Peer-to-Peer Systems, Springer, 2003, pp. 88-97, doi: 10.1007/978-3-540-45172-3_8.

[20] K. Needels and M. Kwon, ”Secure routing in peer-to-peer distributed hash tables”, Proc. of the Symposium on Applied Computing, ACM, 2009, pp. 54-58, doi: 10.1145/1529282.1529292.

[21] B. Roh, O. Kwon, S. Hong, and J. Kim, ”The exclusion of malicious routing peers in structured P2P systems”, Proc. of the 5th Int. Workshop on Agents and Peer-to-Peer Computing, Springer, 2006, pp. 43-50, doi: 10.1007/978-3-540-79705-0_4.

[22] C. Roncancio, M. Villamil, C. Labbé, and P. Serrano-Alvarado, ”Data Sharing in DHT Based P2P Systems”, Transactions on Large-Scale Data- and Knowledge-Centered Systems, Springer, 2009, pp. 327 - 352, doi: 10.1007/978-3-642-03722-1_13.

[23] M. Sánchez-Artigas, P. García-López, and A. Gómez, ”A novel methodology for constructing secure multi-path overlay”, Internet Computing, IEEE Press, 2005, pp. 50-57, doi: 10.1109/MIC.2005.117.

[24] M. Sánchez-Artigas, P. García-López, and A. Gómez, ”By-pass: providing secure DHT routing through bypassing malicious peers”, Proc. of Symposium on Computers and Communications, IEEE Press, 2008, pp. 934-941, doi: 10.1109/ISCC.2008.4625618.

[25] M. Sánchez-Artigas and P. García-López, ”On routing in distributed hash tables: is reputation a shelter from malicious behavior and churn?”, Proc. of the 9th Int. Conf. on Peer-to-Peer Computing, IEEE Press, 2009, pp. 31-40, doi: 10.1109/P2P.2009.5284546.

[26] M. Sánchez-Artigas, P. García-López, and A. Gómez, ”Secure forwarding in DHTs-is redundancy the key to robustness?”, Proc. of the 14th International European Conference on Parallel and Distributed Computing, Springer, 2008, pp. 611-621, doi: 10.1007/978-3-540-85451-7_65.

[27] A. Singh, T. Ngan, P. Druschel, and D. Wallach, ”Eclipse attacks on overlay networks: Threats and defenses”, Proc. of the 25th Int. Conf. on Computer Communications, IEEE Press, 2006, pp. 1-12, doi: 10.1109/INFOCOM.2006.231.

[28] E. Sit and R. Morris ”Security considerations for peer-to-peer distributed hash tables”, Revised Papers from the 1st Int. Workshop on Peer-to-Peer Systems, Springer, 2002, pp. 261-269, doi: 10.1007/3-540-45748-8_25.

[29] M. Srivatsa and L. Liu, ”Vulnerabilities and security threats in structured overlay networks: A quantitative analysis”, Proc. of the 20th Annual Computer Security Applications Conference, IEEE Press, 2004, pp. 252-261, doi: 10.1109/CSAC.2004.50.

[30] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, ”Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, Proc. of the Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, 2001, pp. 149-160, doi: 10.1145/383059.383071.

[31] K. Shudo, Y. Tanaka, and S. Sekiguchia, ”Overlay Weaver: An overlay construction toolkit”, Computer Communications, Elsevier, 2008, pp. 402-412, doi: 10.1016/j.comcom.2007.08.002.

[32] G. Urdaneta, G. Pierre, M. Van Steen, ”A survey of DHT security techniques”, Journal ACM Computing Surveys, ACM Press, 2011, Volume 43 Issue 2, doi:10.1145/1883612.1883615.

[33] R. Villanueva, M. Villamil and R. Arnedo, ”Secure routing strategies in DHT-based systems”, Proc. of the Third Int. conf. on Data management in grid and peer-to-peer systems, Springer, 2010, pp. 62-74, doi:10.1007/978-3-642-15108-86.

[34] P. Wang, L. Osipkov, N. Hopper, and Y. Kim, ”Myrmic: secure and robust DHT routing”, Technical report, University of Minnesota-Twin cities, 2006.

[35] X. Xiang, and T. Jin, ”Efficient secure message routing for structured peer-to-peer systems”, Proc. of the Int. Conf. on Networks Security, Wireless Communications and Trusted Computing, IEEE Press, 2009, pp. 354-357, doi: 10.1109/NSWCTC.2009.124.

[36] M. Young, A. Kate, I. Goldberg, and M. Karsten, ”Practical robust communication in DHTs tolerating a byzantine adversary”, Proc. of the 30th Int. Conf. on Distributed Computing Systems, IEEE Press, 2010, pp. 263 - 272, doi:10.1109/ICDCS.2010.31.