Evaluation of Sybil Attacks Protection Schemes in KAD

Thibault Cholez, Isabelle Chrisment, Olivier Festor

▶ **To cite this version:**

Thibault Cholez, Isabelle Chrisment, Olivier Festor. Evaluation of Sybil Attacks Protection Schemes in KAD. 3rd International Conference on Autonomous Infrastructure, Management and Security - AIMS 2009, University of Twente, Jun 2009, Enschede, Netherlands. pp.70-82, 10.1007/978-3-642-02627-0_6 . inria-00405381v2

HAL Id: inria-00405381

https://inria.hal.science/inria-00405381v2

Submitted on 21 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Evaluation of Sybil Attacks Protection Schemes in KAD

Thibault Cholez, Isabelle Chrisment and Olivier Festor

MADYNES - INRIA Nancy-Grand Est, France
{thibault.cholez,isabelle.chrisment,olivier.festor}@loria.fr

Abstract. In this paper, we assess the protection mechanisms entered into recent clients to fight against the Sybil attack in KAD, a widely deployed Distributed Hash Table. We study three main mechanisms: a protection against flooding through packet tracking, an IP address limitation and a verification of identities. We evaluate their efficiency by designing and adapting an attack for several KAD clients with different levels of protection. Our results show that the new security rules mitigate the Sybil attacks previously launched. However, we prove that it is still possible to control a small part of the network despite the new inserted defenses with a distributed eclipse attack and limited resources.

Key words: P2P networks, DHT, Sybil attack, security, defense, KAD

1 Motivation and scope of work

Peer-to-Peer (P2P) networks have proven their ability to host and share a large amount of resources thanks to the collaboration of many individual peers. They are known to have many advantages compared to the client-server scheme: P2P networks scale better; the cost of the infrastructure is distributed and they are fault tolerant. Most currently deployed structured P2P networks are based on Distributed Hash Tables (DHT). Each peer is responsible for a subset of the network. This organization improves the efficiency of P2P networks ensuring routing in O(log n) but can also lead to security issues like the Sybil Attack.

The Sybil Attack, as described by Douceur [3], consists in creating a large number of fake peers called the "Sybils" and placing them in a strategic way in the DHT to take control over a part of it. Douceur proved that the Sybil Attack cannot be totally avoided as long as the malicious entity has enough resources to create the Sybils. This problem was not considered when designing most of the major structured P2P networks. In this context, the goal of the defense strategies described in the literature is to limit the Sybil Attack as completely stopping it is impossible. Proposed solutions are based on a strong identification of the peer [2] or on a trusted central authority [1]. In [8], the authors propose to bound the degree of overlay nodes by anonymous auditing to limit localised attacks but they also rely on a central certification to limit the number of Sybils. A strong identification being not adapted to many P2P applications, other strategies are

possible to limit the Sybil attack considering that a distributed assignment with free identifiers is possible but, in compensation, must be verified by resource consuming proofs [7] or an underlying social network [10]. The most successfully completed protection against the Sybil attack [4] uses distributed certification coupled with a social network, but no studies consider practical protections set in a real network.

Even if defense against the Sybil attack has been largely investigated, most of the proposed solutions are difficult to set up and are not always well suited for the P2P paradigm. Things are even worse when considering all the constraints of a large public P2P network like KAD without a managed support infrastructure and with the need of backward compatibility. KAD provides a file sharing application based on the academic Kademlia [5] DHT and has been proven vulnerable and totally unprotected against Sybil attacks that can highly affect the network [9]. In this context, managing and protecting the KAD network seems very challenging.

The latest versions of the major KAD clients have introduced new protection mechanisms to limit the Sybil attack, making the previous experiments concerning the security issues of KAD like [9] inefficient. These newly implemented protection mechanisms have neither been described nor been evaluated and assessed. The purpose of this study is to evaluate the implemented security mechanisms against real attacks. We will then be able to have an updated view of KAD vulnerabilities: Is the network still vulnerable to the attack proposed in [9]? Is it now fully protected? Which vulnerabilties have been corrected and which ones keep being a threat? This work is a first and necessary step to design improved defense mechanisms in future works. As far as we know, this paper is also the first attempt to experiment and assess practical protections set by the P2P community to protect a real network.

This document is structured as follows. Section 2 describes the background of KAD, the Sybil attacks performed on it, and the new defense mechanisms. We then present in Section 3 our evaluation of the mechanisms including our methodology, the results obtained by the different versions of KAD clients and the impact of our distributed eclipse attack. Finally, Section 4 concludes the paper with a discussion on the validity of the built-in protections and outlines our future works.

2 KAD and the Sybil attack

2.1 Overview of KAD

KAD is a structured P2P network based on the Kademlia distributed hash table routing protocol [5]. KAD is implemented by the popular eMule file sharing application and its multi-platform version aMule, both are open source. Historically, these clients were mainly designed to join the eDonkey network composed of autonomous distributed servers. However, many major eDonkey servers have been closed by lawsuits so that the eDonkey network is becoming less attractive and users are progressively migrating toward the fully decentralized P2P

network, KAD. With an estimated number of concurrent online users around 4 millions, KAD is the widest deployed structured P2P network.

Each node of KAD has a 128bits KADID defining its position in the DHT. Two types of requests are used to discover the network: the *Hello_REQ* is used by a peer to announce itself and the *Kademlia_REQ* is used to discover new peers toward a specific address. The routing is based on the XOR metric: to fill the routing table, each node registers at level **i** K-contacts (called a K-bucket), each contact being at a distance between 2^{128-i} and 2^{127-i} from its KADID regarding the XOR metric. The deeper the contact is in the tree, the closer it is to the node and the better the peer knows this part of the DHT; this provides routing in O(log n). The routing is done in an iterative way with parallel lookups: asking at first the **n** closest contacts to the target ID found in the routing table for even closer nodes toward the target with *Kademlia_REQ*, waiting for their responses and then reiterating.

As a file sharing application, the purpose of the DHT is to retrieve information like keywords and files. When sharing a new file, the binary and all the keywords are hashed separately with a MD5 function generating KADIDs and then published. The peers in charge of a file or a keyword are those close enough to their hash. This distance is called the tolerance zone and is set to the first common 8bits (most significant bits). The double indexation allows retrieval of a particular file, given a set of keywords. To publish a file, two types of requests are sent:

- *KADEMLIA2_PUBLISH_KEY_REQ* requests are sent to the hash of the keyword and associate a keyword with a file
- *KADEMLIA2_PUBLISH_SOURCE_REQ* requests are sent to the hash of the file and associate a file to a source (a node able to upload the file)

2.2 Previous attacks in KAD

Recent investigations showed that KAD can be cheated in several ways using few resources and resulting in important outages.

Steiner et al [9] were the first to successfully launch a real Sybil attack on KAD, resulting in the full control of a part of the network. The attack is divided in two steps. The first step consists in a crawler gathering information about what peer is in a zone of the network and what its routing table is. This is done by issuing route requests (*Kademlia_REQ*) toward some predefined node IDs in order to obtain new contacts close to those IDs. When almost all nodes of the zone are known by the crawler, the second step consists in individually contacting each node of the list to pollute its routing table with Sybils. Previously trivial, injection of Sybils is now protected by several mechanisms studied in the forthcoming sections.

In [6], the authors describe an attack that allows denial of service on a great part of the network with few resources. This attack does not rely on a classical injection of Sybils but hijacks the references of contacts in the routing table. Malicious nodes are able to overwrite a legitimate routing table entry (KADID/IP

address) with their own reference by sending a simple *Hello_REQ* announcing this KADID. This weakness highlights a real lack of identity management in KAD. Thanks to the information acquired by a crawler, this vulnerability allows to partition the network or to do a massive denial of service. A search request starts from bad references and is then kept by them, until it terminates without any real result, or fails because of a timeout. The experiments on PlanetLab show that the attack is effective and does not use much bandwidth with some optimizations. The authors also noticed the improvements of the latest versions of the clients which could mitigate their attack, but they did not evaluate them.

2.3 Protections in KAD

The importance of these weaknesses forced the developers to react by setting up new protections in the latest versions of the clients. Even if older clients are still vulnerable, progressive updates should be able to mitigate the attacks and constitute a healthy basis for the network. KAD clients implement different levels of protection regarding their version. For similar versions, eMule and aMule are based on the same code and behave in the same way. Table 1 sums up the protections implemented in each of the clients.

Clients	Flood protection	IP limitation	IP verification
eMule 0.48a / aMule 2.1.3	No	No	No
eMule 0.49a / aMule 2.2.1	Yes	Yes	No
eMule 0.49b / aMule 2.2.2	Yes	Yes	Yes

Table 1. Protection enabled according to the client version

No protection: aMule 2.1.3 was chosen as a client without any protection against the Sybil attack. Basically, aMule accepts all incoming messages without any check: requests can be sent as fast as the client can handle them. There is no restriction regarding the KADID or the IP address of the sender. As long as the sender of a request matches a place left in a K-bucket, it can be added. With the KADID of the target and a good knowledge of the routing table mechanisms, it is possible to quickly take control of a legitimate peer.

Packet tracking and flood protection: The packet tracking function records the history of recently received and sent packets (12 minutes). This history is used for two purposes. The first goal is to be able to detect and drop unsolicited request answers. The second purpose is to set up a protection against flooding. With the history and time of previously received packets, it is possible to detect if an IP address is a source of flooding. For each type of KAD request, a threshold is defined per peer under which the number of received requests is considered

legitimate. If the threshold is not highly exceeded, the packets over the limit are simply dropped. If the rate of received requests is more than five times above the limit, the sender is considered as an attacker and his IP address is banned from the client.

IP address limitation: The IP address limitation tries to mitigate the Sybil attack, considering that public IP addresses are more or less restricted per participant. Before adding a new peer in the routing table, its IP address is checked and dropped if it is already used. This verification limits to one the number of KADID per IP address in the routing table. This really helps protecting a client against a Sybil attack which needs a lot of public IP addresses to be significant. The IP address limitation also takes care of IP addresses from the same /24 subnetwork or smaller. In fact, at most ten IP addresses from the same /24 subnetwork can be added to the routing table to prevent a single entity with many resources to be able to launch an attack. These ten peers are also constrained to be in different K-buckets of the routing table, or in other words to have spaced KADIDs. This is done to prevent the entity owning a /24 subnet to launch a localized attack which needs to have Sybils very close to the target ID. This protection does not affect NATed users which can fully participate to the network with the very unlikely constraint that two NATed peers with the same public IP address can not be referenced in the same routing table of another contact.

Identities verification: The last KAD clients implement even more protections concerning identity management. They aim to limit the identity spoofing (both IP address and KADID) between peers. Before adding a contact, KAD now uses a three-way handshake for new contacts, making sure they do not use a spoofed IP. Older version are verified with a similar check as presented in table 2. Oldest clients are checked with a second Hello_REQ - Hello_RES exchange including a random KADID as a challenge for the responding peer. Newer clients use an exchange of PING and PONG messages which add cryptographic identification. Finally, the last versions do the three-way handshake in the most effective way with the specially added new message Hello_RES_ACK.

Clients	Sequence	Nb of messages	Encryption
eMule 0.48a	Hello_REQ - Hello_RES - Hello_REQ(challenge) - Hello_RES(challenge)	4	No
eMule 0.49a	Hello_REQ - Hello_RES - PING - PONG	4	Yes
eMule 0.49b	Hello_REQ - Hello_RES - Hello_RES_ACK	3	Yes

Table 2. Three-way handshake is achieved differently regarding the client version

Moreover, KAD contacts are only able to update themselves in other's routing tables if they provide the proper private key corresponding to the public key

initially presented (supported by 0.49a+ nodes), in order to make it impossible to hijack them. Finally, contacts which fail the challenge are marked unverified and are not used for routing tasks. Theses protections avoid attacks using IP-spoofing and contact overwriting. It is a response to the routing attack issues described in [6] and one of the solutions proposed by the authors.

3 Evaluation of the protection mechanisms

In this section we evaluate different KAD clients and the mechanisms described above when facing a Sybil attack.

3.1 Methodology

As the injection of Sybils is the basis for attacking the network, we measure how permeable the routing table of a single peer is regarding the different levels of protection, and how restricted in their possibilities are the Sybils when they pass the protections. To evaluate the behavior of the defense mechanisms, we developed a basic attack software with the initial objective of polluting the routing table of a single peer running an unprotected version of KAD client. We then incrementally added the protection mechanisms by evaluating newer versions of the client and modified our attack accordingly.

For some of our experiments, our Sybil attack does not target the entire network as the one in [9] but only aims a single client. In fact, the protection mechanisms that we want to study are local to the client and do not require any interaction between peers. In consequence, the results obtained when evaluating a single client are sufficient and can be easily generalized as a good indicator of what the network will be in the future when most of them will be up to date.

Our attack announces Sybils by sending many *Hello_REQ* with forged KADIDs. The main difficulty of this attack is to announce the right identity for each Sybils to match the algorithm filling the routing table of the target and as consequence, maximize the pollution. Designing the generation of Sybil identities requires knowledge of the structure and algorithms of KAD routing table.

Looking at the code of KAD, the real structure of its routing table is slightly different from the proper binary tree defined in Kademlia [5]. First of all, all leafs can be split until level 4 without restriction, resulting in 16 K-Buckets covering the entire address space and used to route the farthest destinations. Then, at the level 4, the closest K-buckets with an index below 4 (which means that considering the 4 first bits: KADID XOR ContactID <= 4) can keep splitting until having an index of 8 or 9 by involving further bits as shown in figures 1. The depth of the routing table is not hardly limited but is limited de facto by the size of the network because fewer contacts are possible for deep buckets whose split becomes unlikely.

With this knowledge of the routing table and a targeted KADID, preparing the attack consists in generating a set of proper masks so that: targetID XOR

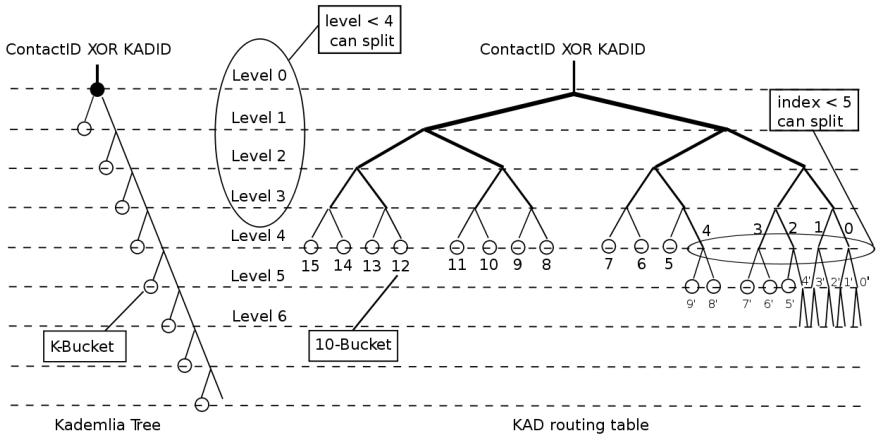

Fig. 1. KAD routing table scheme

mask = SybilID. The value of the masks are defined to generate SybilIDs progressively filling all the routing table of the targeted peer from the top to the bottom, by hitting the right K-bucket.

As we inject specifically forged contacts, they can hit deeper K-buckets in the routing table so that the size of the table under attack is higher than the size during a normal run. From the target point of view, the announcement of very close Sybils is considered as if the DHT space was more populated and consequently closer contacts easily findable.

3.2 Results

No protection: The first client attacked was aMule 2.1.3. This version is unprotected resulting to the realization of the simplest and most effective Sybil attack possible. Sybils KADIDs are forged as described previously and announced with a rate of 4 per second which is sufficient to populate the K-buckets faster that the normal algorithm without overloading the target client.

Actually, KAD contacts are periodically checked by sending an *Hello_REQ* and waiting for the *Hello_RES* to keep the routing table up to date. But this version also presents a kind of optimization that becomes a major design flaw when considering the Sybil attack. In fact, every *Hello_RES* from an IP address (whatever its KADID is) acknowledges every contact with this IP address. As all Sybils share the same IP address, a single *Hello_RES* aknowledges this IP address and is sufficient to keep hundreds of Sybils alive. Moreover, a peer can announce itself with a *Hello_RES* instead of a *Hello_REQ*, and still be added to the table. Following this discovery, our attack sends *Hello_RES* so that each new Sybil acknowledges and maintains the previous ones in the routing table without any message exchange.

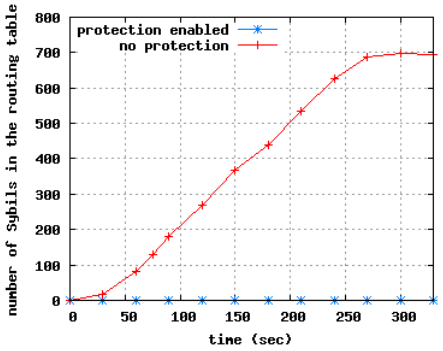

Fig. 2. Propagation of Sybils in the routing table

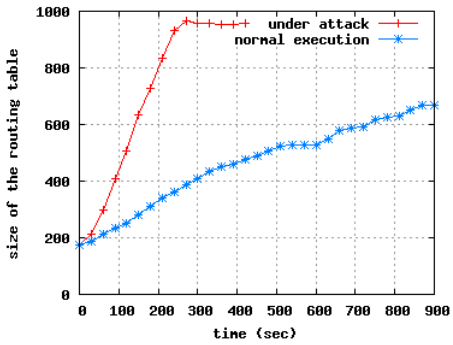

Fig. 3. Filling of the routing table under attack

The first results show that the routing table of the unprotected version is almost fully corrupted under attack (graph 2): at the end of an attack lasting 300s, the routing table of aMule counts 70% of Sybils (689 Sybils for 953 contacts). The major part of the routing table which is not polluted is made up of 200 good contacts saved from the previous execution and pre-loaded to bootstrap. Graph 3 shows at which speed the routing table of aMule is filled regarding if attacked or not. Even if the speed of *Hello_RES* sent by the Sybils was moderate, we can clearly see the Sybil attack as the routing table abnormally grows under the announcements of Sybils. Theses results confirm the need for the following protection mechanisms.

Flooding protection: The second experiment aims to evaluate the flood protection. So, we launched an attack against the eMule 0.49a client with the IP limitation disabled. To be successful, the attack had to be improved in two ways. First, the packet tracking used to implement the flood protection drops *Hello_RES* without previous *Hello_REQ* so that the weakness described above to maintain Sybils is not possible anymore. Therefore, we modified our attack to keep each Sybil alive by responding to the *Hello_REQ* messages from the victim. The problem is that *Hello_REQ* messages do not include the targeted KADID whereas *Hello_RES* messages need it. A normal peer just has to manage its own KADID so that it clearly knows what to respond to a *Hello_REQ* message. On the contrary, our Sybil attack manages multiple identities and does not know which KADID is addressed by the *Hello_REQ* message. To bypass this limitation and find with which KADID to answer, each Sybil communicates on a different UDP port so that we can retrieve the wanted KADID. With this modified attack, we are able to maintain Sybils alive in the long run. Secondly, we sent *Hello_REQ* messages every 30 seconds to be under the threshold.

Our experiments show that the flooding protection works as expected. The limitation rate before dropping *Hello_REQ* is set to 3 packets per minute and above 15 packets per minute, our IP address is banned. The results are shown

in graph 4. We can see that even if the attack is clearly slowed down, taking more than 8 hours to inject the same number of Sybils where previous attack only took few minutes, the routing table is still polluted with more than 60% of Sybils (540/873). The flooding limitation can protect the majority of short connections to the KAD network, but longer connections still suffer from a widely infected routing table.

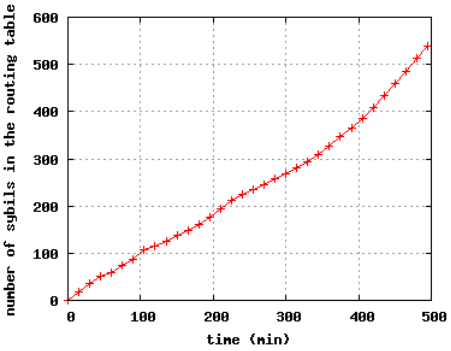

Fig. 4. Propagation of Sybils in the routing table with flood protection enabled

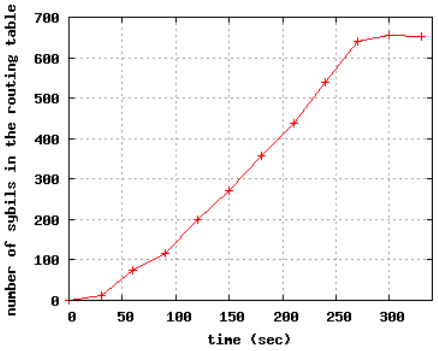

Fig. 5. Propagation of Sybils with spoofed IP addresses with flood & IP limitation protections enabled

Adding IP limitation: Our previous attacks were performed from a computer using a single IP address. With IP limitation enabled, we effectively measure that only the first Sybil enters the routing table, the other KADIDs presenting the same IP address can not pass. To bypass this limitation, we modified our attack using raw sockets so that each Sybil is announced with its own randomly spoofed IP address. The graph 5 shows that Sybils with spoofed IP addresses totally break this protection. The first time we launched the attack to evaluate the IP limitation, we disabled the flooding protection. When activating both of them in a second time, we noticed that the flooding protection is also completely bypassed by the IP spoofing. In fact, the packet tracking used to detect flooding measures the number of incoming requests from a given IP address. When spoofing IP addresses, each message appears to be from a different source so that the flooding is not detected. Even if IP spoofing can not be used to spy or eclipse content in the network, spoofed IP addresses were used in [6] with contact overwriting to partition the KAD network. This attack remains possible at this point, but the very last version of the clients implements a protection to mitigate both IP and KADID spoofing.

Adding identity verification: The attack launched to test the identity verification mechanisms is the same as the one described above, using Sybils with

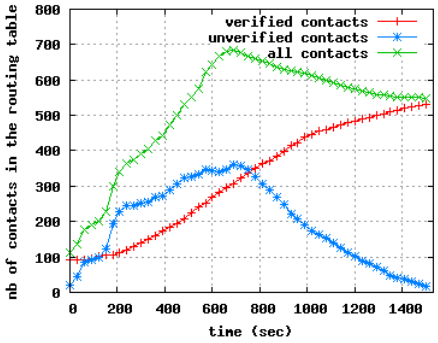

Fig. 6. Evolution of the contacts status in a normal run

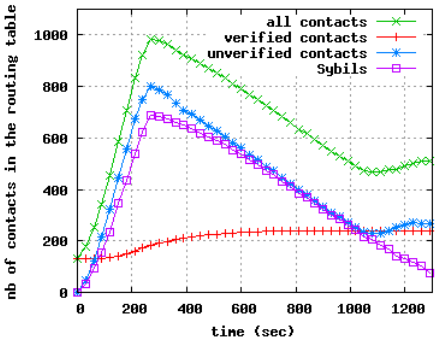

Fig. 7. Evolution of the contacts status under Sybil attack

spoofed IP addresses. Part of the behavior of this last mechanism seems strange and disappointing because the Sybils that fail the three-way handshake are still added (at least temporary) to the routing table and marked as "unverified". The rate of unverified contacts is high even without attack during the first 15 minutes of connection (graph 6). Even if unverified contacts are not used for routing tasks, they can take the place of potentially good contacts in a K-bucket, resulting in a reduction of the routing efficiency for the peer. The positive aspects are that all Sybils are marked as unverified (graph 7) and that unverified contacts are dropped faster from the routing table than with the classical 1-hour timeout. Moreover, unverified contacts can not update themselves before being checked, so that Sybils with spoofed IP addresses really are active for only a limited time. As expected, we also experimented that KADID overwriting is from now on impossible without the proper private key and mitigates the attack described in [6].

The results obtained with the very last version show significant improvements of the Sybil attack defense. As expected, IP address limitation coupled with identity verification avoids the Sybil attack from a single source when it previously would have been able to infect and spy or DoS the whole network.

3.3 Distributed eclipse attack

At this point, we have tested the resilience of the routing table to Sybils with the new protection mechanisms. But a massive Sybil attack is not the only and smartest way to damage the network. More localized Sybil attacks are possible and need fewer resources. The eclipse attack is another well-known issue of the KAD network as described in [9]. It consists in placing few malicious peers very close to the target ID in the DHT and making them known. Prior to the attack, the authors used a "crawler" in a tolerance zone to know many contacts and promote their Sybils. As all requests toward the target will pass through this zone where the peers are polluted, and considering the KAD search algorithm

"return the X closest nodes...", the requests will arrive at malicious nodes with a high probability.

More than the massive injection of Sybils, we think that the heart of the eclipse attack is the possibility to freely chose its KADID very close to a target which keeps being possible. As this issue is not yet handled in the last versions of KAD, the eclipse attack remains a threat even if the IP limitation now requires its distribution (involving several public IP addresses). We did a distributed eclipse attack with 24 nodes from PlanetLab and EmanicsLab[1]. Our client is based on aMule 2.2.2, compiled to run as a daemon and modified in several points to make the attack. Our client uses a forged KADID to be very close (96 bits) to the hash of a given keyword. For this specific ID, our client spies the incoming requests and eclipses the keyword by answering to Publication requests and denying Search requests. To improve its efficiency, our client announces itself actively by periodically sending Kademlia routing and Hello requests toward predefined KADIDs in order to be known in the network and attract requests, as an alternative to a centralized crawler.

We first launched our attack on the keyword "the", known to be the most indexed keyword in the network. Despite a massive indexation of publish requests by our Sybils (16000 requests per minute), we cannot totally eclipse the keyword. So, we analysed the results obtained by a search request on "the" and we found out that all the answers came from the same IP address managing 256 KADIDs sharing 120bits with the keyword, but not denying any Search request. In other words, "the" was already the target of another classical Sybil attack, always possible as a large part of the network is not yet updated with the new clients. In a second time, we launched our attack on the keyword "document", receiving "only" 115 publish requests per minute but resulting in its total eclipse. Even if the needed bandwidth depends on the size of the attack and the popularity of the targeted ID, it keeps very small (few KB/s by node).

In conclusion, distributed and localized attacks do not require substantial resources and keep being a real threat despite the new mechanisms as long as the KADID can be freely chosen and the malicious nodes placed very close to a given target. Controlling a small botnet, or more generally some IP addresses distributed over multiple subnets, allows a malicious user to control several entries of KAD.

4 Conclusion

Considering the previous attacks and our evaluation of an old client, KAD was totally unprotected and very easily and badly hurt by a Sybil attack from a single computer. The local solution chosen by the authors of eMule fits the constraints of the network: no infrastructure cost and backward compatibility. The network is progressively being composed of new clients including the protections and becomming more robust. We have shown that the new defensive rules really

[1] EmanicsLab is an autonomous testbed network based on PlanetLab architecture and sponsored by the European Network of Excellence EMANICS

make the Sybil attack harder to perform and successfuly mitigate the previously experimented attacks at a low cost. Where an attack could have been launched from a single computer few months ago [9], similar results can only be achieved now with a botnet or a more complex distributed architecture.

Even if these security mechanisms are a step forward, they are not yet sufficient. We have shown that a distributed eclipse attack focused on a particular ID still remains possible with a moderate cost. This result shows that the main weakness of KAD has shifted to the possibility to reference many KADIDs with the same IP address to the left possibility to freely choose its KADID. Moreover, if we consider an attacker with many resources, particularly considering the number of IP addresses, the overall protection can be threatened due to the specific design using local rules. As all protections are local to the client, a same pool of IP addresses can potentially infect all peers. Considering this, an attacker with several hundreds of spaced IP addresses can massively infect the network (for example, if controlling a medium-sized Botnet) and realize large attacks on the network scale. Thus, the IP limitation as currently defined, seems to be a very short-lived protection mechanism when considering IPv6. A single personal broadband connection could be assigned with a /64 or even more, which is sufficient to launch massive Sybil attacks with one different public IPv6 address per Sybil. The rules concerning the IP address limitation will have to be adapted carefully in order not to become inefficient.

However, even if not perfect, the defenses against the Sybil attack studied in this paper are absolutely generic and can be easily applied to every structured P2P network. We think that this kind of "common-sense" protection based on self-management is the minimum that every implementation of a structured P2P network should have, unless being totally unaware of the Sybil attack issues. In our future work, we will design and evaluate new security mechanisms to mitigate the distributed eclipse attack. It is interesting to consider the impact of this attack on the routing table of KAD and on the indexing and search processes. Being given the existing security rules of the indexing scheme, evaluating the remaining weaknesses will lead us to design new constraints improving the local protection.

Acknowledgment: This paper was partially supported by the French Ministry of Research project, MAPE, under contract ANR-07-TLCOM-24 and by the EC IST-EMANICS Network of Excellence (#26854).

References

1. Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation*, pages 299–314, New York, NY, USA, 2002. ACM.
2. Jochen Dinger and Hannes Hartenstein. Defending the sybil attack in p2p networks: Taxonomy, challenges, and a proposal for self-registration. In *ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security*, pages 756–763, Washington, DC, USA, 2006. IEEE Computer Society.

3. John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.
4. Francois Lesueur, Ludovic Mé, and Valérie Viet Triem Tong. A sybil-resistant admission control coupling SybilGuard with distributed certification. In *Proceedings of the 4th International Workshop on Collaborative Peer-to-Peer Systems (COPS)*, Rome, jun 2008.
5. Peter Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 53–65, London, UK, 2002. Springer-Verlag.
6. Eric Chan-Tin Tyson Malchow Denis Foo Kune Nicholas Hopper Yongdae Kim Peng Wang, James Tyra. Attacking the kad network. In *SecureComm 2008: the 4th International Conference on Security and Privacy in Communication Networks*, Istanbul, Turkey, 2008. ACM.
7. Hosam Rowaihy, William Enck, Patrick McDaniel, and Thomas La Porta. Limiting sybil attacks in structured p2p networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2596–2600, 2007.
8. Atul Singh, Miguel Castro, Peter Druschel, and Antony Rowstron. Defending against eclipse attacks on overlay networks. In *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 21, New York, NY, USA, 2004. ACM.
9. Moritz Steiner, Taoufik En Najjary, and Ernst W Biersack. Exploiting KAD: possible uses and misuses. *Computer communications review, Volume 37 N5, October 2007.*
10. Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. *SIGCOMM Comput.Com.Rev.*, 36(4):267–278, 2006.