Security in Distributed Hash Tables - an overview of research methods

Jussi-Pekka Erkkilä
Aalto University
Department of Computer Science and Engineering
Espoo, Finland
juerkkil@iki.fi

***Abstract*—Currently Distributed Hash Tables (DHTs) have a significant role in Internet and peer-to-peer protocols. However, DHTs also contain a number of security issues. This paper provides an overview of security research in context of DHTs from methodological point of view. We discuss about security issues and techniques in DHTs but the main focus is in research methods. Also, high-level model is presented to describe the research process in our context.**

***Keywords*-research methods; distributed hash tables; DHT; security; research process; Sybil attack; Eclipse attack; peer-to-peer**

I. Introduction

Distributed Hash Tables (DHTs) are scalable and efficient way to implement a decentralized lookup service in a distributed system. DHTs are widely used for example in peer-to-peer networks. Unfortunately, Distributed Hash Tables are vulnerable for several kinds of attacks. [1] In this paper we are focusing in security aspects of DHTs from methodological point of view.

During late years, many research papers have been published and several studies have been organized in order to learn more about the nature of the DHTs, their security aspects, and of course to find new solutions for known security vulnerabilities. As a good overview about the topic, Urdenate et. al (2011) [1] have published a comprehensive paper which summarizes the most critical security vulnerabilities and describes the techniques developed against those threats.

The purpose of this paper is not to analyze the security itself or provide new solutions. Instead, the goal is to give a clear overview about research methods in the context of security and Distributed Hash Tables. However, at first it might be useful to give a short explanation about security techniques and vulnerabilities in DHTs. To be able to understand the security we need to understand how DHT networks operate and what effects certain aspects of they may cause from security point of view.

As a distributed and highly scalable system, the Distributed Hash Tables are difficult target of practical study. It is extremely complicated to set up an authentic and realistic testing environment with millions of nodes all around the world. This is usually out of the question because lack of the resources as well. That is why computer simulations and mathematical modeling have a significant role in the research. Experimenting proof-of-concept implementations are usually done by simulating the physical network layer and running the real implementations on virtualized environment. However, these topics will be discussed more closely on chapter three and four.

This paper is divided in six sections: after this chapter a brief introduction to DHTs and their overall functionality is presented. In third section we focus on security aspects of DHTs presenting three main security issues and some approaches to address them. In fourth section relevant research methods in this context are covered. After that we present a typical research process in context of Security in Distributed Hash Tables. We will also focus on the techniques the research methods are be applied in this context. Last chapter concludes the whole paper.

II. Overview about the Distributed Hash Tables

By Distributed Hash Table we mean basically a normal hash table structure that is distributed to many network nodes. Every node contains a subset of key-value pairs so that together they form the whole data structure. Every node is connected to subset of other nodes. The size of subset is usually around *log N* where N is the total number of the nodes in the system. [2], [1]

As normal hash tables, DHTs also implement lookup service so that by searching the *key* we can obtain the corresponding *value* from the hash table. Lookup requests are forwarded to the responsible node by other nodes. Routing a lookup request should take at most $O(logN)$ hops [2]. Many papers exist that propose solutions for significantly better routing performance [3], [4]. However, these proposals often contain some drawbacks such as higher memory consumption or more complex network architecture.

Here we list the basic requirements and characteristics of DHTs.

A. Scalability

The fundamental requirement for DHTs is scalability. DHTs should be scalable up to millions of nodes and even more key-value pairs. In practice this means that the

search and storage complexity should not grow more than by magnitude of *O(log N)*. [2]

B. Decentralization

No central server exists, every node in the network is equally important. [5], [2]

C. Availability

All the data should be available from any node in the network despite that the nodes are rapidly joining the network and exiting from the network. This also requires some replication. [2]

D. Load balancing

Node identifiers and data items should be distributed in a way that every node would need to carry roughly equal amount of requests. [5], [2]

III. Security in Distributed Hash Tables

In this section we discuss about security issues in DHTs. We are not focusing on implementation-specific vulnerabilities but on high level security issues that exist because of fundamental characterstics and properties of DHTs. We also provide a brief overview about the possible ways to defence against these threats.

A. Security Issues

There are three main security issues related to DHTs mentioned in literature. These are called Sybil attacks, Eclipse attacks and Routing and Storage attacks. [1]

In Sybil attacks the idea is that an attacker generates large amount of nodes in the network in order to subvert the reputation system or mechanisms based on redundancy [1]. These nodes do not necessarily need to be real physical computers but they can be virtual nodes controlled by single attacker. The Sybil attack is not specific to DHTs but DHT is type of a system which is vulnerable to Sybil attacks. The vulnerability to Sybil attack depends on how cheap it to generate new nodes.

Eclipse attack is based on poisoning the routing tables of honest nodes. As there are many nodes joining and exiting from the DHT all the time, nodes need to actively update and synchronize their routing tables with their neighbors in order to keep lookup system functional. Thus, one malicious node can potentially poison many of its neighbors' routing table by providing false information [6]. If the attacker possesses a "narrow" point in a network, he can utilize the Eclipse attack to potentially isolate the network in two parts.

Routing table and Storage attack is a type of an attack where a single node is not following the protocol. Instead of forwarding the lookup requests, it may drop the messages or pretend being the responsible of the key. Hence, it may provide corrupted or malicious data - such as viruses or trojan horses - as a response. [1]

Sybil attacks or Eclipse attacks do not directly break the DHT nor damage the other peers. They are more like tools for attacker to control the routing and data flow in DHT. Instead, Routing table and Storage attacks are something that actively try to harm the network and the other peers. Thus, effective way to organize attack in DHT would be setting up a malicious node providing corrupted information and then utilizing Eclipse attack or Sybil attack to forward the requests to that node.

B. Security Mechanisms

Much research have been done in order to protect the peers and the whole network against existing security threats. We are not going in details here as the focus of this paper is not in the security itself but research methods. However, some practical examples of security solutions in DHTs are listed below.

1) Sybil attacks: As a defence against Sybil attack, there are several different approaches. Borisov (2006) [7] proposes a challenge-response protocol based on computational puzzles. The idea is that every node should periodically send computational puzzle to its neighbors. Solving the puzzle "proves" that the node is honest and trustworthy, but it also requires CPU cycles. The goal is to make organizing Sybil attack more difficult: running one peer client does not require much of CPU power, but running thousands of active virtual nodes is computationally infeasible.

2) Eclipse attacks: An obvious way to shield against Eclipse attacks is to add some redundancy in routing. This approach is utilized by Castro et. al (2002) who propose two separate routing table: the optimized routing table and the verified routing table. [8]

3) Routing and Storage attacks: As an example, Ganesh and Zhao (2005) [9] propose a solution where nodes sign "proof-of-life" certificates that are distributed to randomly chosen proof managers. The node which is making lookup request, can request the certificates from proof managers and that way detect the possibly malicious nodes.

IV. Research methods

In this chapter we discuss about research methods that have been applied on security research in context of DHTs. We focus on four most important methods which are computer simulations, data analysis, mathematical modeling and experimental research.

Practical approach on research process utilizing following methods, is provided in chapter five. In this chapter, some examples are provided about how these methods are applied in existing scientific studies.

A. Computer simulations

Computer simulation plays a big part in DHT research. From academic point of view, being aware of behaviour of routing and traffic flows and overall state in network is

very important. Also, organizing a large-scale experimental research is often out of question as running millions of nodes all around the world requires quite much of resources. Thus, the most practical way to get data about certain aspects of network, is simulation.

Usually the computer simulations is utilized to evaluate the severity level of known vulnerabilities or to evaluate the efficiency of proposed solutions.

B. Data analysis

Data analysis is also very important research method in our context. Simulations, experimental research and real life implementations provide us much data about the DHTs and their behaviour. However, we need to understand the meaning of that data and how certain variables and values affect to characteristics and "real life" behaviour of DHTs.

Along with simulation, data analysis is one of the most importants research methods in DHTs and their security. Basically in every survey and paper about the topic, at least some data have been analyzed at some level.

C. Mathematical modeling

Mathematical modeling has also a significant role in DHT research. Whereas simulation and data analysis often clarify us *what* are the issues and *why* they exist, we can apply mathematical modeling to provide solutions to these problems and to optimize our solution. Simulation and data analysis may provide our model the variables and relations. However, we can utilize modeling to optimize these relations and values and that way find better real-world solution as well.

As a reference, Naor and Moni (2003) utilize mathematical modeling to improve lookup performance and fault tolerance in DHTs. [10]

D. Experimental research

The role of experimental research in DHTs and their security aspects is not as significant as the methods listed above but it is still important. We need to have real life examples to support our simulation results. Also, experimenting proof-of-concept implementations must be done in real environment and real machines.

As mentioned, organizing a large-scale experimental study about DHTs is not very practical. Hence, one way to experiment DHTs is simulating the system partially. For example, the physical network can be simulated but application level implementation can be run in real environment. This approach was utilized for example by Condie et. al (2006) [11].

V. Typical research process

In this chapter we present a typical research process that is applied in research of DHTs. This is not a strict step-by-step procedure that should always be applied. Consider it more as a high level guideline that most surveys seem to roughly follow.

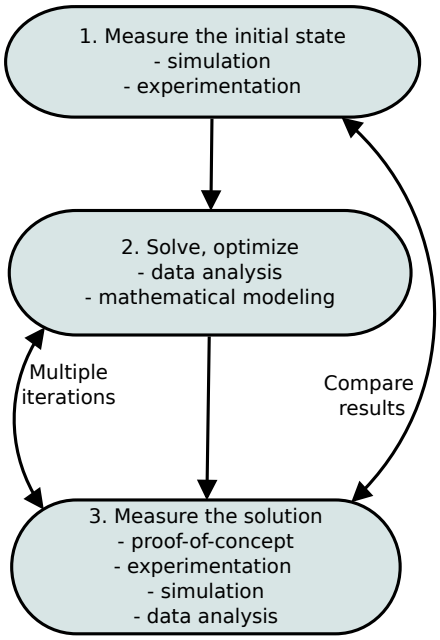

Figure 1. Typical approach to study DHTs and their security aspects

Mostly the purpose of research is not to find new security vulnerabilities. They are usually found by analyzing the system, protocol or implementation, and by experimentation. Instead, the goal in resesarch process is to analyze the severity of known vulnerabilities, examine under what circumstances they appear and how to shield against them.

In this example we divide the research process in three phases: analyzing the intial state, finding and optimizing the solution, and finally analyzing the suitability of the solution and drawing the conclusion. These phases are discussed more closely below.

A. First phase: Measuring the initial state

The research process always has some certain objective. In DHT and security research the objective is typically to address some existing security issue. First step is to study the security issue more closely. Consider for example Sybil attack. We know an existence of the security issue but we do not exactly know how big percentage of nodes should the attacker possess in order to subvert the network. Moreover, we do not know how easy it is to generate large amount of

virtual nodes.

Hence, we must experiment the situation either by simulating an attack or by implementing an attack in closed environment. In case of Sybil attack, we might also utilize mathematical modeling to approximate certain values. The first phase can also be considered as proving the severity of problem and providing the motivation for research.

B. Second phase: Finding and optimizing the solution

Once we have a clear picture of the problem and enough data to analyze the reasons for it, we can start finding the solution. Finding a solution that is applicable in practice is not always straightforward. In case of Sybil attack we could for example require a public key certificate signed by authority for every node [12]. However, in network of millions of nodes it is not always practical solution. [1]

There is no fixed algorithm nor a single research method that could always be applied to solve a security issue. However, in some cases we can for example build a mathematical model where the variables or characteristics that might cause the vulnerability, have been modified or removed. The same approach can also be used to improve the existing solution or performance.

Also, we need to pay attention to possible trade-offs that we need to accept in order to obtain the sufficient level of security. For example, adding security properties to protocol may increase the CPU cycles required by peers or overall traffic in the network.

C. Third phase: Measure and verify the solution

After we have built a model or found a solution for the security issue some other way, we need to proof the validity of our solution. Straightforward way to do this is to repeat the steps of first phase again - but now with different model and parameters. We run the simulations again, analyze the data and see how our security model affected the behaviour of the network and the protocol.

We can also build a proof-of-concept implementation, experiment the solution in real network or in closed environment and analyze the data again. Considering again the case of Sybil attack, the solution could for example make the network more tolerant against malicious nodes or make the generating of large amount of nodes infeasible.

The whole research process is iterative: if we are not satisfied with our solution, we can return to second phase and try to optimize our model and then experiment it again. Once the optimal solution have been found, it is important to compare the final outcome to the initial state: how significant were the findings, is there any drawbacks or trade-offs in the solution and were the initial goal achieved.

Whatever conclusion we draw, the important thing is that the study is repeatable, scientifically valid and provable.

VI. Conclusion

In this paper we studied security in Distributed Hash Tables from the methodological point of view. The goal of this paper was to present the most important research methods in context of Distributed Hash Tables and their security aspects. The another main objective was to give an overview how those methods are applied in practice.

As a conclusion, according to existing surveys and research papers about our topic and what we learnt in previous chapters, the two most important and popular methods were computer simulations and data analysis. Those are utilized almost in every paper in this research field. DHT is basically an overlay network so it is obvious that the network simulation tools are major asset for a researcher. Data analysis is strictly connected to the simulations: as we have much data as outcome of the simulations, we need to understand that data and realize how certain values affect to security properties.

Mathematical modeling is also in significant part in this topic: the security solutions are often based on mathematical conclusions and ideas and they can often be presented as mathematical models.

Experimental research is maybe not as big role as one might expect. Some examples exist where proof-of-concept implementations have been run in real environment. However, simulations are the primary tool for gathering data from network behaviour. The main reason for this may be that building a real testbed with millions of nodes all around the world is quite demanding and requires huge amount of resources as well. Also, simulations generally give us relatively good picture of network behaviour.

Still, security is quite unpredictable element in all networks. Experimenting the solutions and systems more actively in real or "semi-real" environment might reveal new characteristics and properties of DHT systems and their security.

The research process presented at chapter five is not the only way to do research in this field. Basically every research method can be applied in any part of the research - the approach presented in this paper is more like *usual way* to organize the research. The most important is to provide scientifically valid, repeatable findings and be able to present them in convincing form with appropriate data and background information. If no practically suitable solution for a security issue is found, that is an important outcome as well.

References

[1] G. Urdaneta, G. Pierre, and M. V. Steen, "A survey of dht security techniques," *ACM Comput. Surv.*, vol. 43, pp. 8:1–8:49, February 2011. [Online]. Available: http://doi.acm.org/10.1145/1883612.1883615

[2] K. Wehrle, S. Gotz, and S. Rieche, "7. distributed hash tables," in *Peer-to-Peer Systems and Applications*, ser. Lecture Notes in Computer Science, R. Steinmetz and K. Wehrle, Eds. Springer Berlin / Heidelberg, 2005, vol. 3485, pp. 79–93.

[3] H. Zhang, A. Goel, and R. Govindan, "Incrementally improving lookup latency in distributed hash table systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, pp. 114–125, June 2003. [Online]. Available: http://doi.acm.org/10.1145/885651.781042

[4] H. Sitepu, C. Machbub, A. Langi, and S. Supangkat, "Uno-hop: Efficient distributed hash table with o(1) lookup performance," in *Broadband Communications, Information Technology Biomedical Applications, 2008 Third International Conference on*, nov. 2008, pp. 76 –81.

[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.

[6] A. Singh, T. wan johnny Ngan, P. Druschel, and D. S. Wallach, "Eclipse attacks on overlay networks: Threats and defenses," in *In IEEE INFOCOM*, 2006.

[7] N. Borisov, "Computational puzzles as sybil defenses," in *Peer-to-Peer Computing, 2006. P2P 2006. Sixth IEEE International Conference on*, sept. 2006, pp. 171 –176.

[8] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 299–314, December 2002. [Online]. Available: http://doi.acm.org/10.1145/844128.844156

[9] L. Ganesh and B. Zhao, "Identity theft protection in structured overlays," in *Secure Network Protocols, 2005. (NPSec). 1st IEEE ICNP Workshop on*, nov. 2005, pp. 49 – 54.

[10] M. Naor and U. Wieder, "A simple fault tolerant distributed hash table," in *Peer-to-Peer Systems II*, ser. Lecture Notes in Computer Science, M. Kaashoek and I. Stoica, Eds. Springer Berlin / Heidelberg, 2003, vol. 2735, pp. 88–97.

[11] T. Condie, V. Kacholia, S. Sankararaman, J. M. Hellerstein, and P. Maniatis, "Induced churn as shelter from routing-table poisoning," in *In Proc. 13th Annual Network and Distributed System Security Symposium (NDSS*, 2006.

[12] J. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, ser. Lecture Notes in Computer Science, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Springer Berlin / Heidelberg, 2002, vol. 2429, pp. 251–260.