Towards Securing Communications in Infrastructure-poor Areas

Daouda Ahmat[1], Tegawendé F. Bissyandé[2,3] and Damien Magoni[1]

[1] LaBRI - CNRS, University of Bordeaux, France
{adaouda,magoni}@labri.fr
[2] SnT, University of Luxembourg, Luxembourg
tegawende.bissyande@uni.lu
[3] FasoLabs, Burkina Faso

Abstract. Structured P2P networks have proven to be effective in the exchange of data between nodes whose identity and content are generally indexed in a DHT. For years, such DHT networks have allowed, among other users, third world inhabitants, such as African people, to exchange information among them and with the rest of the world without relying on a centralized infrastructure. Unfortunately, more than ever, reliability of communication across the Internet is threatened by various attacks, including usurpation of identity, eavesdropping or traffic modification. Thus, in order to overcome these security issues and allow peers to securely exchange data, we propose a new key management scheme that enables to handle public keys in the absence of a central coordination which would be required in a traditional PKI.

Key words: P2P, Security, DHT, Key Management, Distributed Systems

1 Introduction

Opportunities of development are countless in a connected world. Unfortunately, participating to the information society remains a challenging endeavour for developing countries whose priorities lie elsewhere. Indeed, the lack of health care centers, schools, and practicable roads has pressing and damaging impact on the communities, outranking the need for investing in telecommunication infrastructures. In such contexts, opportunistic networking remains a prime choice to provide affordable means for sharing data, exchanging information and attempting to keep in touch with the rest of the world through internet [14].

Opportunistic networks are ideal for infrastructure-poor areas, where, contrary to traditional networking (1) all nodes of the network are not deployed together, (2) the size of the network cannot be approximately predicted, and (3) even the locations of the nodes are not pre-designed [10]. They constitute a special category of P2P networks where peers cannot rely on an 'observing' and 'mighty' central infrastructure whose existence is paramount. Instead, most implementations of P2P networks rely on a Distributed Hash Table (DHT) that provides a lookup service that each node in the network can use to map a given *key* with its corresponding *value*. Maintenance of the database of key/value mappings is obviously distributed among the nodes so as to avoid the use of a single point of failure that a central infrastructure would represent. Previous work has shown that it was possible to make DHTs scalable and reliable [16, 23]. Consequently, in the absence of a central infrastructure, structured P2P networks can still benefit from the properties of DHTs to allow scalable interaction among connected nodes. Nonetheless, in the context of such setups, security and privacy remains challenging to implement and maintain.

Without guarantee of provision of a central infrastructure, P2P networks are bound to face security and reliability issues that existing common policies and techniques fail to take into account in their implementations. For example, standard securization measures for communication involve the use of key-based encryption which is often implemented with public-key cryptography. However, a central problem with the use of such type of cryptography lies in confidence that a given public key is authentic, i.e., that it is correct and belongs to the entity claimed, that it has not been tampered with or that is has not been replaced by a malicious third party. Usually, this confidence is guaranteed by a central Public-Key Infrastructure (PKI), in which one or several certificate authorities certify ownership of key pairs. There is therefore a requirement for a strong centralization to manage cryptography keys, a luxury that P2P networks cannot practically afford [24].

We propose in this paper to take into account the specificities of P2P networks and the security and reliability requirements for exchanging information in this era, to design a new approach for the management of public keys. Contrary to the traditionnal PKI, the management of public keys in our scheme is *decentralized*. Indeed, this management is distribued among the peers that form the P2P system. In practice, we rely on the "hyperbolic plane"-based topology where each node can select his parent accordingly in a distributed process. Public keys can then be forwarded safely for use by peers to ensure communication security. In previous work [1], we have proposed an approach to address the security issues of the communication sessions when users are mobile across the network of nodes. This approach targetting user-level applications is complementary to the approach developed in this paper which targets the security of the underlying network nodes.

The main contributions of this paper are:

- We first discuss the opportunities of P2P networks for developing regions: where and how they can be harnessed to deliver connectivity in a truely beneficial way. We then discuss the challenges that arise in such networks, focusing on the safety and reliability of communications.
- We then discuss the challenges for implementing authentication of sources in a P2P network. We emphasize on why traditional PKI which are currently successful on the Internet appear to be inadequate for P2P systems.

– We detail our approach for distributing the management of public keys across the different nodes. The novelty of this approach is that it leverages existing principles in a state-of-the art topology, to provide a reliable and secure way to manage public keys.

The remainder of this paper is organized as follows. Section 2 discusses P2P networking in the context of African developing regions. Section 3 enumerates the challenges that must be overcome to secure communications in P2P networks. Section 4.2 presents our approach. Section 6 discusses related work and Section 7 concludes.

2 P2P and developing areas

The network environment in developing regions, in particular Africa, is challenged. Networks in such areas are often characterized by frequent, lengthy, and unpredictable link outages, as well as congested usage of an already limited bandwidth [4]. A recent comparison of bandwidth available and its costs in developed and developing regions showed that the discrepancy reaches an order of magnitude. Saif *et al.* have reported that while a 2 Mbps ADSL link in the United States costs around US$40/month, a 2 Mbps broadband connection in Pakistan costs close to US$400/month [18]. Mainly three reasons explain such differences: (1) the cost of incoming bandwidth on links between developed and developing countries; (2) the lack of performant and adequate ISPs at the right scale; and (3) inadequate provisionning for "pre-paid" users who account for most internet users but whose base is harder to anticipate.

To address these issues, researchers and practitionners have relied on alternative paradigms involving Delay Tolerant Networks [4] and Opportunistic Networks [10, 11, 14, 15] which appeared to suit the requirements of developing regions. In this context, Peer-to-peer systems have been proposed to support the enhancement of connectivity across developing areas. Researchers have thus proposed to apply P2P technologies to networking needs that are more urgent than simple (and often illegal ?) file sharing. For instance, P2P was found suitable for offline internet access [17, 19].

3 Challenges

Securing communication in P2P networks is a challenging endeavor, especially with regards to the standard practice of crypting information. In this section, we precisely detail some obvious and non-obvious challenges to highlight the constraints of finding a solution for P2P systems.

Key distribution. The first challenge that we encounter is the mode of distributing cryptographic keys in infrastructure-poor or infrastructure-less areas. Indeed, traditionally, a management-friendly central infrastructure, called PKI, is relied upon for this task. In absence of infrastructure we propose a fully distributed approach to spread keys. To this end, we benefit from the distributed features of existing “hyperbolic plane” overlay networks, in particular self-organization.

Use of alternate infrastructure. Relying on the hyperbolic tree to assure the forwarding of cryptographic keys also comes with problems that traditional PKI was able to easily handled. Indeed, there is a new need to ensure in a distributed system that the construction of the overlay network where each peer is properly identified will guarantee the robustness of the exchange scenario with little possibility of corruption by any intermediate peer.

Exchange of keys When initiating a communication in a P2P network, there will be a need for the peers to agree on the generation of a session key that the two peers will shared. A challenge in this requirement lies in the negotiation which should also be secured. Since usage of cryptography asymmetric algorithm to secure information is expensive, we propose to only rely on it in the key negotiation phase.

Detection of attacks The last but note least issue is to provide a mechanism for detecting corrupted keys. Indeed, when a corrupted cryptographic key is detected, it should be revoked by peers who are aware of the corruption. The information must also be broadcasted in the network through a notification message that should assure that all corrupted keys are flushed out of the memory of connected peers. Similarly, a challenging endeavor will be to prevent the Man-in-the-middle attacks during session key negotiation phase

4 System Design

In this section we discuss the different roadmaps that could be used to secure communications in P2P networks. We then detail our approach and the involved algorithms and heuristics.

4.1 Roadmap

In the absence of the traditional PKI for a central management of public keys, different schemes have been implemented to allow two peers to share a key (i.e., a secret) in a reliable way. We propose to go over those methods so as to highlight their limitations. We build our approach on top of these schemes, aiming at adressing with a DHT the different issues that we encounter in them:

- **Diffie-Hellman (DH) protocol** [5]
  The DH protocol is a strong cryptographic algorithm that enables sharing a secret between two nodes. Indeed, the proposed key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret. The strong point of the algorithm proposed by Diffie and

Hellman also lies in the fact that the key exchange can be performed over an insecure communication channel. This protocol has been demonstrated to be secure against eavesdropping, and an extended version of the DH protocol has been implemented by Steiner *et al.* to share a secret in a group of n participating nodes [21].
Unfortunately, the DH scheme has been demonstrated to be opened to Man-in-the-middle (MITM) attacks, a category of attacks that should be ruled out in P2P networks.

- **DH-based multipath key exchange method** [22]
  To address the MITM attacks with the DH protocol, Takano *et al.* have proposed a multipath key exchange scheme that enables to forward different components of an encryption key through separate paths, in order to prevent MITM attacks. Their approach is focused on P2P networks using a ring topology, such as Chord [13] and Symphony [12], and uses clockwise/anti-clockwise routing technique. Nevertheless, despite its interesting features, this approach still suffers a few issues:
  – it is restricted to a specific category of P2P networks (Chord and Symphony), and the ring topology does not appear to be realistic in the scenario of opportunistic networking where P2P connections are ad-hoc.
  – it makes various assumptions that cannot be guaranteed in a truely P2P opportunistic network.

4.2 An enhanced multipath DH-based key management

In this section, we propose an improvment (extension) of the model designed by Y. Takano et al. [22]. Scalable, decentralized and self-organized networks such as DHT-based P2P systems enable many users to join them. Thus, each node can be connected to a lot of other nodes; there can be then several paths between two endpoints of the network. For the above reasons, network topology can be transformed into a graph: each node represents an edge of the graph and each link indicates a connection between two nodes. Considering that P2P networks consist of a large number of peers that have multiple connections with several peers, network topology could be represented by a p-connected graph.

Hyperbolic Addressing Tree. Our approach leverages previous work [3] where the authors have demonstrated that hyperbolic geometry can be efficiently used for characterizing the overlay in a P2P network. Figure 1 thus illustrates the hyperbolic plane where the area is subdivided in disjoint zones allowing for each node to be positionned in a unique way while being able to be connected to a finite number of other nodes. This number corresponds to the degree q of the spanning tree that is thus built.

In the hyperbolic plane, a node of the tree can independantly compute the addresses corresponding to its children in the tree, and the degree of the tree determines how many addresses each peer in the network will be able to give. Thus by fixing the degree of the tree at the birth of the overlay, and by allowing each peer to connect to any peer at any time, the network can scale. In their

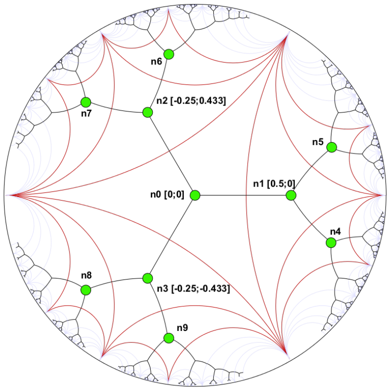

Fig. 1. A regular tree in the hyperbolic plane where each node has a degree 3 (Image credit: Cassagnes *et al.* [3])

Algorithm 1: Dispatching, at source peer, for subkeys accross separate paths

```
Input: sourcePeer, listKeyComponents
Output: success / failure
degree ← getPeerDegree(sourcePeer);
list ← listKeyComponents;
if $degree \leq 0$ || $list = \emptyset$ then
    return failure;
foreach $subKey \in list$ do
    path ← selectSeparatePath();
    routeSubKey(subKey, path);
return success;
```

work, Cassagnes *et al.* have proposed a distributed algorithm which ensures that the peers are contained in distinct spaces and have unique coordinates. As the global knowledge of the overlay is not necessary in their approach, "a new peer can obtain coordinates simply by asking an existing peer to be its parent and to give it an address for itself. If the asked peer has already given all its addresses, the new peer must ask an address to another existing peer. When a new peer obtains an address, it computes the addresses (i.e hyperbolic coordinates) of its future children. The addressing tree is thus incrementally built at the same time than the overlay" [3]. For more details on the creation of the addressing tree, we refer the user to the work of Cassagnes *et al.* [3].

We modify the integration of a new peer to the overlay to allow :

– a more careful selection of the parents so as to allow the existence of redundants paths that do not immediately interset at a parent node of a given source node.
– a flexible construction of a P2P network that takes into account the security priorities of each connecting peer.

In our approch explicited by Algorithm 2, since every peer can connect to any other peer, a new peer can be forced to pick parents with specific properties when security is an important requirement for him. In our scheme, each peer must have at least two parents when it is concerned with security issues. Thus, while the redundant paths will allow to protect communication against eavesdropping, a selective connection will mitigate MITM attacks. Indeed, during the connection of a peer, each contacted potential parent will reply with the address it offers to a child as well as the addresses of the peers to whom it is already connected. Thus, the new peer can realise if this potential parent is also linked to his already selected other parent. In such cases, if the new peer requires very secure communications that will be MITM attack-proof, it must contact a different parent.

Algorithm 2: Attempt a connection to a parent node p

```
Input: p, alreadyConnectedParentAddrs
       allAddrLinkedToParent
Output: newAddr
newAddr ← ⊥;
(proposedChildAddr,allAddrLinkedToParent) ←
requestAddrToParent(p);
foreach $addr$ ∈ alreadyConnectedParentAddrs do
    if $\exists p_a$ ∈ allAddrLinkedToParent | $p_a = addr$
    then
        return ⊥; /* try to connect to a
        different parent */

newAddr ← proposedChildAddr;
allAddrLinkedToParent ← allAddrLinkedToParent ∪ $\{p\}$;
return newAddr;
```

Algorithm 3: Paths selection

```
Input: keyToTransmit, nodeNeighbours
nodeNeighbours ← sortConnectedNodes(nodeNeighbours );
keyComponents ← splitKey(keyToTransmit );
$usedNode$ ← lastOf(nodeNeighbours );
while keyComponents ≠ ∅ do
    if $usedNode$ = lastOf(nodeNeighbours ) then
        $usedNode$ ← firstOf(nodeNeighbours )
    else
        $usedNode$ ← nextOf(nodeNeighbours )
    component ← firstOf(keyComponents );
    transmitViaSeparatePath(component, $usedNode$)
    ;
    keyComponents ← keyComponents \ {component};
```

Key Exchange Method

Although the DH crytographic algorithm [5] has been widely used to share secrets on unsecure communication channels, it is MITM attacks. In order to overcome this limitation of DH-based scenarios, we propose to use a multipath key exchange scheme.

Building upon a scalable P2P overlay [3] which uses virtual coordinates taken from the hyperbolic plane that is indifferent to underlying P2P network topology, our key exchange mechanism is based on multipath key negotiation as described in Algorithm 4. Takano *et al.* [22] have previously proposed a similar key negotiation mechanism. However, unlike our approach, their method is restricted only to Symphony and Chord P2P networks with a ring topology.

Algorithm 4: Key negotiation mechanism

$\boldsymbol{p}$: a prime number
$\boldsymbol{g}$: a generator

1. **Alice** selects a random number n, and computes $Key_a = g^n (mod\ p)$.
2. **Alice** selects q random numbers $ka_0, ..., ka_{q-1}$, such that $Key_a = \sum_{k=0}^{q-1} ka_k (mod\,p)$.
3. **Alice** routes all ka_k to **Bob** via q potential separate paths [1].
4. **Bob** receives q Key_a's components $ka_0, ..., ka_{q-1}$ sended by **Alice** and computes $Key_a = \sum_{k=0}^{q-1} ka_k (mod\ p)$.
5. **Bob** selects a random number m, and computes $Key_b = g^m (mod\ p)$.
6. **Bob** chooses q random numbers $kb_0, ..., kb_{q-1}$, such that $Key_b = \sum_{k=0}^{q-1} kb_k (mod\,p)$.
7. **Bob** sends all kb_k to **Alice** via q potential separate paths [1].
8. **Alice** receives q Key_b's components $kb_0, ..., kb_{q-1}$ from **Alice** and computes $Key_b = \sum_{k=0}^{q-1} kb_k (mod\ p)$.
9. **Alice** computes $Key = {Key_b}^n = g^{n \cdot m} (mod\ p)$.
10. **Bob** computes $Key = {Key_a}^m = g^{n \cdot m} (mod\ p)$.

Method for Dispatching Splitted Key Components

When a source node s is attempting to exchange a key with a destination node d, it splits this key into several subkey components that must be dispatched through separate channels. Algorithm 1 illustrates how the dispatching of key components is implemented. We propose a straightforward technique to overcome the challenge of selecting the different paths that are necessary to route separately the each subkey towards the destination node.

[1] see **algorithm 1** and next section for more details on the dispatching technique

As described previously, in the tree representing the nodes of our overlay network, the tree degree is set and known from start, and each connected node has a degree that may be less or equal to the tree degree. Thus, given n the tree degree, each node wishing to forward a key must split it into $n-1$ components that will be routed in separate paths. However, the selection of paths is iterative to account for the possibility that a few nodes may have less connections than the maximum possible. Thus, each node numbers the different nodes it is connected to and considers each of these nodes as the begining of a different path. Thus when a node has a degree k, it sends the first component to its first connected node (whether a parent or a child), and the second component to the second connected node, and so on until all connected nodes are used, and then it comes back to the first node to send the $(n-1-k)th$ component. We formally describe this path selection in Algorithm 3.

5 Security analysis

In a multipath key exchange scheme, a malicious node that wishes to compromise a key being exchanged must be able to collect each and all subkey components routed in the network. Formally, when paths $\mathcal{P}_0, ..., \mathcal{P}_{q-1}$ are used to send several distinct keys from source $\mathcal{S}$ to destination $\mathcal{D}$, the only malicious nodes that could compromise the key should be located at the intersection of all paths, i.e. $M = \mathcal{P}_0 \cap ... \cap \mathcal{P}_{q-1}$. $\mathcal{S}$ and $\mathcal{D}$ are obviously ignored in this set. Thus, when $\mathcal{P}_0 \cap ... \cap \mathcal{P}_{q-1} = \{\mathcal{S},\ \mathcal{D}\}$ (*bigon criterion* is respected [6, Lemma 2.5]) then all paths are disjoint and any MITM attack attempt cannot succeed. In such a desirable case, there exists a q-connected subgrah between $\mathcal{S}$ and $\mathcal{D}$ in the network topology. Consequently, the probability to have a MITM attack is estimated by $\sigma = \frac{|\mathcal{P}_0 \cap ... \cap \mathcal{P}_{q-1}|-2}{|\mathcal{P}_0 \cup ... \cup \mathcal{P}_{q-1}|-2}$ (where each path $\mathcal{P}_i$ is constituted of a set of consecutive hops from source to destination, and 2 represents the source and destination nodes, i.e $|\{\mathcal{S}\}| + |\{\mathcal{D}\}|$).

The number of distinct paths is dependant on the source node degree. Thus, for a given *q-regular tree*, if q is a large number, then there is a very high probalitiy to have several disjoint transmission channels. Nonetheless, despite the robustness of our multipath negocation approach, cooperative MITM attacks, where several nodes maliciously cooperate to compromise a key, are possible. However, it is very hard, and excessively costly to execute such an attack in a real environment, especially in infrastructure poor areas.

6 Related Work

Previous work have proposed to use DHT infrastructure to manage cryptographic keys with more or less success and mostly with many caveats [8, 25]. Wen *et al.* [26] have proposed a key management mechanism for DHT networks that transforms DHT table entries into a tree structure [7, 9]. Srivasta and Liu have relied on the Diffie-Hellman algorithm to deliver a solution that prevents threats in DHT networks [20]. Their scheme, however, remained sensitive to Man-in-the-middle attacks.

Wang *et al.* have built a distributed PKI on top of the Chord structured overlay network [2]. They have used threshold cryptography to distribute the fonctionnality of the PKI across the nodes of the DHT network. This Chord-PKI provides traditional PKI features such as certification, revocation, storage and retrieval.

Takano *et al.* have proposed a Multipath Key Exchange similar to that proposed in our work [22]. Their techniques however were designed to fit the Symphony and Chord P2P systems and their ring topologies.

7 Conclusion

Scalable P2P networks are self-organizing, autonomous systems that accept any peer without the need of resorting to a central coordination. Each peer alternatively plays the role of a router in order to relay traffic to other peers. The flexibility of such settings thus allows them to operate effectively in infrastructure poor-areas. Unfortunately, the features that make them desirable also make them vulnerable to eavesdropping and modification attacks.

In order to address the security challenges of P2P networks, we propose an improvement of P2P paradigms devised in previous work [3, 23] and a new approach for key exchange that generalizes a model proposed by Takano *et al.* [22]. On the one hand, we can still benefit from a truly scalable P2P overlay using hyperbolic coordinates where each node can decide on the strategy for accepting a parent node depending on its security priorities. On the other hand, we do not restrict our security scheme to a specific P2P topology, thus allowing our approach to be implemented in any P2P setup without any organization constraint.

In future work, we plan to implement a prototype of our system and assess it with simulation tools as well as with real-world experiments.

References

1. D. Ahmat and D. Magoni. Muses: Mobile user secured session. In *Wireless Days (WD), 2012 IFIP*, pages 1–6, 2012.
2. A. Avramidis, P. Kotzanikolaou, C. Douligeris, and M. Burmester. Chord-pki: A distributed trust infrastructure based on p2p networks. *Comput. Netw.*, 56(1):378–398, Jan. 2012.
3. C. Cassagnes, T. Tiendrebeogo, D. Bromberg, and D. Magoni. Overlay Addressing and Routing System Based on Hyperbolic Geometry. *ISCC'11 - IEEE International Symposium on Computers and Communications*, page 294–301, June 28-July 1 2011.
4. M. J. Demmer. *A Delay Tolerant Networking and System Architecture for Developing Regions.* PhD thesis, University of California, Berkeley, 2008.
5. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

6. D. B. A. Epstein. Curves on 2-manifolds and isotopies. In *Acta Math*, pages 15–16, 1966.
7. Y. Kim, A. Perrig, and G. Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups, 2000.
8. H. Kwon, S. Koh, J. Nah, and J. Jang. The secure routing mechanism for dht-based overlay network. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 2, pages 1300–1303, 2008.
9. P. P. C. Lee, J. C. S. Lui, S. Member, and D. K. Y. Yau. Distributed collaborative key agreement and authentication protocols for dynamic peer groups. *IEEE/ACM Transactions on Networking*, 14:263–276, 2006.
10. L. Lilien, Z. Kamal, and A. Gupta. Opportunistic networks: Challenges in specializing the p2p paradigm. In *Database and Expert Systems Applications, 2006. DEXA '06. 17th International Workshop on*, pages 722–726, 2006.
11. A. Lindgren and P. Hui. The quest for a killer app for opportunistic and delay tolerant networks (invited paper). In *CHANTS*, pages 59–66, 2009.
12. G. S. Manku, M. Bawa, P. Raghavan, and V. Inc. Symphony: Distributed hashing in a small world. In *In Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, pages 127–140, 2003.
13. R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM 2001*, San Diego, CA, September 2001.
14. J. Ouoba and T. F. Bissyandé. Leveraging the Cultural Model for Opportunistic Networking in sub-Saharan Africa. In *4th International IEEE EAI Conference on e-Infrastructure and e-Services for Developing Countries*, AFRICOMM, pages 1–10, Yaoundé, Cameroun, 2012.
15. J. Reich and A. Chaintreau. The age of impatience: optimal replication schemes for opportunistic networks. In *CoNEXT*, pages 85–96, 2009.
16. J. Risson and T. Moors. Survey of research towards robust peer-to-peer networks: search methods. *Comput. Netw.*, 50(17):3485–3521, Dec. 2006.
17. U. Saif, A. Chudhary, S. Butt, N. Butt, and G. Murtaza. Internet for the developing world: Offline internet access at modem-speed dialup connections. In *Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on*, pages 1–13, 2007.
18. U. Saif, A. L. Chudhary, S. Butt, N. F. Butt, and G. Murtaza. A peer-to-peer internet for the developing world. *Information Technologies & International Development*, 5(1), 2009.
19. U. Saif, A. L. Chudhary, S. Butt, N. F. Butt, P. Rodriguez, U. Saif, and A. L. Chudhary. Poor man's broadband: Peer-to-peer dialup networking. *ACM SIGCOMM Computer Comm. Rev*, 37, 2007.
20. M. Srivatsa and L. Liu. Vulnerabilities and security threats in structured overlay networks: a quantitative analysis. In *Computer Security Applications Conference, 2004. 20th Annual*, pages 252–261, 2004.
21. M. Steiner, G. Tsudik, and M. Waidner. Diffie-hellman key distribution extended to group communication. pages 31–37. ACM Press, 1996.
22. Y. Takano, N. Isozaki, and Y. Shinoda. Multipath key exchange on p2p networks. In *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, pages 8 pp.–, 2006.
23. T. Tiendrebeogo, D. Ahmat, and D. Magoni. Reliable and scalable distributed hash tables harnessing hyperbolic coordinates. In *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, pages 1–6, 2012.
24. G. Urdaneta, G. Pierre, and M. V. Steen. A survey of dht security techniques. *ACM Comput. Surv.*, 43(2):8:1–8:49, Feb. 2011.
25. P. Wang, I. Osipkov, and Y. Kim. Myrmic: Secure and robust dht routing. Technical report, 2007.
26. Z. Wen, N. Shao-zhang, and Z. Jian-cheng. A key management mechanism for dht networks. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, pages 339–342, 2012.