This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Implementing Secure P2P-ONS

Benjamin Fabian
Institute of Information Systems
Humboldt-Universität zu Berlin
Spandauer Str. 1, D-10178 Berlin, Germany
e-mail: bfabian@wiwi.hu-berlin.de

***Abstract*—Name Services for the Internet of Things (specifically, the EPCglobal Network) are distributed systems that serve the following fundamental lookup function: Given an identifier for a real-world object, e.g., an Electronic Product Code (EPC), they return a list of Internet addresses of services, which offer additional information about this object. Without name services acting as a broker between items and their information sources, the Internet of Things could not achieve the flexibility and global scalability necessary to live up to its vision. The currently specified Object Naming Service (ONS) for the EPCglobal Network has severe security drawbacks in its architecture and design.**

In this paper, we present the implementation of a Peer-to-Peer name service architecture based on Distributed Hash Tables (DHT) on the research platform PlanetLab. This alternative ONS architecture named OIDA, if deployed as an infrastructure network, offers enhanced overall multilateral security compared to ONS, combined with potentially better functionality, scalability, and roughly equivalent performance.

I. Introduction

Within supply chain communities, the – from a networker's perspective somewhat imprecise – term *Internet of Things* (IOT) is established to describe an emerging global, Internet-based information service architecture for RFID-tagged items (Radio-Frequency Identification). The most influential architecture and potential future nucleus for the IOT is the EPCglobal Network [1]. The IOT is anticipated to facilitate information exchange about goods in global supply chain networks, increase transparency, and enhance their efficiency. In an extension of this initial application scope, the IOT could also serve as backbone for *Ubiquitous Computing*, enabling smart environments to easily recognize and identify objects, and retrieve information from the Internet to facilitate their adaptive functionality; for example, in smart homes where items or furniture of daily use could be upgraded to provide information and counseling services.

Name Services for the IOT (IOTNS) are distributed systems that serve the following fundamental lookup function: Given an identifier for a real-world object, e.g., an Electronic Product Code (EPC), they return a list of Internet addresses of services, which offer additional information about this object. Without name services acting as a broker between items and their information sources, the IOT could not achieve the flexibility and global scalability necessary to live up to its vision. The currently specified Object Naming Service (ONS) [2] for the EPCglobal Network has severe security drawbacks in its architecture and design [3], and involves problems of international control over a future critical business infrastructure [4].

In this paper, we present an alternative IOTNS architecture based on Distributed Hash Tables (DHT) and its implementation on the research platform PlanetLab. This extends the discussion of the conceptual architecture presented in [5] by empirical evidence gathered from an implementation on an international research testbed. This alternative ONS architecture named OIDA, if deployed as an infrastructure network, offers enhanced overall multilateral security compared to ONS, combined with potentially equivalent or even better functionality, scalability, and performance.

II. OIDA Revisited and Extended

In our earlier paper [5], the outline of a DHT-based IOTNS architecture called *Object Information Distribution Architecture* (OIDA) was presented. Extending the functionality of ONS, OIDA can take arbitrary object identifiers as input and resolve them to lists of corresponding information sources, e.g. EPC Information Services (EPCIS).

OIDA involves the following key ideas: Each interested company deploys dedicated OIDA nodes. Those nodes form an overlay network using an ID space specific to the DHT in use, where a cryptographic hash function (CHF) maps object identifiers and nodes to overlay IDs. This pseudo-random mapping of identifiers to storage nodes balances load more evenly, allows for easy replication, avoids single points of failure and control, and reduces the feasibility of targeted attacks against specific information providers or clients. The DHT provides the routing to the responsible nodes, as well as joining, leaving, repair, and optimization procedures, without a central entity managing those operations.

Nodes store deterministically assigned – but from the perspective of node owners or adversaries, who are interested in specific EPCs, apparently random – encrypted and signed documents belonging to hash value ranges. Those documents may contain object data or pure EPCIS IP addresses, because if possible indirect use of DNS should also be avoided for privacy reasons. For scalable data authenticity, the existence of a certification authority (CA) infrastructure or a web of trust is assumed. The deployment of an X.509-based public-key infrastructure (RFCs 3280, 5280) is anticipated for the EPCglobal Network [6].

An information provider P who likes to publish EPCIS addresses or other information for a given object identifier (e.g., EPC) e prepares a document, d, cf. Fig. 1. d contains the publisher's name P_ID, public key P_{pub} and the EPCIS

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

Publisher ID	Publisher Public Key	Publisher Certificate	h(EPC)	EPCIS IP Addresses	(Optional) Item Data	Meta Data

Fig. 1. OIDA Example Document Structure

address information i. For detecting authenticated, but wrongly assigned messages the cryptographic hash of e is added. Additionally, version control information, time stamps, and TTL values should be included as Meta Data. If a central CA is used for OIDA, a certificate linking P_ID with its public key P_{pub} can be included. P signs a CHF value – for storage and performance efficiency – of this document (or parts of it) by using his private key P_{priv}, and adds this as a signature. For access control and to reduce the risk of inference attacks from the data included in the returned document, this data should be encrypted, for example by using a shared key k and a symmetric cipher like AES. Let s be an optional shared salt between provider and client to counter dictionary attacks (cf. the discussion in [5]); d is then stored r_{max} times in the DHT at the nodes responsible for overlay IDs $h(s, e, r)$, $1 \leq r \leq r_{max}$, by contacting a DHT node acting as a client gateway, for example situated in the information provider's company itself.

The storage step could include identification and authentication of P by the responsible nodes to avoid spam, mutual authentication for enhanced security, and further replication mechanisms to increase availability. Authorized clients will then retrieve d as is depicted in Fig. 2.

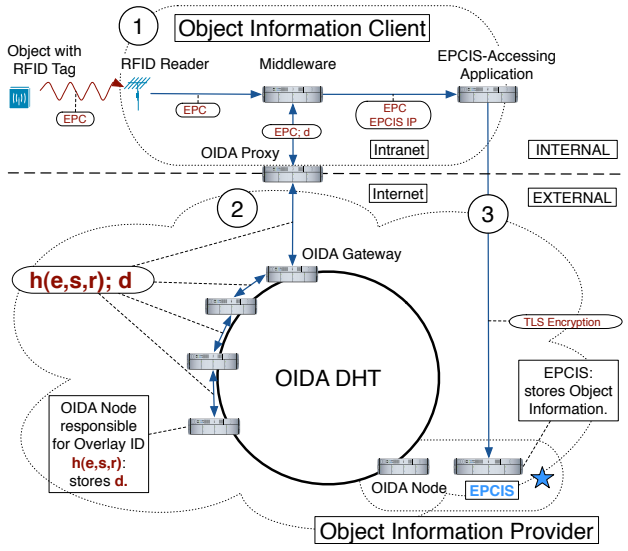

Fig. 2. Object-Information Distribution Architecture (OIDA)

OIDA decouples IOT name service tasks from the classical DNS infrastructure, which prevents an overburdening of the DNS with new applications depending on the IOT. Using a DHT for ONS will fulfill many of the IOT name service requirements like those we gathered in [5], and even many Discovery Services requirements stated in [7]. OIDA inherits the advantages of the underlying DHT architecture, which includes scalability, load distribution, redundancy, self-organization, and automatic repair mechanisms if nodes fail.

OIDA Security Measure	Positive Effect on Security Requirement
DHT Architecture	Availability, (Political) Multipolarity
Easy Document Replication	Availability, (Political) Multipolarity
Digital Signatures	Data Integrity, Authenticity
CHF, Salts	Query Confidentiality
Document Encryption	Document and Query Confidentiality
Recursive Routing	Anonymity
Node PKI (optional)	DHT and Routing Integrity, Availability

TABLE I
MAIN OIDA SECURITY MEASURES

OIDA offers enhanced availability and authenticity compared to ONS, and better confidentiality and anonymity under the assumption that recursive routing is used, salts are available to better protect the pre-image of the CHF, as well as keys to encrypt the documents – a necessary condition for access control to the published data as well as well. OIDA makes it significantly more difficult for adversaries and most function providers in the IOT – beyond local network boundaries – to collect (IP, EPC) pairs, or to track clients. This enhances client-side confidentiality, and overall multilateral security compared to ONS, which in its currently specified architecture involves high risks with respect to the requirements of Table I [3].

III. OIDA PROTOTYPE

In the following, we present an OIDA prototype implemented on PlanetLab[1] (PL), an international research network for the development of new network services. For PL design principles confer to [8], and for experimental system research on PL in general see [9].

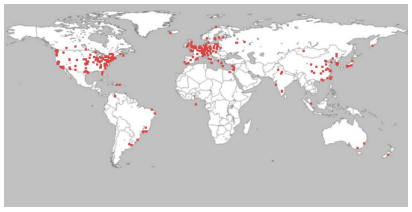

Fig. 3. Geographical Distribution of PlanetLab Nodes (Source: PL Web Site)

Virtual hosts on PL nodes can be reserved for projects and assigned to host groups (*slices*) under exclusive control of one experimenter, but the actual physical nodes must be shared with dozens of concurrent experiments at a given time. Therefore, PL offers a real world testbed under load, but cannot guarantee that experiments are exactly reproducible at a later time. In our experience, however, the general quality level of the results remained quite stable. PlanetLab consisted of around 850 nodes at 428 sites in April 2008 (Fig. 3). Out of those, our experiments used roughly 350 nodes, mostly stable nodes with long uptime, and a better network connection to our testing clients to avoid timeouts.

[1]URL: https://www.planet-lab.org/ (09-2008).

Our OIDA prototype (Fig. 4) is based on the Bamboo DHT [10], mainly because of its relatively mature status, and due to its design goal of withstanding *churn*, that is, frequent change in membership due to ongoing node departures, failures, and arrivals, a property we deemed important for a prototype using a globally distributed experimental platform. For a production version of OIDA, deployed as an infrastructure network, business contracts should guarantee a more stable node membership, but the ability to handle churn would be a plus for service robustness.

Bamboo has evolved from the Pastry DHT [11] and inherits its overlay geometry (a circle) and routing mechanisms. However, a larger identifier space $[0, ..., 2^{160})$ of cardinality 2^{160} is used. This corresponds to the possible output space of the SHA-1 CHF that is used in Bamboo for creating an overlay ID from a pre-image, which is a node's (IP address, port) tuple, or a data identifier.

Routing in Pastry and Bamboo uses two main sets of state information that have to be maintained by each node: The first is the *leaf set* L for connections to the k preceding and k subsequent nodes in the ID circle. This set is comparable in function to the set of *successors* in Chord [12], in general DHT studies the term set of *sequential neighbors* is used [13]. The second set is the *routing table* for larger hops through the ID space – similar to the *finger table* in Chord. The routing table of a node A contains nodes whose overlay IDs share successively longer prefixes with the overlay ID of A, where each ID is regarded as sequence of digits with base 2^b, and b is a fixed parameter of the deployed DHT. The routing table consists of $\frac{160}{b}$ rows and $2^b - 1$ columns, but is in general not completely filled. If available, an entry $R(i, j)$ of row i and column j should contain the IP address of a node whose identifier matches that of A in exactly i digits and whose $(i + 1)$th digit is j. It is possible that no such node is known, leaving the entry empty, or that multiple candidates exist, in which case one of them is chosen according to a specific metric, for example in Bamboo proximity in the network topology. On average, for a network of N nodes, only $\log_{2^b} N$ routing table rows are populated.

In addition, we used client scripts to encrypt, sign, store, retrieve, and verify data from several machines outside of PL, however, without implementing a truly global CA issuing certificates. Direct signature verification was used, trusting in the correctness of a publishers public key, because the deployment of a real CA and trust hierarchy was considered to be part of established network engineering, outside of the scope of the prototype. Bamboo was deployed in a dedicated PL slice on more than 350 nodes, distributed over all continents.

Management clients included tools from the CoDeeN content distribution network project like *codeploy* for deployment of new builds, and *multiquery* for parallel execution of startup and stop commands triggering local scripts on the PL nodes [14]. The operating systems used for the prototype included Fedora 4 on the PL nodes running Bamboo, Fedora 8, and MacOS X 10.5 for the rest of the infrastructure. The client scripts were programmed in Python, adapting and extending the short Python clients for Open DHT [15].

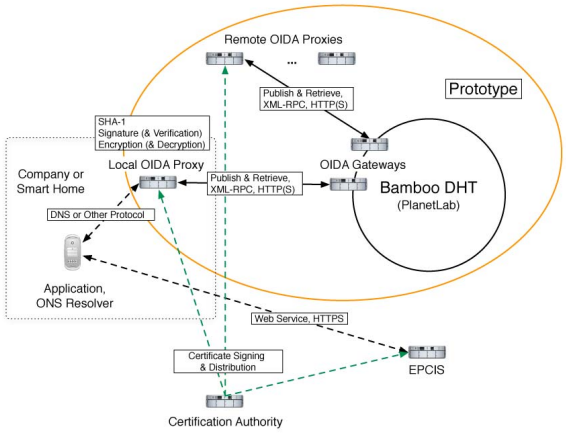

Fig. 4. OIDA Prototype on PlanetLab

IV. Testing

In the following, a set of experiments using the prototype is described, which have been conducted to confirm that OIDA is able to fulfill the functional requirements for an IOT name service, as ONS based on DNS does. In addition, some initial results on its performance are presented.

A. Document Preparation

The first step in testing the prototype involved the creation of individual address documents, for simplicity only containing fictional NAPTR records [2] corresponding to a chosen EPC set, and no additional data fields. The documents are encrypted by AES [16] and an RSA signature [17] is added. RSA was chosen in the prototype for obtaining rough lower bounds for the signature speed; in practice, its secure application and implementation must be verified, cf. [18, pp. 559]. Most cryptographic operations were implemented using the Python Cryptography Toolkit. Note that for the prototype we chose to first encrypt before signing, to be able to detect possible corruption during network transmission more easily. The results are stored locally in a Berkeley DB database instance.

One major test run for the record creation script generated 100,000 documents, using AES-128 encryption and RSA-2048 signatures on a desktop PC.[2] The test aimed to show rough lower bounds for the speed of encryption and signing. A production implementation could use more efficient Elliptic Curve Cryptography, see [19]. This experiment took approximately 37.33 minutes, with an average speed of 44.68 records per second, which was confirmed in magnitude by repeated test runs. The size of the database file was 132 MB. We conclude that even for massive amounts of data records to be stored in OIDA, the local preparation process, including encryption and signing of data, is feasible and very fast.

[2]Pentium 4, 2.80GHz, 1 GB RAM, Fedora 8, Python 2.5 (r25:51908 GCC 4.1.2), python-crypto-2.0.1-7.1.fc7.

B. Document Storage

This experiment simulated the publishing of EPCIS address data by an information provider. The data had been prepared in advance. The documents were published to the DHT from a client situated in the same university LAN as the OIDA gateway. We did not use additional salts s as input for the CHF during this test, because its impact on the performance is negligible. For this test it was assumed that the provider uses an OIDA proxy in his own organization, which in turn contacts an OIDA node via XML-RPC to store data in the DHT. We assumed this node also to be situated somewhere near, for example in a demilitarized zone (DMZ) of the local organization, similar to externally reachable company DNS servers. This was modeled by choosing a local PL node running OIDA as a storage gateway.

During the experiment, the Bamboo DHT suffered from moderate churn and network timeouts common to PL, around 2% of its nodes became unavailable – some of which reappeared later, however. The script used a timeout of 30 seconds, storing attempts taking longer than this – for example, due to network latency, load of the gateway, or storage node – were considered a failure. The number of EPCs and therefore individual documents was 2,000, each of which was stored in five independent copies ($r_{max} = 5$). The average storage time per copy was 580 ms, including failed attempts and some longer durations, which raised the average in comparison to the median time of 290 ms. Of the total of 10,000 storage attempts, about 99% were successful; for each EPC, at least three documents were stored successfully. This means, even in face of loss, a client application could still resolve 100% of the EPCs to corresponding EPCIS addresses, which was confirmed by the retrieval experiments below. In a real application, the detailed list of failed attempts could be used for selected storage retries of more copies at later times. Not surprisingly, document storage to the DHT was around 25 times slower on average than the document generation, but less than 13 times slower for half of the documents, which is still relatively fast. In conclusion, at least within the experimental settings and under moderate churn, storage to OIDA is practically feasible.

Finally, we measured the retrieval times for two different clients, representing a corporate OIDA proxy and a smart home application, respectively.

C. Document Retrieval from a Corporate Network

The retrieval process is the OIDA analogon to an ONS lookup. The final set of experiments measured the time to retrieve the documents stored during the tests described in the previous sections, in parallel from three different OIDA gateways around the world: Berlin (same LAN as the client), Helsinki, New York (Table II). The last two – arbitrarily selected from nodes with different RTTs – gateway sites served to test the feasibility of choosing remote OIDA gateways for failover, round robin, or increased confidentiality with respect to specific gateways. It must be noted however, that the XML-RPC connections were not secured by TLS during the test. Therefore, the impact of TLS connections from clients to the OIDA gateways on the performance was not measured – however, we anticipate this overhead not to be too critical in practice because it is possible to multiplex several TCP connections over the same TLS channel over a longer time, and the TLS delay is mostly dependent on this single session establishment, ideally performed once for all documents to be retrieved. Again the timeout for the experiments was 30 seconds, which is reflected by the maximum and average duration of all retrieval attempts, not only successful ones. The choice of a specific timeout value is up to the client application within the limits provided by the DHT.

OIDA Gateway	Berlin	Helsinki	New York
IP Address	141.20.103.211	193.167.187.187	216.165.109.81
RTT avg. (ms)	0.32	51.40	112.24
Success EPC	100%	100%	100%
Success Replica	99.78%	99.84%	99.68%
Total Duration (s)	4924.94	7068.37	9130.31
Median (s)	0.2136	0.3870	0.5253
Average (s)	0.4925	0.7068	0.9130
Minimum (s)	0.0063	0.1536	0.3347
Maximum (s)	30.0026	30.0579	30.1139
STD (s)	1.6716	1.7883	1.9639

TABLE II
OIDA DOCUMENT RETRIEVAL – COMPANY

OIDA Gateway	Berlin	Helsinki	New York
IP Address	141.20.103.211	193.167.187.187	216.165.109.81
RTT avg. (ms)	11.925	42.054	103.320
Success EPC	100%	100%	100%
Success Replica	100%	99.95%	99.69%
Total Duration (s)	5139.95	6383.69	9572.74
Median (s)	0.2504	0.3621	0.5140
Average (s)	0.5140	0.6384	0.9573
Minimum (s)	0.0468	0.1328	0.3231
Maximum (s)	30.0150	30.0464	31.0519
STD (s)	1.3542	1.3191	1.9634

TABLE III
OIDA DOCUMENT RETRIEVAL – SMART HOME

D. Document Retrieval from a Smart Home

While in the previous test the client was situated in a very fast university network representing a corporate client, we also tested the retrieval of documents from a client connected via a home DSL connection in Germany, suffering from approximately 1% packet loss on the average during pings to the gateways. This experiment was conducted to model a possible UC application retrieving address data for EPCs, for example as would be gathered by a periodic inventory process by smart shelves. The tests were conducted on another day, using a different EPC set of the same size, same document size, and roughly equivalent size of the DHT (330 nodes). In spite of these differences and the fluent state of PL, the results shown in Table III are surprisingly consistent with the previous retrieval experiment. Connection to remote gateways took longer and had higher miss rates due to timeouts, but were able to retrieve all documents because replication was provided. During these particular experiments and similar test runs, very rarely less than four copies of each document were successfully retrieved, and never less than three.

V. Discussion and Related Work

Our real-world experiments, though still limited in deployment scale, combined with the theoretical results from DHT research on scalability, give good reason to pursue further research and development on DHT-based name services for the IOT. OIDA, for example, if supported by an appropriate membership and authorization procedure and trust structure, fulfills all the functional requirements, and offers – as we experienced during PL deployment – appropriate robustness in face of random errors, assuming a good replication of the data, which is very easily achieved with DHTs. DNS may have some advantages over DHT-based systems by achieving ultra-low latency often well below one second in global deployment, mainly due to its caching mechanisms. But it is currently not clear if IOT name services need to provide the same very low latency as DNS offers e.g. for Web surfing to popular sites, because many IOT application scenarios may not involve direct human interaction. Even then, proactive caching mechanisms like BeeHive [20] could be adopted once the future query distributions for IOT identifiers, which perhaps – unlike DNS names [21] – may not follow a Zipf distribution, have been studied in more detail.

In addition to related work already identified in [5], there is currently work in the EU Bridge project on Discovery Services [7], which involve different functional requirements than basic IOT name services like ONS and OIDA, for example standing queries and the support for complex semantics, e.g. using business context. MONS [4] is a modification of the ONS Root architecture to provide distributed international control over the ONS, combined with an ONSSEC architecture for data authentication, but without enhancing anonymity or confidentiality of ONS queries and content. In the area of key distribution for RFID and the IOT, which could also applied to OIDA, relevant recent advances include [22], but further research should be conducted in this area, especially for IOT applications in personal environments.

VI. Conclusion

OIDA decouples IOT name service tasks from the classical DNS infrastructure. As our prototype and experiments showed, OIDA is able to fulfill all basic functional IOT name service requirements, besides many of the non-functional, especially security requirements stated in [5] and [7]. OIDA inherits the advantages of the underlying DHT architecture, which include scalability, load distribution, and absence of special nodes. Future work should focus on the quantification of IOT diffusion and its scalability and performance demands, but also on further security requirements elicitation of its stakeholders, and on methods for secure and scalable cryptographic-key distribution among them. In face of growing global Internet surveillance, stronger DHT anonymity enhancements may become necessary. Emerging designs for IOT Discovery Services could take possible modifications of our DHT-based architecture into consideration.

References

[1] EPCglobal, "The EPCglobal Architecture Framework – Version 1.2," September 2007. [Online]. Available: http://www.epcglobalinc.org/standards/

[2] M. Mealling, "EPCglobal Object Naming Service (ONS) 1.0," EPCglobal, 2005. [Online]. Available: http://www.epcglobalinc.org/standards/

[3] B. Fabian, O. Günther, and S. Spiekermann, "Security Analysis of the Object Name Service," in *Proc. 1st IEEE Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2005), in conj. with IEEE ICPS 2005, Santorini, Greece*, 2005, pp. 71–76.

[4] S. Evdokimov, B. Fabian, and O. Günther, "Multipolarity for the Object Naming Service," in *Proc. Internet of Things (IOT 2008), Zurich, Switzerland, 2008*, ser. LNCS 4952. Springer-Verlag, Berlin-Heidelberg, 2008, pp. 1–18.

[5] B. Fabian and O. Günther, "Distributed ONS and Its Impact on Privacy," in *Proc. IEEE International Conference on Communications (IEEE ICC 2007), Glasgow*, 2007.

[6] EPCglobal, "EPCglobal Certificate Profile – Version 1.0.1," May 2008. [Online]. Available: http://www.epcglobalinc.org/standards/

[7] BRIDGE, "BRIDGE WP02 – Requirements Document of Serial Level Lookup Service for Various Industries," August 2007. [Online]. Available: http://www.bridge-project.eu/

[8] L. Peterson and T. Roscoe, "The Design Principles of PlanetLab," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 11–16, 2006.

[9] L. Peterson and V. S. Pai, "Experience-Driven Experimental Systems Research," *Communications of the ACM*, vol. 50, no. 11, pp. 38–44, November 2007.

[10] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *Proc. USENIX Annual Technical Conference (ATEC '04), Boston, MA, USA*. USENIX Association, Berkeley, 2004, pp. 127–140.

[11] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware '01), Heidelberg, Germany*, ser. LNCS 2218, R. Guerraoui, Ed. Springer-Verlag, 2001, pp. 329–350.

[12] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[13] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The Impact of DHT Routing Geometry on Resilience and Proximity," in *Proc. ACM SIGCOMM 2003, Karsruhe, Germany*. ACM Press, New York, 2003, pp. 381–394.

[14] L. Wang, K. Park, R. Pang, V. S. Pai, and L. Peterson, "Reliability and Security in the CoDeeN Content Distribution Network," in *Proc. USENIX 2004 Annual Technical Conference, Boston, USA*, 2004.

[15] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "OpenDHT: A Public DHT Service and Its Uses," in *Proc. ACM SIGCOMM '05, Philadelphia, Pennsylvania, USA*. ACM Press, New York, 2005, pp. 73–84.

[16] NIST, "Advanced Encryption Standard (AES)," National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 197, November 2001.

[17] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[18] W. Mao, *Modern Cryptography – Theory & Practice*. Prentice Hall / Pearson Education, Upper Saddle River, 2004.

[19] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer, New York, 2004.

[20] V. Ramasubramanian and E. G. Sirer, "Beehive: O(1) Lookup Performance for Power-Law Query Distributions in Peer-to-Peer Overlays," in *Proc. 1st Symposium on Networked Systems Design and Implementation (NSDI'04), San Francisco, USA*. USENIX Association, 2004.

[21] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS Performance and the Effectiveness of Caching," in *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement (IMW '01), San Francisco, California, USA*. ACM Press, New York, NY, USA, 2001, pp. 153–167.

[22] M. Langheinrich and R. Marti, "Practical Minimalist Cryptography for RFID Privacy," *IEEE Systems Journal, Special Issue on RFID Technology*, vol. 1, no. 2, pp. 115–128, Dec. 2007.