This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Distributed ONS and its Impact on Privacy

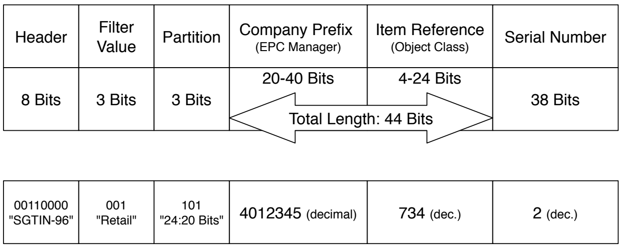
Benjamin Fabian and Oliver Günther
Institute of Information Systems
Humboldt-University Berlin
Spandauer Str. 1, D-10178 Berlin, Germany
e-mail: {bfabian, guenther}@wiwi.hu-berlin.de

Abstract— **The EPC Network is an industry proposal to build a global information architecture for objects carrying RFID tags with Electronic Product Codes (EPC). A so-called Object Naming Service (ONS) is used to locate information sources for these objects in the EPC Network. But ONS is based on DNS, which suffers from well-studied weaknesses in robustness, configuration complexity and security. There are promising approaches to enhance the performance and robustness of DNS by using structured P2P systems based on Distributed Hash Tables (DHT) that have a high potential as a replacement for ONS as well. We investigate if and how a decentralized alternative to ONS based on DHT could additionally offer data access control and enhance the privacy of its clients. As it turns out, the strength of privacy protection will slightly increase by using DHT compared to DNS, but strong protection will depend on the feasibility of secure out-of-band key distribution mechanisms.**

I. The EPC Network

Radio Frequency Identification (RFID) is about to be deployed in supply chains worldwide within the next years. The main incentive for RFID deployment is to improve efficiency and transparency by enabling real-time analysis and control of flows of goods. On the other hand, RFID is a core technology for Ubiquitous Computing and Ambient Intelligence, where smart, context-aware environments identify all objects they contain and adapt to explicit or anticipated user needs.

Besides the future ubiquity of RFID tags and readers there is another important factor that facilitates these applications: The standardization of a global numbering scheme for physical objects, the Electronic Product Code (EPC).



Header	Filter Value	Partition	Company Prefix (EPC Manager)	Item Reference (Object Class)	Serial Number
8 Bits	3 Bits	3 Bits	20-40 Bits	4-24 Bits	38 Bits
			Total Length: 44 Bits		

00110000 "SGTIN-96"	001 "Retail"	101 "24:20 Bits"	4012345 (decimal)	734 (dec.)	2 (dec.)

Fig. 1. An Electronic Product Code (EPC)

The EPC standard actually comprises a whole family of data structures. In its Serialized GTIN-96 variant (see Fig. 1), the EPC includes a header to denote its class (SGTIN-96), a filter value for fast logistic decisions, a partition value that indicates the boundary of the next two fields, and a company prefix that is a unique identifier of the item manufacturer. Finally, the manufacturer can assign item reference numbers to classes of objects she produces. Within the same class similar objects can be distinguished by their serial number – this is a fundamental extension of the conventional barcode.

The intention of some companies is to add RFID tags to as many objects that leave production as possible, and even to permanently integrate tags into clothes and devices. These tags store EPCs, which are unique keys to retrieve additional information from a large distributed network of databases around the globe, the EPC Network. Standardization and (to some extent) administration of this EPC Network is the task of the consortium EPCglobal Inc. (`www.epcglobalinc.org`). According to their vision, data about items should not be stored on the RFID tags, but be retrieved from the EPC Network by using the EPC as a search key. Looking-up data from the EPC Network for a specific EPC consists of two main phases (for generality, we disregard purely local information storage and caching here), cf. Fig. 2 and [1]:

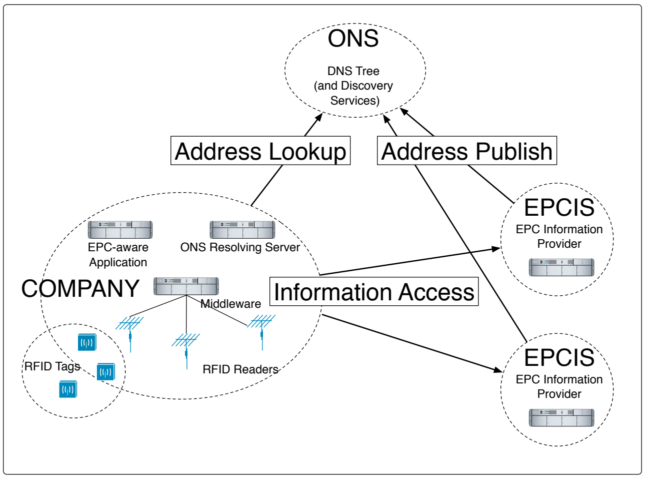


Fig. 2. The EPC Network

1) Lookup of EPC Information Services (EPCIS) that correspond to the tagged item at hand by using the Object Naming Service (ONS) and EPCIS Discovery Services (not yet specified [2]).
2) EPCIS access to retrieve the actual item information.

For ONS, a hierarchical distributed architecture [1] has

1-4244-0353-7/07/$25.00 ©2007 IEEE

been proposed by EPCglobal. ONS is based on the Domain Name System (DNS), its central ONS root will be operated by the company VeriSign. Further delegation works as in DNS, and information providers itself will deploy authoritative ONS servers (for their EPC space) and actual EPCIS (e.g., as Web Services). This architecture choice will have a deep impact on the reliability, security and privacy of the involved stakeholders and their business processes, especially for information clients.
Our article is structured as follows: First we discuss problems of the current ONS design (section II), then we give an informal summary of requirements that a system like ONS should fulfill (section III). Afterwards, we describe a basic architecture called OIDA using a DHT for ONS (section IV-B) and discuss possible security and especially privacy enhancing features for this example architecture (section V). Finally, we discuss related work (section VI) and conclude with some of our current and future research topics in this area (section VII).

II. Problems with ONS

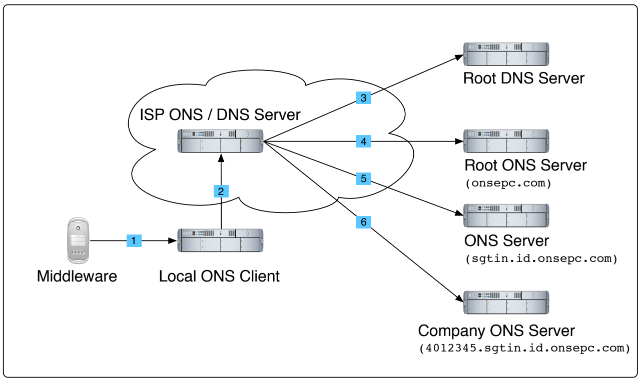


Fig. 3. The Object Naming Service (ONS)

To use DNS for ONS will inherit all of the well-documented DNS weaknesses, cf. [3] [4] and RFC 3833. Though distributed, DNS suffers from limited redundancy in practical implementations. Authoritative name servers for any given zone should be redundant according to RFC 1034. Recent studies on real implementations, however, show that for a non-insignificant part of the global name-space this formal requirement does not hold. Name servers holding the same information for a given zone are often few and not redundantly placed with respect to geographical location and Internet Protocol (IP) subnets, and often reside inside of the same Autonomous System. There are many servers that have single distinct routing bottlenecks on paths to reach them (from any place in the world).
The small number of servers for a given zone information, and their limited redundancy, create single points (or small areas) of failure. These are also attractive targets for Denial-of-Service Attacks – not only at the DNS root, which is run by fewer than a hundred servers, and has been attacked with some success before. Failure of the root, though, would (after some time) imply failure of the whole system, not only of some of its subtrees. Root and top-level domain (TLD) servers, as well as name servers for domains that rise in popularity (flash crowds, or Slashdot effect) suffer from strong load imbalance induced by the architecture. Omnipresent DNS caching, on the other hand, reduces flexibility and the speed of update propagation. Studies also show the significance of human configuration errors that (at least) slow down the resolution process [5] [6]. Part of the problem is the complexity of DNS delegation that is based on cooperation across different organizations.
Another hot topic of constant debate is a rather global political problem. Who should control and operate the root and TLD servers, and the name space as a whole? To let a single company, in addition to its major role in the DNS root and Certification Authority (CA) services, take control of the ONS root, might hinder international acceptance of the system as a whole. Not least, there is the vast history of implementation errors and bugs in DNS software, which will not be different for ONS. Exploits continue to be produced to control unpatched servers and the information they contain. Cache Poisoning pollutes the records stored by resolvers and non-authoritative name servers, Man-in-the-Middle attacks on the resolution paths are easily possible, as there are no widely deployed countermeasures at the DNS protocol itself, the transport layer (in most cases, DNS uses the connectionless User Datagram Protocol, UDP), or IP layer.
In the wake of consumer concerns about RFID [7], there is the important issue of client privacy. To use ONS, the EPC (as of now, without the last serial number part) is encoded as domain name (part of the domain onsepc.com), and then queried for by standard DNS resolution means. Whereas with classical DNS, analyzing the queries issued by an individual or organization can create profiles of their online habit, the same profiling techniques transferred to ONS profile entities with respect to physical items and their movements. This can include a large part of their belongings and personal travel, once RFID readers permeate society, and certainly product flows in supply chains. With a global EPC Network in place, concerns about the actual RFID tags and local privacy violations might refer just to the tip of the iceberg.

III. Requirements

In this section, we (informally) point out some of the most urgent requirements a lookup system S for EPCIS should fulfill. A deeper analysis, especially of the multilateral security requirements of RFID and EPC Network use in selected scenarios, is ongoing work at our institute, in cooperation with researchers from other universities.

1) Functional Requirements
   a) Primary function (mandatory): On input of EPC e, S should output a current list of servers that contain information about the object corresponding to e.
   b) Secondary function (optional): S should itself be able to store and return small amounts of object information about e to reduce query overhead.
2) Performance and Robustness

S must be able to deliver a performance that is suitable for global use (scalability for supply chains and Ubiquitous Computing environments). The following requirements have been identified in [4]:

   a) High Performance and Scalability: The system should be able to work on a planetary scale. If used for a so-called "Internet of Things", it is probable that in the long run S must cope with much more traffic than the use of DNS for URL name resolution generates today.
   b) Fast Update Propagation: Information changed by authorized entities should be propagated fast throughout the system, to avoid stale data.

3) Security and Privacy Requirements
   a) Resilience to Attacks: The system should avoid single points of failure, and be able to adjust itself to node failures.
   b) Data Authentication: Retrieved address and object information must be authenticated.
   c) Access Control: Information providers must have the ability to implement access control on the data they provide.
   d) Client Privacy: (i) (Strong Version) No one, except for the information provider (to fulfill the requirement on data access control), should be able to infer from observing the use of S which client asked information about what particular object. A much weaker variation: (ii) For casual attackers, this inference should be very hard to conduct.

IV. A Basic Architecture for ONS Using DHT

A. Advantages of Using DHT

Peer-to-Peer (P2P) systems in general have proven their scalability and performance in real-world applications. Structured P2P systems based on Distributed Hash Tables (DHT) in particular are the foundation for many distributed network services that already run in reality and on testbeds such as PlanetLab. Some of these aim to enhance and finally replace name services on the Internet. CoDoNS (Cooperative Domain Name System) [4], for example, combines the decentralization, scalability, simple administration and robustness of a DHT with a replication system called BeeHive to reduce the average lookup latency. In addition, it offers data authentication based on cryptographic delegation and DNS Security Extensions (DNSSEC). A straightforward application of CoDoNS to ONS could simply combine both systems as they are designed today. This would let ONS inherit the excellent performance and robustness properties of CoDoNS. In addition, some of the security requirements such as resilience to attacks and data authentication would be much better fulfilled than by using classic DNS alone.

Access control, however, could only be implemented at the actual EPCIS itself, not on the data stored in the DHT, as there is no encryption offered by any of these two designs. Consequently, the DHT could not easily be used to store actual item data, if it is subject to confidentiality requirements.

Even harder still is the fulfillment of the client privacy requirement. DNS is a pure clear text protocol, and DNSSEC is explicitly not going to change anything about that, because its function is pure authentication of entities and data, not encryption (RFC 4033). So, if DNS is used for ONS as proposed by EPCglobal, the client network identity (i.e., its IP address), the query, and the response can be easily read by any networking device in the resolution path (in addition to the endpoints of the query, and all other DNS servers used for resolution). Using DNS on top of a DHT like CoDoNS will obfuscate the query (to some extent) by querying not for the original name, but for the derived hash key to locate a DHT storage node. But the answer will not be obfuscated in any way, offering eavesdroppers potentially the same information as the query. The only advantage for client privacy would be against those attackers who are not able to analyze the client's network traffic, but rely on ONS server log analysis for profiling. The number of server nodes needed to be put under surveillance is probably harder to determine and larger when using DHT combined with a replication system like BeeHive.

Our thesis, that the most of the requirements could be better met by DHT than by classical DNS, cannot be considered finally "proven" by the work in this area (due to the lack, so far, of real system on the same usage scale as classical DNS), but it seems in our opinion highly plausible. Additional questions remaining are: Can data access control and client privacy also be enhanced by using DHT for ONS? How strong could such a protection turn out, and what else does it need in terms of additional infrastructure and key distribution?

The following conceptual design takes first steps towards answering these questions.

B. The Basic OIDA Architecture

We present a bare bones DHT architecture for ONS at a conceptual level. Our point is to analyze if and how access control and client privacy can be enhanced, while keeping the main DHT functionality unchanged. To have a reference name at hand, we call our conceptual system OIDA (Object Information Distribution Architecture). In OIDA, information providers publish address documents to the DHT for single EPCs or whole EPC classes (a convention could be to set the serial part to zero). These documents contain the address lists of corresponding information servers that are queried for by OIDA clients, or even (partial) information itself. To keep the decentralized and self-organized aspects of P2P systems, we do not demand security features on the nodes themselves, except for the ability to verify the identity of information providers. Nodes should not be trusted more than unauthenticated DNS servers used daily on the Internet. This design choice is made to investigate the limits of untrusted P2P systems, and to keep their advantage of self-organization. In a real implementation of nodes, for example on those special corporate hosts already designated to work as ONS servers, this choice could be lifted easily and additional (inter-)node security be implemented. For now, security features in OIDA are put into the stored documents.

The hash value of an EPC plays two important roles: first, as a DHT lookup key, second, to some extent, as a privacy-enhancing measure to avoid sending the EPC in clear text across third party networks. How strong this second feature really turns out in practice, however, depends on its ability to withstand dictionary attacks that try to precompute the hashes for all probable EPCs, we will discuss this further in section V. To increase the strength of this privacy aspect of the protocol, it would be helpful if one could assume that the information provider and client do share a common value s. s could then be used as a salt for the hash function. We discuss its role and particular requirements in section V. If such a s is unfeasible due to lack of secure distribution channels, or is unwanted because unrestricted search is required, s can be assumed to be the empty string.

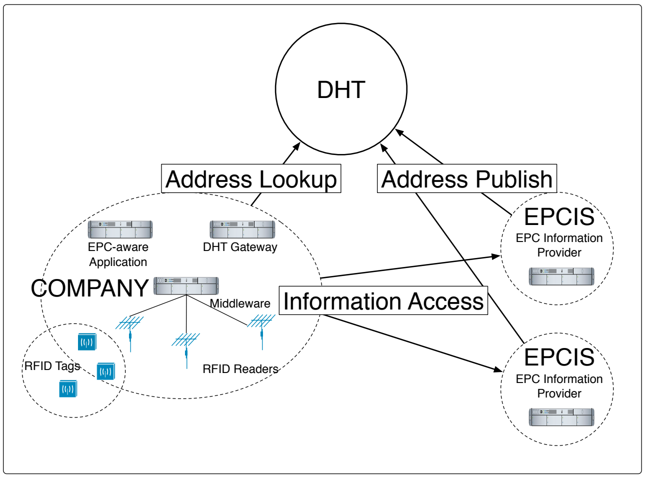


Fig. 4. OIDA Architecture

An information provider S who likes to publish information i (e.g., the address of a corresponding EPCIS) for an EPC e first creates a document containing its name S, the information i, and (for additional verification purpose) the hash h of e and s. S signs this document by using his private key. To implement access control and to reduce inference attacks from the data included in the returned document, it should be encrypted. How this could be achieved is discussed in section V. The final document d is then stored in the DHT at the nodes responsible for $h(s, e)$. This step could include verification of S by the responsible nodes to avoid spam, and the use of caching layers for the DHT such as BeeHive [4]. A topic for future research in proactive caching would be the feasibility of popularity metrics for single EPCs and their hashes, as well as for whole classes of EPCs, and what privacy implications would follow from such proactive caching methods. At the moment, we simply demand redundancy of data storage nodes to avoid single points of failure.

Later, the client C (in possession of salt s, see V) requests information about e by issuing a request for $h(s, e)$ to the DHT. The DHT replies by sending d to C who then verifies the signature of S to determine if S really created this document. This may include the addition of a public key and a certificate signed by a CA (that should be trusted by C) to the document interior. If C trusts the certificate and likes to fetch information from S, a direct connection (e.g., via Web Services) to the address i stated by S is established. This EPCIS access can be authenticated and encrypted, e.g. by Transport Layer Security (TLS). At the conceptual layer, the basic OIDA protocol works as follows (see Fig. 5):

1) Publish: $S \longrightarrow DHT$: *dht-store*$(h(s, e), d)$. d is a document containing information (i.e., EPCIS addresses or actual data) regarding the EPC e, and potentially additional security features.
2) Lookup Request: $C \longrightarrow DHT$: *dht-request*$(h(s, e))$.
3) Lookup Reply: $DHT \longrightarrow C$: d.
4) C verifies d (cf. section V), and if genuine, starts a request to the EPCIS S (located at a network address i extracted from d), using TLS.
5) EPCIS reply from S to C, using TLS.

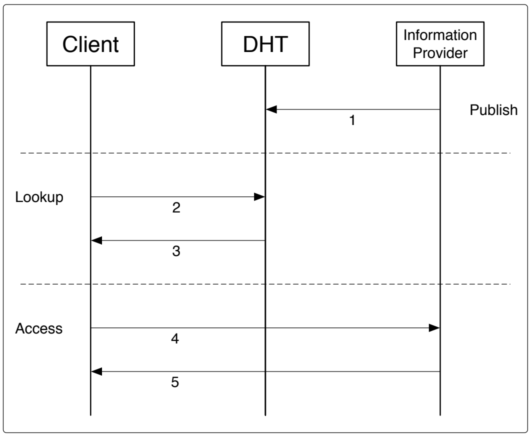


Fig. 5. OIDA Protocol

V. Security and Privacy

A. Key Distribution as a Prerequisite?

In section IV-A we identified data access control and client privacy as requirements not fulfilled by DHT-based systems like CoDoNS (certainly not by classical DNS). A central question for estimating the kind and strength of the cryptography that can be used to achieve these goals is: Will there be something more than the EPC that is shared between information provider and client? Will they share a common value, perhaps even a secret? Will there be a global Public-key Infrastructure (PKI) in place that makes the use of asymmetric cryptography and certificates feasible, especially for the client side?

Depending on the answers, different modifications to OIDA can be designed. As indicated in section IV-B, security features would be implemented at the document level. For data authentication (e.g., DNSSEC), this is straightforward by signing the information before storing it. In the following, we discuss the options for client privacy (confidentiality) and access control.

B. Client Privacy

For pure data authentication (e.g., using DNSSEC), a PKI-like hierarchical certification system needs to be in place – for the information providers, not the clients. Though it is probable that some kind of global PKI will be run by EPCglobal for supply chain use [2], it is not clear yet if this would be opened for or even scale to the much larger set of possible Ubiquitous Computing applications. Therefore, assuming a global PKI to include all clients seems to be quite a strong requirement [8] (not withstanding non-global PKI use for access control on some of the data stored in the DHT, see below). On the other hand, can we assume the existence of shared keys between information providers and clients? For EPC tags standards include kill and access passwords. If these passwords are transferred securely from manufacturer to the store and finally to the end user, it would be easy to transfer a third password k on the same channel (and a salt s for increasing the protection of the hash value against precomputation and dictionary attacks) for accessing online information. k could be used to encrypt the stored document d. In reality, though, shared secrets scale badly, are hard to manage and distribute securely, and have huge usability problems if there is no local management device (e.g., a PDA) at hand, which itself could become a target for attacks. However, key distribution does at least not seem impossible.

What can be done, if information provider and client share nothing but the EPC? First, the query: The hash value is in theory computed over a space of at least 2^{88} possible inputs (the space of all possible SGTIN-96 EPCs, minus header bits). This would not be bad as a protection against even more advanced attackers. In practice, however, only a small fraction of this space would be in use at a given time. Depending on the diffusion of the EPC standard it is quite possible that the actual number of EPCs to precompute the hash values is small in comparison. In addition, the EPC is highly structured (see Fig. 1), and serial numbers might be created in a regular, non-random fashion. This would further reduce the effort to derive the preimage EPC from a captured hash value. Second, the reply: The stored document could be encrypted by using a value somehow derived from the EPC in a publicly known way (e.g., by hashing twice, or using a second hash functions). But the same problem of probably small search space would apply to encryption. If the EPC is identified by a dictionary attack the encryption would be useless at the same instant.

To conclude, without any additional shared value between provider and client the privacy protection offered by the hash function and encrypted documents would only help against casual attackers that skim the network traffic. However, if it could be managed to share a truly random salt s (changed randomly for different clients) between provider and client, dictionary attacks on the hash function would become much harder (cf. also RFC 2898). A final problem could be the ongoing discovery of weaknesses in hash functions. Research for a new standard in hash functions would help to increase client privacy, too.

C. Access Control

If access control on the stored data is a required by the publisher, the publisher needs to offer a way for clients to authenticate themselves, e.g., by issuing shared secrets or using public-key cryptography. This key material needs to be distributed using secure channels, separately from the actual system in use. The same key material, however, can be used by the information provider S to encrypt the document d stored in the DHT. If necessary, multiple copies encrypted by different keys can be stored in the DHT – or different information documents for different clients. To locate these documents, the hash value could be computed using the EPC and some identifier or other value s shared by client and publisher (e.g., client Distinguished Name, or hash of its public key). Even though third parties could analyze the network traffic and locate the document, they cannot read it without the corresponding shared or private key. Against those attackers, client privacy would also be enhanced. If the additional identifier turns out to be too regular to help against precomputation, it should be easy to use a true random salt and distribute it to the client using the same out-of-band channel used for key material.

D. Open Questions in Client Privacy

Even if the actual ONS query would stay confidential against eavesdroppers, a potential subsequent DNS request would be not. So it is important either to store pure IP addresses in the DHT document d, or to include additional name resolution features. If the two phases are kept separate the EPCIS access, though encrypted, could give hints about the nature of the queries issued. One solution, besides using stronger anonymity systems like TOR [9], would be to store more naming information in the DHT itself. Finally, there is the question who has to be considered as potential adversary. In this article, we considered the actual service providers as trustworthy, and examined only protection methods against third party eavesdroppers. If this does not hold, much stronger anonymity measures would have to be implemented. This is one topic for future research.

VI. Related Work

A. Alternative DNS Architectures

To improve DNS robustness and performance by using DHT [10], several designs have been proposed, see e.g. [11] [4]. However, comparing the performance of these approaches to each other (or even to reliably measure the performance of the existing DNS infrastructure) is meeting methodical and practical difficulties [12] [13]. But implementations of DHT with additional proactive caching layers (BeeHive) on PlanetLab indicate that DHT seem to be quite capable of the task [4]. Some hybrid architectures have also been proposed, e.g. [14] [15]. But so far, all approaches did not foresee the privacy and security issues caused by RFID and EPC, resulting in additional requirements for lookup systems like ONS.

B. Privacy-enhancing Technologies (PET)

There is much past and current work on enhancing the privacy of users of network and service infrastructures. Important approaches include mix networks and private database access. Mix networks, for example used in onion routing systems like TOR [9], are general-purpose systems that offer a high degree of anonymity for its users. TOR is very important for user privacy on the Internet, but we argue that a newly designed service like ONS should offer client privacy on its own, without depending on optional external (and demanding) measures. Both mix networks and P2P systems can offer enhanced anonymity [16]. Freenet [17] is an anti-censorship system that even combines elements of mix networks and P2P systems. But DHT-based P2P systems promise better performance as a look-up service than mixes, because the latter require extensive cryptographic operations on intermediate nodes. Yet, more extensive studies of this performance vs. anonymity trade-off (using different and realistic traffic patterns and attacker models) have to be conducted before a final conclusion can be reached. Methods for private database access could be adopted for EPCIS access, but seem yet to lack scalability and performance for use in global and dynamic lookup services.

There is also much research on distributed storage systems, including anonymity and censor-resistance [18]. Some of these systems could be quite capable as a replacement for ONS and even the EPC Network. The main problem we see is the possible lack of acceptance on the information provider's ("owner") side to "let go of their information" and store it somewhere in untrusted systems without access control. The same problem will occur with DHT, and for this reason we think it practical to keep the two phases of the original design (i.e., lookup and EPCIS access). If this objection should not hold in future, distributed storage systems (often based on P2P) with client anonymity and access control are interesting alternatives to the EPC Network as a whole.

VII. Conclusion

Using a DHT for ONS will fulfill many of the system requirements stated in section III better than DNS. In addition, access control could be implemented on the documents. Without appropriate key or salt distribution methods, however, client privacy can only be slightly enhanced by protecting queries and responses from casual eavesdroppers. In this case, stronger anonymity systems might become necessary.

Our current and future work in this area includes the methodical elicitation of multilateral security and privacy requirements [19] in selected EPC scenarios, designing ONS and EPC Network alternatives that offer stronger client anonymity, and the adaptation and application of anonymity metrics [20] [21] and prototypes to evaluate and compare different design approaches.

References

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

[2] K. Traub (Ed.), "The EPCglobal Architecture Framework Version 1.0," 2005. [Online]. Available: http://www.epcglobalinc.org

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

[4] V. Ramasubramanian and E. G. Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2004, pp. 331–342.

[5] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang, "Impact of configuration errors on DNS robustness," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2004, pp. 319–330.

[6] D. Wessels, "Is your caching resolver polluting the internet?" in *NetT '04: Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*. New York, NY, USA: ACM Press, 2004, pp. 271–276.

[7] O. Günther and S. Spiekermann, "RFID and the Perception of Control: The consumer's view," *Communications of the ACM*, vol. 48, no. 9, pp. 73–76, September 2005.

[8] J. Lopez, R. Oppliger, and G. Pernul, "Why have Public Key Infrastructures failed so far?" *Internet Research*, vol. 15, no. 5, October 2005.

[9] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[10] H. Balakrishnan, M. F. Kaashoek, D. R. Karger, R. Morris, and I. Stoica, "Looking up data in P2P systems." *Communications of the ACM*, vol. 46, no. 2, pp. 43–48, February 2003.

[11] R. Cox, A. Muthitacharoen, and R. Morris, "Serving DNS using a peer-to-peer lookup service," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer, 2002, pp. 155–165.

[12] R. Liston, S. Srinivasan, and E. Zegura, "Diversity in DNS performance measures," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. New York, NY, USA: ACM Press, 2002, pp. 19–31.

[13] J. Pang, J. Hendricks, A. Akella, R. D. Prisco, B. Maggs, and S. Seshan, "Availability, usage, and deployment characteristics of the Domain Name System," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM Press, 2004, pp. 1–14.

[14] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the Internet," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2004, pp. 343–352.

[15] Y. Doi, "DNS meets DHT: Treating massive ID resolution using DNS over DHT," in *Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'05)*, 2005, pp. 9–15.

[16] R. Böhme, G. Danezis, C. Diaz, S. Köpsell, and A. Pfitzmann, "On the PET workshop panel – Mix cascades versus Peer-to-Peer: Is one concept superior?" in *Privacy Enhancing Technologies (PET 2004)*, ser. LNCS, D. Martin and A. Serjantov, Eds., vol. 3424. Springer, 2005, pp. 243–255.

[17] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with Freenet," *IEEE Internet Computing*, vol. 6, no. 1, pp. 40–49, Jan.-Feb. 2002.

[18] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.

[19] S. Guerses, J. H. Jahnke, C. Obry, A. Onabajo, T. Santen, and M. Price, "Eliciting confidentiality requirements in practice," in *CASCON '05: Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2005, pp. 101–116.

[20] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Privacy Enhancing Technologies Workshop (PET 2002)*, ser. LNCS, R. Dingledine and P. Syverson, Eds., vol. 2482. Springer, 2003, pp. 54–68.

[21] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Privacy Enhancing Technologies (PET 2002)*, ser. LNCS, R. Dingledine and P. Syverson, Eds., vol. 2482. Springer, 2003, pp. 41–53.