Computers in Industry xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT

Benjamin Fabian *, Tobias Feldhaus

Institute of Information Systems, Humboldt-Universität zu Berlin, Spandauer Str. 1, 10178 Berlin, Germany

ARTICLE INFO

Article history:
Received 27 March 2014
Received in revised form 1 June 2014
Accepted 1 July 2014
Available online xxx

Keywords:
Privacy
Home appliances
RFID
P2P

ABSTRACT

Smart homes are about to become reality, involving technology such as Radio-Frequency Identification (RFID), which enables querying for extended product information from the manufacturer or from public databases. However, every query for information issued from a smart home could adversely affect the privacy of its inhabitants. Since every Electronic Product Code assigned to RFID-equipped objects is unique, it is easily possible to create detailed and long-term customer profiles by linking queries on particular items to a specific person. In order to address the privacy problem, this paper proposes a novel peer-to-peer (P2P) infrastructure for organized sharing and private querying of data, which is formed by many smart devices across several homes. The efficiency of this approach is examined with experiments involving several hundred nodes. Furthermore, larger peer-to-peer networks of smart home appliances are simulated. According to our evaluation, the proposed solution is able to satisfy real-time requirements in settings where smart devices are geographically close. Moreover, the architecture can be used to store information of nearly one million different products within a network of one thousand nodes, which is a reasonable size for a local collaborative infrastructure between smart homes in towns or cities.

1. Introduction

Intelligent homes have been envisaged at least since the 1980s: in an early vision of the year 2010, Skrzypczak describes a kitchen with flat video monitors attached to the refrigerator, which can be controlled via voice recognition and used for preparing shopping lists that automatically get transmitted to a "*grocery shop at home service*" for order fulfillment [1]. Today, this vision is about to become reality. Besides classical home automation, there are new developments that help to foster its realization. The *Internet of Things* (IOT), which in the supply chain community is understood as a global information network for smart objects based on wireless and Internet technologies, has been widely adopted in the industry, and can be regarded as predecessor of an even more extensive *Web of Things* built from even smarter devices with more elaborate communication abilities between heterogeneous local and remote Web services [2]. *Radio-Frequency Identification* (RFID) technology for automatic inventory tracking and theft prevention has been applied by Wal-Mart and many other companies. RFID tags are integrated or attached to products in a similar way as barcodes, but can be read out without line-of-sight [3]. In order to increase transparency of supply chains, global item-level tracking via RFID has been introduced in many companies. This makes it easier for firms to analyze and automate processes and quickly react to changes. However, products in grocery stores, or arbitrary products for the mass consumer market, are not yet usually equipped with an RFID tag, mainly because as of today they are still considered too expensive on a per item basis. The 'smart fridge', a refrigerator that is able to track its inventory, such as the one currently realized by the company LG, is therefore still relying on the customer to manually scan the barcode of a product when it gets stored in the fridge [4]. Nevertheless, tag prices fell from about 13 cents in 2006 to 5 cents per tag in 2011 [5]. Furthermore, some refrigerators in the professional health care market come already equipped with RFID readers [6]. Therefore, a ubiquitous presence of smart home appliances using RFID could soon become reality. The market value for smart appliances in 2011 was already estimated at $ 509 million, more than ten times higher than in 2010 ($ 10 million), and a global revenue of $ 2.68 billion for smart refrigerators in 2015 is expected [7]. In another forecast from 2010 by Zpryme [8], the market value for smart appliances in

* Corresponding author. Tel.: +49 30 2093 5662.
E-mail addresses: bfabian@wiwi.hu-berlin.de (B. Fabian), tobias.feldhaus@gmail.com (T. Feldhaus).

http://dx.doi.org/10.1016/j.compind.2014.07.001
0166-3615/

2015 is estimated even at a much higher value of $ 15,175 million.

Using RFID technology in the mass market can have important benefits for the consumers: household appliances can easily be integrated into the IOT, resulting in a more convenient use, since the refrigerator, for example, knows at any time what items are stored inside of it. Automatic creation of grocery lists that can be synchronized with a smartphone of the user while she is away from home, and an alarm triggered by goods that are past their use-by dates yields a practical benefit as well. Additionally, via their smart fridges customers could automatically query product information from independent third parties, such as Greenpeace or Foodwatch who provide independent information about ingredients and their ecological background. When producers of goods start to attach RFID tags to their products, they can also easily send out product recalls to their customers with smart fridges. Because such a fridge knows what is stored inside of it, it can simply check the Electronic Product Code (EPC) against a propagated list of recalled products by the producers. This can help to decrease the direct and indirect costs of product recalls.

On the other hand, the use of RFID chips has raised many privacy concerns of consumers. In 2003, the Benetton Group began to sew RFID chips into their clothes for better inventory control. A boycott website was formed shortly after this news was revealed to public. People feared to be tracked and not being able to travel without leaving a digital fingerprint in the form of the unique identification number corresponding to each RFID chip, even from longer distances [9]. In fact, these chips were built to resist washing machines and were indeed intended to be used to monitor the paths that customers took in the stores. Soon it was realized that privacy threats to costumers would not be limited to local scanning and tracking of tags, but would also massively affect emerging global backend infrastructures for inter organizational RFID use, such as the EPCglobal Network [10,11]. This complex challenge of providing a privacy-friendly and robust infrastructure for retrieving information for smart home appliances motivated our research.

The main goal of this paper is to solve the problem of designing a privacy-preserving, reliable and scalable backend infrastructure for using RFID in smart home appliances, which offers substantial improvements on anonymity compared to earlier designs such as the one of [12]. Since losing privacy is one major fear for customers in the context of RFID, enabling them to use new services that rely on this technology while still taking care of their privacy is essential. On the other hand, usability is another important and complex factor, which is to no small extent influenced by the speed of information retrieval. The performance of our architecture is therefore examined by latency studies on Amazon EC2 and network simulations within the OverSim framework.

The rest of the article is structured as follows. Section 2 discusses technological background and related work. The privacy-preserving P2P architecture is presented in Section 3. Experimental evaluations of the proposed architecture by latency measurements and simulations are given in Section 4. Section 5 discusses the results, our contributions as well as open challenges, and gives an outlook on future research, followed by the conclusion of the paper in Section 6.

2. Background and related work

2.1. RFID and GS1 standards

RFID technology consists mainly of two components: tags, which are attached to objects of the real world or even directly integrated into them, and readers, which are able to read information from tags through radio waves [13]. Readers do not need a line-of-sight contact with objects, and multiple objects can be read simultaneously. This makes RFID superior to classical barcodes. Furthermore, the process of scanning barcodes is in most cases requiring some human intervention, for example to rotate packages for a direct line-of-sight contact. Additionally, barcodes can become scuffed and might therefore be unreadable or misinterpreted by a scanner. RFID technology enables two-way communication between objects and readers and is therefore the basis for the Internet of Things, where every communicating object carries its own global serial number and is uniquely identifiable. These serial numbers allow RFID tracking systems to count every tag only once and read every serial number correctly if a tag is not completely damaged and radio conditions are favorable [14].

Global Standards One (GS1) is an organization responsible for the definition and implementation of global standards for value chains in more than 108 countries. The association also offers code identifications for products. These identifications include barcodes and RFID identifiers, which are managed by EPCglobal, a subsidiary of GS1 [15]. The Global Trade Item Number (GTIN) refers to all trade items within GS1 system, which includes products as well as services. A trade item is defined by GS1 as "*any product or service that may be priced, ordered, or invoiced at any point in the supply chain*" [16]. The GTIN and other identification keys are used within the Global Data Synchronization Network (GDSN). The GDSN lets all participants share product data with one another in real time by so called *data pools*, a network of synchronized databases in different countries. In October 2009, the Foodservice GS1 US Standards Initiative was launched. It aims at letting 75% of the foodservice industry use GS1 Standards by the year 2015.

The Electronic Product Code (EPC) is extending the GTIN and can be viewed as a fundamental identification schema for the Internet of Things. The EPC can function as an identifier for any unique physical object (or its RFID tag); it is worldwide unique and, in theory, valid forever. In its most common form, an EPC is defined as a serialized string of 96 bit length. This string, called SGTIN, can be generated by encoding a GTIN together with a unique serial or product number. By using a unique serial number for each object of the same class, all objects are, in the end, distinguishable.

The EPC concept is superior to the barcode because it allows to differentiate between two bottles of milk from the same producer, even if the milk is originating from the same cow. However, no or only a few additional data items are stored onto the RFID tag itself: The main data is provided by the EPCglobal Network, an architecture for locating and accessing so-called EPC Information Services (EPCIS). All participants, such as manufacturers or retailers, are able to run their own local EPCIS and register it within the network. An EPCIS can be integrated into the IT landscape by several interfaces; the GS1 specification mentions XML, SOAP over HTTP, and AS2 [17]. The EPCIS provides corresponding data for each EPC that is stored on an RFID tag or sensor, and additional information such as the business context. In order to fulfill this task at a global scale and with many participants interacting with one another, a discovery service is needed, since the EPCIS of company A may not know about the product change of company B or even the product at all [11]. For this reason, an Object Name Service (ONS) and emerging Discovery Services (DS) [18] are situated between clients and EPCIS of participating companies. ONS and DS return the responsible EPCIS for a given identification key. By convention, ONS is targeted at locating EPCIS of the item manufacturer at the granularity of an object class, whereas DS aim to provide full serial-level lookup of EPCISs managed by several data stakeholders. Further components of the EPCglobal Network could involve dedicated traceability services [19].

2.2. Security and privacy in the EPCglobal Network and P2P alternatives

An early security and privacy analysis of the ONS was conducted by Fabian et al. [10]; ONS is based on the same

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

technology as DNS that resolves human-readable addresses into IP-addresses within the Internet, but involves new privacy problems. Concerning secure DS, Shi [20] propose SecTTS, a relay service that respects policies for information with varying sensitivity.

A research question is posed in this article if and how such a critical lookup service for RFID-based information for home and business applications could provide *privacy by design*, without relying on third-party anonymity systems. In follow-up research, Fabian and Günther [12] studied the impact of a P2P-based ONS on privacy. They formulated requirements that a global lookup system for EPCIS should provide, and presented a solution by storing naming information inside of a Distributed Hash Table (DHT) (more details on DHTs will be given in Section 3). The so-called *Object Information Distribution Architecture* (OIDA) would be operated by companies that deploy the necessary nodes, including a common Certificate Authority (CA) responsible for issuing and revoking certificates of nodes and providing a trust anchor for signing data published in the DHT. The DHT provides load balancing through consistent hashing, and it removes a single point of failure caused by the typical ONS architecture. A central entity that manages the nodes is no longer necessary. In [21], an OIDA prototype on PlanetLab was tested with more than 350 nodes spread over all continents. As alternatives to the EPCglobal Network, further P2P designs have been proposed. An information architecture for global supply chains based on the FreePastry DHT was simulated with up to 20,000 nodes in [22]. However, the simulation approach is very basic and not conducted with an established and detailed simulator such as OverSim and OMNeT++ used in the current article. Similar and partly more evolved P2P approaches were later developed in [23–26] and [27], which also includes an access control service, and [28] where also the network capacity of decentralized ONS and DS over DHTs was investigated. None of those approaches provides mechanisms for client privacy, however, in contrast to our works.

In [29], the authors studied client privacy where no earlier key-distribution to clients of a DS is available for cryptographically hiding the requests for RFID-based information. In order to improve privacy of a DHT in such a situation, they designed SHARDIS, a P2P-based discovery service architecture for the EPCglobal Network that offers improved client privacy by enhanced confidentiality of the lookup content. The main adversary model is a casually profiling insider that can be part of the DHT itself. Their general idea is to split a document that stores the information for a specific EPC into cryptographically derived shares using Shamir's Secret Sharing Scheme, and store each share at a different node. An implementation of SHARDIS, based on the Pastry DHT implementation FreePastry, was tested on PlanetLab with a latency low enough for automated inventorying and even interactive use. These approaches improved privacy by enhancing the confidentiality of the request. A different line of research focuses on improving client anonymity in DHTs. At the time of this writing, the state of the art with DHT anonymity is "Octopus", a lookup procedure for the Chord DHT that combines many earlier as well as new approaches and is offering several probabilistic guarantees with respect to security and privacy [30] (see Section 3).

3. Privacy-preserving P2P infrastructure for home appliances

3.1. Benefits of a P2P infrastructure

Because the market for smart home appliances is growing, and it is expected that corresponding services will be used by many people, an architecture for data exchange must be able to scale out accordingly. Simply storing all data on RFID tags would not allow for flexibly updating it once the products circulate. Moreover, additional data providers could not easily added. A basic client-server architecture would have several drawbacks since the server is a single point of failure and surveillance. If the server, or the central gateway, goes down or is not reachable due to a distributed denial of service attack, clients can no longer retrieve any information. Furthermore, this model does not offer sufficient load balancing without a huge server farm and several load balancers, or the outsourcing of such data services to external cloud or content-distribution providers, which are potentially subject to a different privacy legislation. Most importantly, this basic architecture does not provide any sufficient degree of client privacy: The manufacturer, curious third-party providers or other adversaries could easily analyze IP addresses of customer refrigerators and build up profiles. A similar argument applies to a network of servers which are located via discovery services, such as the EPCglobal Network (see Section 2).

In contrast, any structured P2P architecture has several major advantages for this particular use case: scalability, availability, and low operating cost. In a P2P network every participant acts as a client and as a server at the same time, therefore no server or server farm is needed. Furthermore, if the network protocol replicates information on a sufficient number of nodes, the network can deal with node failures and offers high availability. In addition to that, the system is fully decentralized and is therefore able to offer robustness. Moreover, the network organizes itself, which means no administration is needed. Most importantly, with every new node the total capacity of the network grows, it therefore scales automatically. The distributed storage of data is also beneficial for preventing adversaries from easily compiling profiles of clients. Therefore, in earlier works, some P2P architectures have been recommended for similar IOT use cases and successfully tested in multiple experiments, see Section 2.2, but none of them so far increases the level of anonymity of clients beyond the basic decentralized features of a P2P system.

In order to enhance the anonymity of clients, we suggest a privacy-preserving P2P infrastructure, which is formed by smart home appliances such as smart fridges in several households in the same area, for example a town or city. Consumer goods companies participate in the P2P infrastructure by providing their own nodes, and store information into the system according to the data-organization principles of the Distributed Hash Table (DHT).

Distributed Hash Tables are P2P systems that offer a lookup functionality analogous to a hash table, but in a distributed and decentralized fashion, involving multiple computers without central control. DHTs offer a simple lookup and storage interface based on a one-to-one correspondence between data items and keys. The underlying distributed DHT algorithms determine which nodes are responsible for storing the data by organizing keys and nodes in a logical overlay network, which is in general independent of the physical or IP network topology on lower layers, using concepts such as consistent hashing with only few local information about the whole system. Consistent hashing balances data items to nodes in a roughly uniform way, and allows for node joining and leaving without the need for major redistribution of keys and data in the running system.

Most DHTs resolve lookups in $O(\log N)$ hops through the overlay network, where N is the number nodes in the DHT, which offers excellent scalability. This scalability is enhanced by the fact that the overlay routing table size, which includes pointers or fingers to selected nodes that are very distant in the overlay topology, and the amount of state information stored at any particular node also scales with $O(\log N)$. This means that every node just needs to know a very small part of the whole overlay graph. DHTs also offer functionality like message routing in the overlay, node joining and leaving procedures in a

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

self-organized fashion. Our architecture is based on the Chord DHT. Chord [31] is a lookup service based on a DHT with a ring topology that functions as a database for (*key*, *value*) pairs. It provides a PUT method that stores a (*key*, *value*) pair at the correct node and a GET method for retrieving the value for a given key. This approach addresses the following problems: Load balancing is ensured via a consistent hashing function (here, Secure Hash Algorithm, SHA-1) that assigns keys and their values uniformly to the node identifiers of the system. Moreover, high system availability, self-organization and decentralization are offered, which means that all nodes are equally important and, if a new node joins or a node fails, the new responsible node for a key can be found after a short consolidation phase. Furthermore, the key-space is kept flat; it is up to the application to decide how names are mapped to Chord keys. Most important for scalability is that the cost of a lookup grows only logarithmically with the number of nodes in the system.

Once consumers buy goods and put them into their smart fridge, data can be retrieved from the P2P system, using the EPC or parts of it as pre-image for the overlay search key, which is generated by applying SHA-1 to the pre-image and determines the storage location. Information can be shown to users via screens of the appliance itself or sent to smart phones, PCs or tablets. Fig. 1 shows the communication between intelligent home appliances and the consumer goods companies.

3.2. Further privacy enhancements

At its core, client anonymity is provided by the decentralized nature of the P2P system and the Octopus lookup for the Chord DHT. Octopus involves a set of mechanisms that give several guarantees regarding security and privacy [30]. As important aspects for anonymity it provides initiator anonymity, target anonymity, as well as query unlinkability [32]. Octopus uses three techniques to achieve its security and anonymity goals: in order to hide the lookup target, nodes request the whole finger table from one another within a lookup over an *anonymous path*. An anonymous path is defined as using multiple encryption layers (similar to "onions" used in Tor [33]) combined with a random walk over several nodes for forwarding the query toward the target. Additionally, the initiator builds up and uses multiple anonymous paths for different queries in a lookup, and the nodes issue dummy queries in order to obscure the actual query from adversaries.

Furthermore, mechanisms for the removal of malicious nodes from the network are introduced. As malicious nodes could tamper with overlay routing (finger) tables and mislead lookups, Octopus uses these mechanisms in order to provide finger table and lookup correctness as well as finger table trustworthiness. In order to enable nodes to sign and encrypt messages, as well as for excluding adversaries from the network, Octopus uses a CA and a Public Key

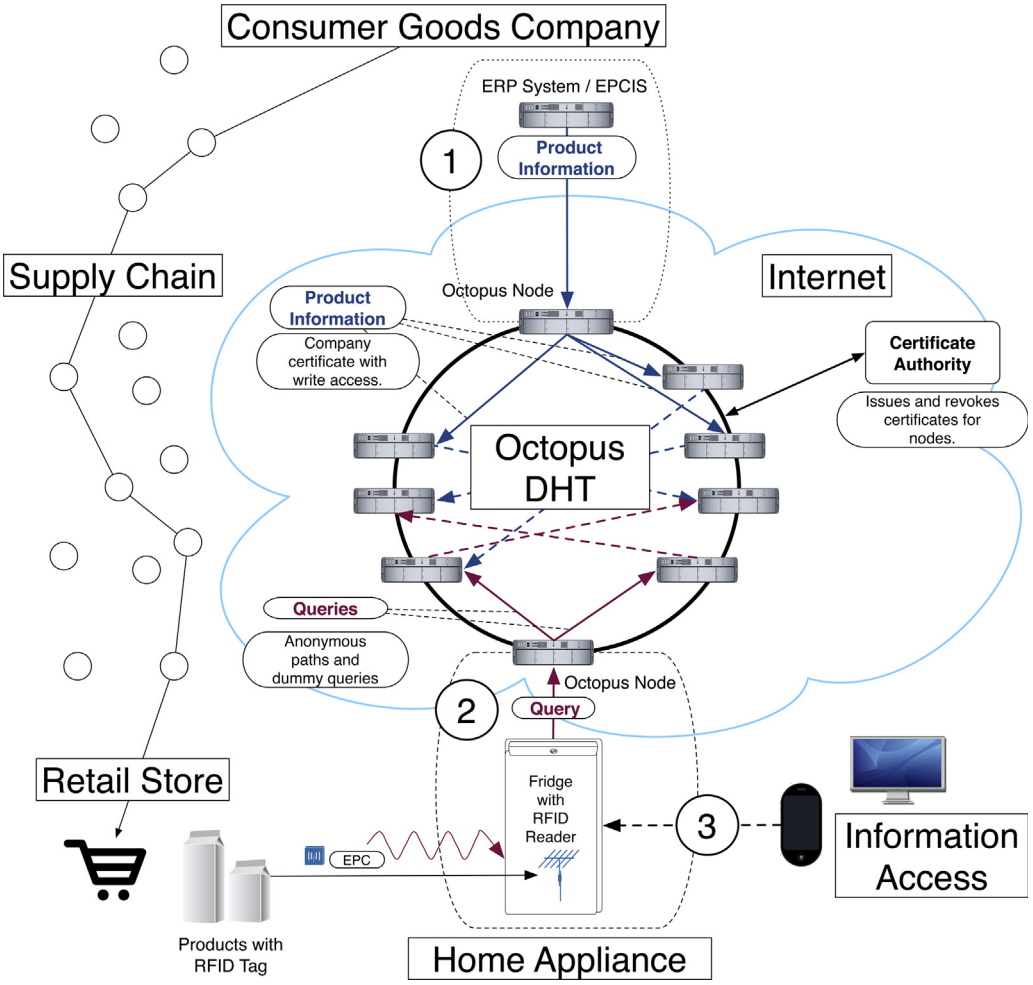


Fig. 1. The proposed infrastructure for storing product information using a Chord DHT with the privacy-preserving Octopus lookup: (1) companies and non-profit organizations store the product data of manufactured goods on the nodes. A valid product recall must be signed by the company that made the product. (2) Nodes are mainly provided by embedded systems inside of home appliances, a maximum of 20% company nodes are allowed. (3) Home appliances offer convenient access to product information.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

Infrastructure (PKI), similar to [21]. The CA is responsible for issuing and revoking certificates of the nodes. A certificate is a document that links a public key to a node by using a digital signature. The digital signature is, in this case, generated by hashing the certificate content (for instance a node identifier such as the IP address) and encrypting it with the private key of the CA. In this way each node can check with the public key of the CA that the certificate is valid.

Our proposed modified *Octopus DHT* differs in the following aspects from the approach presented by Wang and Borisov [30]: instead of providing a key-value store in which the value is very small and often just a link to an external database, in our approach the DHT is used to store the entire product information or at least all user-relevant parts of it directly into the DHT. Here, the SHA-1 hash of the EPC (for instance in its SGTIN-96 variant, or parts of it up to the object class) is used as a key, and the full product information as a value. In the current paper, the level of granularity at which we will evaluate our system is limited to queries at the product level, similar to ONS, not at the level of the serial number – but this does not imply that the system would be technically unable to do so. The current statistics about the amount of stored product information inside the GDSN databases, which are used for evaluating our architecture using real-world data, state how many product classes are currently listed. They do not allow to derive the total number of produced milk bottles of a specific milk bottle GTIN for example, since this involves potentially sensitive business data from companies.

Further differences in our infrastructure concern security. The certificates, which are needed for authentication and are provided as well as revoked by the CA, are issued as X.509 certificates [34] of two different types: smart refrigerators and producer nodes are assigned different values for a particular attribute, which allows one to distinguish between them. Consequently, a fridge only accepts and stores key-value pairs that originate from producers. In this way smart refrigerators cannot be misused to store irrelevant information or spam into the network. The certificate authority needs to be publicly operated, or at least controlled by the public, in order to ensure that there is no conflict of interest, as the privacy of the whole system relies on mechanisms that remove malicious nodes from the network by revoking their CA certificate.

Product data could originate from the Enterprise Resource Planning (ERP) or EPCIS systems of the consumer goods companies, but other sources can be integrated as well, for example databases from Greenpeace or Foodwatch. It is assumed that some companies may have a hidden interest in spying on consumers, for instance for creating customer profiles. As the Octopus lookup has been proven to be able to deal with up to 20% of malicious nodes [30], producer nodes should therefore not exceed 20%. In this way the privacy of the users can be protected even if all producers would collude against consumer privacy. The consumer goods companies profit from a reliable and robust system that they do not need to operate as a whole, except for a small number of nodes. As the product data is already available in their internal information systems, developing an interface to provide data for the Octopus DHT can be seen as a one-time expense. The supply chain delivers the products, each with a RFID tag attached, to a retail store where the customer is able to buy them. This brings several advantages for the store: RFID-tagged products have built-in theft prevention, as the point of sale system is able to recognize which product has already been paid for. Another benefit is the ability to scan multiple products at once, so it is no longer necessary to scan one barcode after another, and the inventory management of intelligent shelves is getting easier as they can check for misplaced or missing products and alert the staff [35,36]. When the EPC of the product is recognized by the home appliance, such as a refrigerator, it first checks its local cache to see if it has any information about the product available, otherwise it queries the Octopus DHT via multiple anonymous paths. In order to learn which node ID holds the desired information, the refrigerator calculates the SHA-1 hash of the EPC and provides it to the query procedure of the Octopus lookup. As noted before, Octopus relies on dummy queries to blur the observations of an adversary and hide the target of a lookup, therefore the system is constantly issuing queries. Thus, in order to deploy the system on a new refrigerator, it must be pre-loaded with realistic data before it can be used for privacy protection. In the given scenario, the companies know which EPCs are associated with real products, since they produced them, thus they can easily distinguish between dummy and real queries. Therefore, a refrigerator that joins the DHT must pre-load itself with as much data as possible, but at least have enough product types available in its local memory to be able to issue dummy queries for all products it contains.

Consequently, the home appliance is able to provide information such as the types of product it contains, their ingredients, or other product attributes such as allergens. The benefits of the system can now be safely realized by accessing the information on the home appliance through a smartphone or a browser. This enables the user to generate grocery lists automatically, or check if its fridge contains everything for a specific recipe. Also, the system checks for updates about the product information once per night. This way the user can be informed if a product recall has been issued. The fridge is then able to warn the user via acoustic or visual signals, or directly send a message to his smartphone. In order to issue a product recall at the current level of granularity, a general warning can be published at the granularity of product classes, which contains ranges of particular serial numbers. A company must own the respective certificate to sign a recall accordingly, which is the same certificate the product data was signed with when it had been stored in the DHT. Otherwise such a recall is considered unauthorized and ignored by the nodes.

In the following we discuss the major security and privacy mechanisms of Octopus that our design inherits. All of these are the contribution of [30]. Octopus makes use of a certificate authority (CA) as trust anchor. The CA does not need to be constantly online for the system to work, and the workload of the CA is considered manageable and diminishing over time as malicious nodes are removed. Octopus also involves a set of mechanisms to identify malicious nodes [30], which are then excluded from the system via certificate revocation.

The so-called *secret neighbor surveillance* watches the neighbors of a node and looks for successor list manipulations. By manipulating the successor list, a malicious node could introduce a colluding node as a direct successor of a key (*lookup bias attack*). The colluding node would then be regarded as the key owner. Secret neighbor surveillance uses a predecessor list that is composed and maintained in the same way as the successor list in Chord. An alternative way of tampering with the successor lists of trustworthy nodes is to send false lists throughout the stabilization phase (*successor list pollution*). While Octopus forces every node to sign its successor list, each node saves a series of the received successor lists to be able to prove to the CA that it has generated its successor list correctly, based on the information and successor lists that have been provided by the other nodes. The CA is able to verify the signed lists and to check one after another to find the node that is unable to come up with a valid proof. Consequently, the certificate of this node is revoked as it is marked as a malicious one. Another protection against malicious nodes, also introduced by Wang and Borisov [30], is called *secret finger surveillance*. As the name implies, it watches the correctness of finger tables provided by other nodes during former lookups or secret checks and uses a series of finger tables received and saved by each node.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

Other attacks, which are trying to identify nodes on an anonymous path via timing analysis by controlling more than two nodes on it, are difficult because Octopus is using one path per message. Additionally, one relay node on the path is introducing a random delay. Wang and Borisov [30] demonstrate the effectiveness of this approach in a simulation: With an error of 99.91% and an information leak of only 0.018 bit it is hard for adversaries to conduct a timing analysis.

For Octopus, Wang and Borisov [30] use a threat model where up to 20% of malicious nodes are controlled by an adversary, and do not consider to defend against the Sybil attack [37], where one adversary constructs a substantial amount of fake identities in order to gain influence. In Octopus, similar to OIDA and SHARDIS, the CA is responsible to lessen the effectiveness of the Sybil attack. Wang and Borisov [30] conduct several simulations to find out how fast their mechanisms remove malicious nodes from a network: Within a network of 1000 nodes in total and 20% malicious ones, they claim that after 20 min nearly all malicious nodes, which are trying to conduct a lookup bias attack, are discovered by the secret neighbor surveillance. A finger table pollution attack and the manipulation of finger tables are prevented by the secret finger surveillance. Wang and Borisov [30] report that the detection of these attacks shows a lower accuracy (higher false negative rates between 14% and 20%), but more than 80% of malicious nodes are identified within 30 min.

Furthermore, they provide an anonymity analysis based on the entropy metric, which is a theoretic anonymity model introduced by Díaz et al. [38] that quantifies the degree of anonymity of a system by investigating how much information it is leaking. They compare the initiator and target anonymity of Octopus with original Chord [31], Halo [39], NISAN [40], and Torsk [41], which are other recent contributions concerning anonymous systems.

4. Experimental evaluation

4.1. Latency as an important evaluation metric

Besides anonymity, formalized by Díaz et al. [38] as an entropy metric of how much information an anonymous system leaks and evaluated in depth for Octopus by Wang and Borisov [30], and efficiency of malicious node detection, latency is a crucial performance metric for the end-user of a privacy-preserving network solution. In modern networks, such as the backbone of the Internet, latency can be expressed as a function of the speed of light, since cables are often composed of fiber optics. However, in reality, not only the fiber and amplifiers are responsible for additional delay, but also other hardware and in particular routers contribute to network latency. Therefore, in a theoretically ideal test scenario for measuring the latency of a lookup algorithm, there would be only one type of machines, capable of the same transmission rate and connected over the same type of cable. However, distributed system research is also using more realistic global deployments under not exactly repeatable, but real-world conditions.

Using this approach, our privacy-preserving infrastructure as presented in Section 3 is investigated in the following. Our first key question: Is Octopus fast enough to respond to lookups in less than a few seconds? Usability studies on the Tor anonymization network have discovered a higher user tolerance of latency for interactive tasks when privacy-protection is involved [42,43]. Though a direct transfer to our context may be debated without further user experiments based on prototypes, such a latency at least allow for efficient background retrieval of information. Our second key question: is the Chord DHT able to handle the load caused by storing realistic amounts of product information in it, and how is the response time?

This study adopted two main research methodologies: Firstly, experiments were conducted that investigated the effects of network distance on latency for the Octopus DHT on Amazon EC2, both for geographically centralized and distributed nodes. Secondly, the impact of increased network traffic caused by consumer goods companies during their storage of product information into the DHT was investigated with the help of the OverSim network simulation framework. Moreover, a small experimental study was conducted that indicated the practical feasibility of the cryptographic operations involved with Octopus when they are implemented on a current chip suitable for smart home appliances (see Appendix C).

4.2. Latency experiments in the cloud

In the following, the geographically centralized and distributed measurements represent two extreme scenarios: an optimal case for latency, in which all refrigerators (nodes) are in the same town and connected over short distances, and the worst case where nodes are located on different continents. It was assumed that distance massively impacts latency efficiency: network latency and physical distance of two nodes should be positively correlated. In a pre-study, differences between both treatment groups were compared. In the main study, latency efficiency for a larger set of distributed nodes was measured. Qiyan Wang kindly provided us with the C++ simulator code used in their experiments [30]. This was adapted and modified to deploy the code for our experiments on Amazon EC2. In order to make result comparable, no changes were made to the measurement technique or the lookup algorithm itself. A region in EC2 consists of several availability zones, which mask the underlying data-center facilities hosting the virtual machines. The three EC2-regions, as shown in Fig. 2, were selected for deployment because of their geographical distribution.

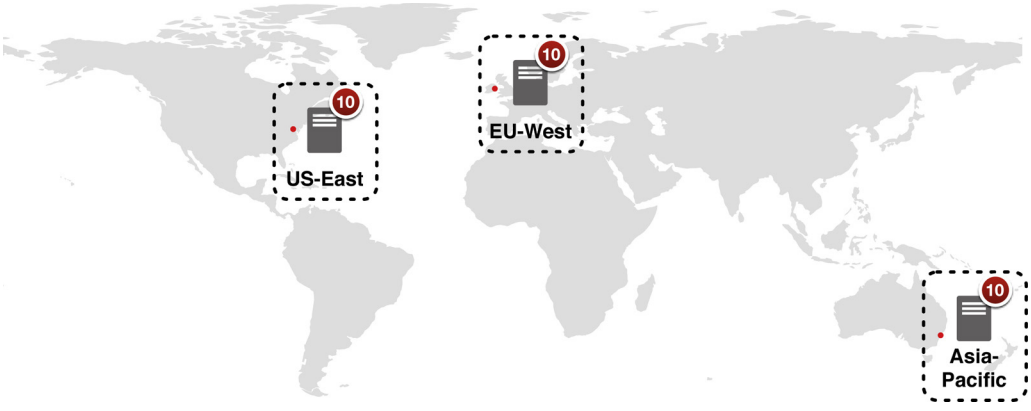


Fig. 2. The distributed pre-study with 30 nodes on Amazon EC2. (For the centralized pre-study, 30 nodes were deployed in the region US-East.)

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

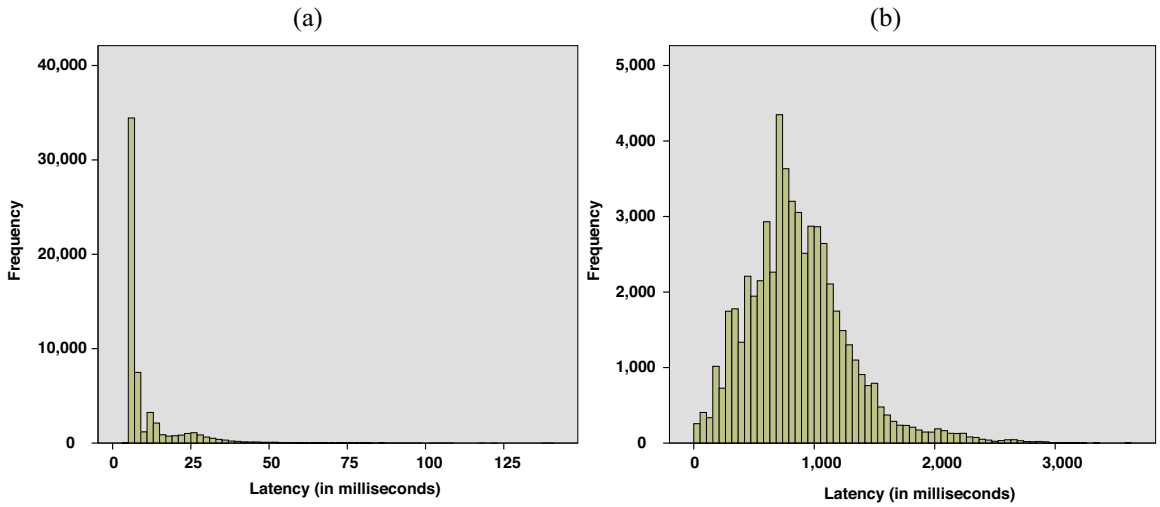


Fig. 3. Pre-study on EC2. Latency histograms in the centralized and distributed scenarios of the Octopus DHT lookup on 30 EC2 nodes. Both figures include outliers.

In both pre- and main studies, each node performed 2000 lookups independently with random lookup keys. Lookup latency was measured as follows: each node recorded the passage of time between t_1 when the lookup query was sent out by the node, and t_2 when the node received the correct lookup result. Each node performed on average three lookups per second until 2000 had been reached. Incorrect or ambiguous lookup results were marked as such, as were lookups that did not return a result within the test phase of 20 minutes. Those were considered failed lookups in the experiment. The parameters of the DHT, which are defined in the code, were left unchanged. As a result, the size of the finger table was twelve, as was the size of the successor list. As discussed, the latency measurements on EC2 aimed at comparing a centralized deployment with a distributed deployment. Therefore, the experiment was once performed centralized, with n nodes all from the same region, and once again decentralized with $n/3$ nodes in each of the three regions (see Fig. 2 for the regions selected). For this reason, the number of nodes n was chosen to be a multiple of three.

4.2.1. Pre-study on latency in the cloud

For ensuring validity and consistency of the experimental setup, an initial pre-study was conducted. This tested the experimental conditions within the network and deployed 30 nodes for analyzing the impact of both scenarios, i.e., centralized and distributed network communication on latency efficiency (see Fig. 2). Ten nodes were deployed per region, 30 nodes in total, each on its own virtual machine instance. For the centralized pre-study, 30 nodes were deployed in the region US-East. The same sampling and measurement procedure was also used for the subsequent main study.

During data collection, latency values for 120,000 lookups in total were collected, with 60,000 measurements per treatment group. The centralized pre-study reported an error rate of 3.4% with $N = 57,962$ valid measurements for latency evaluation. For the distributed pre-study, an error quote of 3.39% was reported, with $N = 57,966$ valid data points. The error rate lays in an expected range, as Octopus DHT is based on Chord and should have a slightly higher error rate than Chord due to its design. (The Chord GET success rate with a standard payload length of 20 bytes was near 100% in the simulation, as seen in Fig. 8a.) Comparable to the simulation in OverSim, the experiment on Amazon EC2 had near optimal network conditions and no churn, therefore the experimental design was considered valid. Fig. 3 shows the histograms for the centralized and distributed pre-studies.

Without missing values, but including outliers, the centralized lookup was more than 100 times faster in terms of the median (centralized: $M = 10.27$ ms, SD = 9.221 ms, Mdn = 6 ms; distributed: $M = 867.85$ ms, SD = 423.239 ms, Mdn = 819 ms). As the Amazon EC2 network traffic was routed over very long distances for the distributed test setup and the Octopus DHT prototype is based on the User Datagram Protocol (UDP), there was neither a guarantee of the delivery, nor the ordering of packets or protection from duplicates. Furthermore, [44] reported that virtualization on EC2 could cause abnormal delay variations. Therefore, a random sample was drawn from an outlier-cleaned[1] data set with 50,000 data points per group. As the outlier lower bound was negative (the value for the interquartile range was 5 ms for the centralized group and 495 ms for the distributed group) and latency is generally a positive value, only the outlier upper bound (centralized: 18.5 ms, distributed: 1831.5 ms) was considered for filtering values.

The randomized outlier-cleaned sample showed a slightly lower median latency for the distributed group (centralized: $M = 7.15$ ms, SD = 2.688 ms, Mdn = 6 ms; distributed: $M = 826.44$ ms, SD = 356.617 ms, Mdn = 804 ms). The 25th percentile value of the centralized (distributed) measurements equaled 6 ms (586 ms), the 50th percentile – which is by definition equal to the median – equaled 6 ms (804 ms), and the 75th percentile, 7 ms (1063 ms). The empirical cumulative distribution function (ECDF) in Fig. 4 illustrates that all lookups were answered within two seconds in the outlier-cleaned sample of the pre-study. It was concluded that the pre-study latency measurements on Amazon EC2 were comparable to the ones reported by Wang and Borisov [30] from PlanetLab. Thus, the experimental setup was considered correct and the main-study could be conducted.

4.2.2. Main study on latency in the cloud

The data collection for the main-study consisted of latency values in milliseconds for 954,000 lookups in total, with 476,000 (238 nodes) measurements for the centralized treatment group, and 478,000 (239 nodes) for the distributed treatment group.[2] The network using a centralized distribution of nodes reported an error rate of 0.5%, with $N = 473,553$ valid data points for latency evaluation. For distributed network communication, an error rate

[1] Outlier removal using the interquartile range (IQR) = $Q_3 - Q_1$, outlier lower bound: $Q_1 - 1.5 * IQR$, outlier upper bound: $Q_3 + 1.5 * IQR$ (see [45]).

[2] Amazon EC2 did not provide all 240 nodes that have been requested for both groups. This issue is discussed in Section 5.3.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

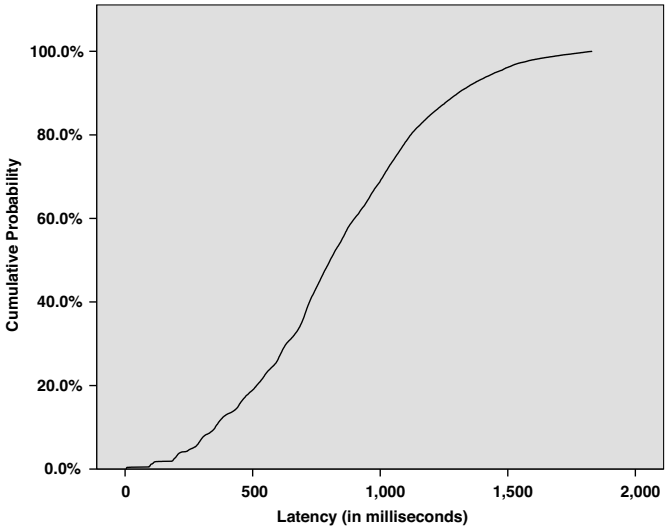


Fig. 4. ECDF of latency in the distributed pre-study of the Octopus DHT lookup on Amazon EC2 (ten nodes in each of the three regions). It is based on a random sample ($N = 50,000$) of the outlier-cleaned values.

of 3.3% was reported, with $N = 462,326$ valid data points for further evaluation. The error rate of the centralized setup was six times lower than the distributed setup, which is expected, given the geographical distances of the nodes (centralized: $M = 21.38$ ms, SD = 14.862 ms, Mdn = 14 ms; distributed: $M = 1504.61$ ms, SD = 673.704 ms, Mdn = 1466 ms).

This data set was further processed using the interquartile range to produce an outlier-cleaned version. For this the outlier upper bound[3] (54.5 ms for the centralized and 3372.5 ms for the distributed scenario) was used to discard any values greater than the respective value in both groups. As there were only positive values in both groups, the lower bound was irrelevant. Thus, 19,949 measurements were deleted in the centralized data set and 3357 in the distributed data set. As in the pre-study, randomized samples were pulled from both groups, with $N = 450,000$ data points for each group ($N = 900,000$ in total). The outlier-cleaned sample had a slightly lower median for the distributed group (centralized: $M = 19.34$ ms, SD = 11.213 ms, Mdn = 14 ms; distributed: $M = 1488.18$ ms, SD = 647.538 ms, Mdn = 1459 ms). The 25th percentile value of the centralized (distributed) measurements equaled 12 ms (997 ms), the 50th percentile equaled 14 ms (1459 ms), and the 75th percentile equaled 26 ms (1936 ms).

The histograms of the centralized (Fig. 5a) and the distributed (Fig. 5b) outlier-cleaned samples feature different distributions. The centralized sample shows a distribution skewed to the right (positively skewed), while the distribution of the distributed sample is similar in shape to a gamma distribution. The differences in the distributions of both test samples are also statistically significant. Both of these statistically independent, random samples are not normally distributed according to a Kolmogorov–Smirnov test ($\alpha < 0.001$), and both show different homogeneity of the variances with Levene's test ($\alpha < 0.001$). Mood's median test, which tests for differences between groups if the data is not normally distributed with non-homogeneous variances, indicates that the difference of the medians of both groups is highly significant (also with $\alpha < 0.001$). This is particularly obvious in the nearly overlapping error bars in Fig. 6a.

Furthermore, the analysis also included a sorting of the 450,000 measurements (of the outlier-cleaned sample) in the distributed scenario and the calculation of the percentage of cases in which the lookup was answered within two or three seconds: $(1/4500) \sum_{i=1, x_i \leq T}^{450,000} 1$. Here, the threshold T is defined as 2000 ms (respectively 3000 ms). For each case (x_i) that has a latency less than or equal to T, the sum is increased by 1, and in the end, divided by 4500 in order to get the percentage of cases that are equal to or below the threshold. This analysis revealed that in 64% of the cases, the correct answer to a lookup call was received within two seconds, and in 97% of the cases within three seconds. These results are also visualized by the ECDF in Fig. 6b.

4.3. Simulation of larger networks

4.3.1. The OverSim simulation framework

OverSim is a simulation framework for overlay and P2P networks [46], and was developed at the Karlsruhe Institute of Technology. At its core it is based on the *Objective Modular Network Testbed in C++* (OMNeT++), which is a modular and component-based simulation framework for network simulators [47]. OverSim has been shown to be an important advancement compared to other P2P simulators such as P2PSim [48,46]. From a performance perspective, the *Simple* underlay model is the best choice for a network model, as it is able to simulate the highest number of nodes on a given hardware. In order to accomplish this, it uses a global routing table and delays each packet either by a fixed period of time or a delay that is derived from the distance of the two nodes. With this, OverSim is able to simulate all relevant influences of the network in a single simulation event, which is the reason why the *Simple* underlay model causes the lowest overhead of all three models while still providing high accuracy [46]. The overlay layer of OverSim provides various P2P protocols, besides Chord. The application layer includes a DHT implementation and TestApp, which stores (PUT) and retrieves (GET) random overlay keys in order to test the performance of the DHT. Also, statistics about these operations and other parameters are collected in this layer. The Global Observer module provides a global view of the whole overlay network and is used to serve a joining node with the address of a random node that is already existing in the overlay network. Furthermore, it allows the collection of global statistics.

4.3.2. Design of the network simulation

The methodology for the network simulations in OverSim was based on the performance vs. cost evaluation framework (PVC) [48]. It focuses on two aspects of the evaluation of structured P2P systems: how to quantify the cost and performance for highly configurable protocols (e.g., offering different routing table maintenance modes, parallelization of lookups, or favoring neighbors with lower latency), and how to measure the effect of a specific parameter on efficiency. The two metrics used were the average latency of successful routing actions and their error rate. In order to merge these two metrics, "failed routing attempts [were] counted as successful with a latency that equates to the routing timeout" ([49], p. 88). The routing timeout in OverSim is defined as ten seconds by default. In a simulation with PVC, Key-Based Routing (KBR) protocols are compared with different parameter settings. The parameter in this case was the size of the value that is looked up and associated with a key. The version of the framework used for the simulation was OverSim-2012120. It was released on December 6, 2012.

In order to store product information into the DHT, the size of the value field must be increased to hold more than simple information. The GS1 defines several standards for the exchange of product information. For determining how much information must be stored, the Global Data Synchronisation Network (GDSN) definitions from the Foodservice GS1 US Standards Initiative were used (in the revision of May 2012).[4] This standard defines

[3] $Q_3 + 1.5 * IQR$ (see [45]).

[4] http://www.gs1us.org/industries/foodservice/tools-and-resources/gtin-gln-gdsn.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

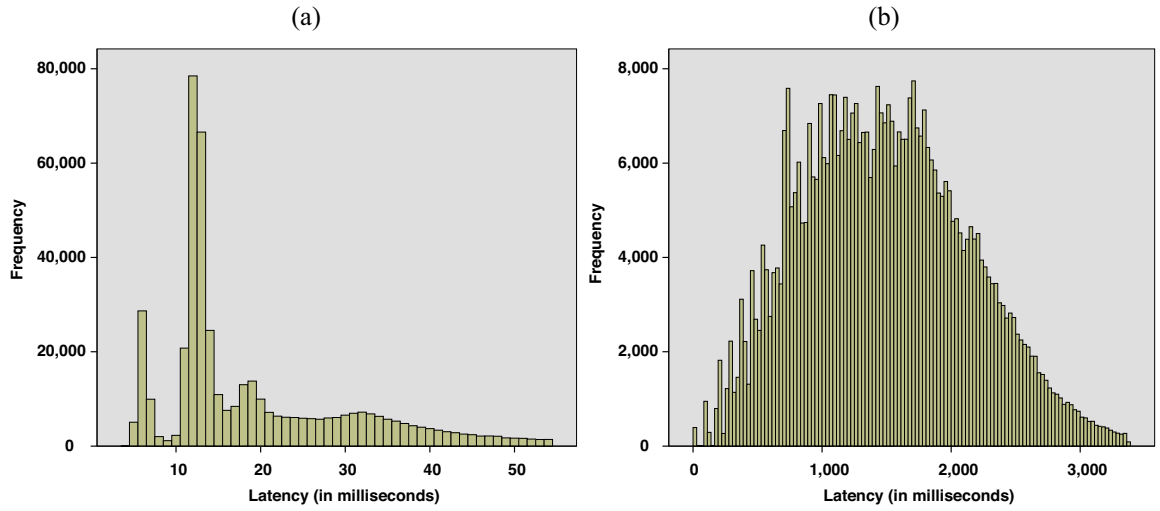


Fig. 5. Main study on EC2. (a) Left: Histogram of lookup latency in the centralized setting (outlier-cleaned sample, $N = 450,000$); data is based on 238 nodes in the US-East region. (b) Right: Histogram of lookup latency in the distributed setting (outlier-cleaned sample, $N = 450,000$). Data is based on 239 nodes; 80 were deployed in the US-East, 80 in the EU-West, and 79 in the Asia-Pacific regions.

attributes such as *storageHandlingTemperatureMaximum* for two implementation phases. These two phases consist of 66 attributes for phase 1, and 41 attributes for phase 2. The business scenarios of GS1 US do not yet involve consumers and focus on the foodservice industry, but the definitions were considered a good estimate upon which to model the data for the refrigerator-lookup use case. As OverSim is written in C++, a program in C++ was developed that models the data fields and calculates the total size needed to represent them. The length of a value field for a key is defined to be 20 bytes by default in OverSim. Space requirements for all attributes in the phase 1 definitions of the GDSN Foodservice add up to 858 bytes. For the underlay, the *SimpleUnderlayModel* was chosen, since it is able to model queuing impacts and offers characteristic Internet latencies. Churn was not considered, meaning that the NoChurn generator was selected in the model, because the nodes (refrigerators) were considered to form a quite reliable infrastructure; households are expected to have their refrigerators running all the time. Every node performed lookups of random node IDs at specific points in time set by a truncated normal distribution in intervals with a mean of 60 s. PUT requests were modeled in the same way, but in order to increase the load and model the push-effect of a grocery producer, for every GET request there were 100 PUT requests. Node IDs were generated with a uniform distribution and a size of 160 bit. For all nodes, the channel Type was set to *simple_dsl*, which models a very conservative ADSL (Asymmetric Digital Subscriber Line) connection with only 1 megabit per second up- and downstream. The init-interval, which defines the time between each creation of a node by the churn generator in the init phase of the simulation, was set to 0.1 s. A measurement time of 500 s was chosen, since for a network with 1000 nodes this interval was sufficient to store up to 1,000,000 keys in the DHT during test runs. The transition time, when effects of fast joining nodes are still present and therefore no statistics are recorded, was set to 100 s.

The number of replicas for each key was set to four, and the time to live (TTL) of a key was set at 90 days, which should reflect that a product in a refrigerator is consumed within this period of time. The number of twelve successors in the Chord DHT was chosen the same as in the experiment on Amazon EC2. The stabilization interval was left unchanged from its standard value (20 s), as was

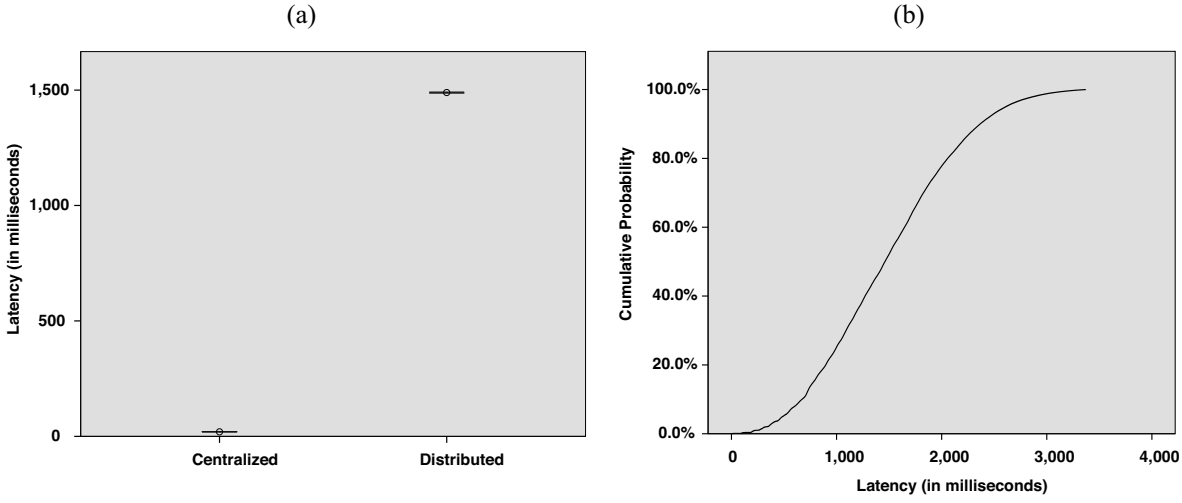


Fig. 6. Main study on EC2. (a) Left: Mean latency and 0.95 confidence intervals. (b) Right: ECDF of latency (distributed scenario). Data basis is an outlier-cleaned sample of the lookups ($N = 450,000$).

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

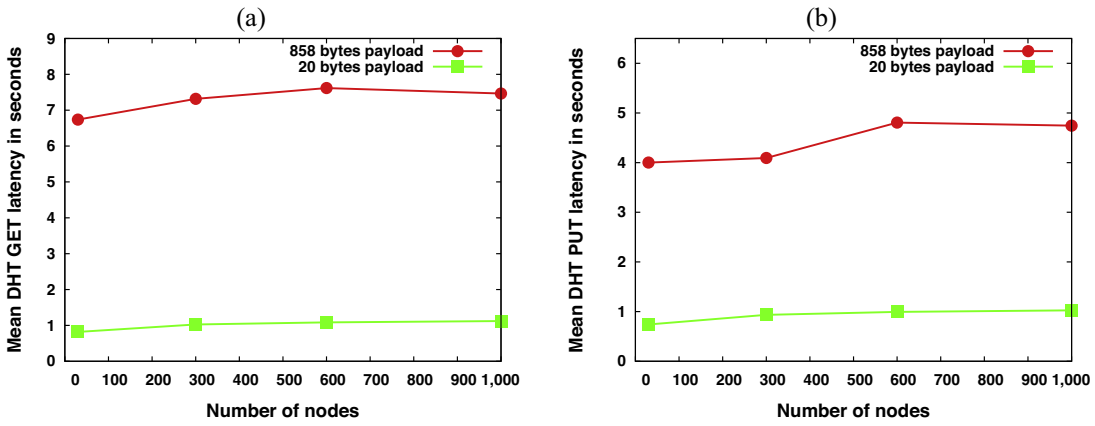


Fig. 7. Chord DHT simulation results. (a) Left: GET latency with different payload lengths (OverSim). (b) Right: PUT latency. The standard payload length is 20 bytes; 858 bytes is the length of GDSN Phase 1 data.

the fix fingers interval (240 s). Neither the extended finger table mechanism nor proximity routing was turned on for the simulation. The routing type was set to *semi-recursive*, since recursive routing is comparable to the routing used in the Octopus lookup procedure. Full recursive routing caused a significantly higher use of system memory during the simulation (Section 5.3). All other settings were left unchanged with their default value. The results of the network simulation in OverSim are shown in the following figures.

Fig. 7a shows the average GET latency of lookups for different payload lengths, with seconds plotted on the *y*-axis and number of nodes in the DHT on the *x*-axis. The simulation generated values for up to 1000 nodes with 858 bytes of payload length (data length of GDSN Phase 1). With 858 bytes of payload, an average latency of 7.62 s was reported for a network of 600 nodes. The minimum average latency measured with this payload was 6.74 s in a network with 30 nodes. For the smaller standard payload length, which is only 20 bytes in OverSim, an average delay of 0.82 s in a network with 30 nodes, and a maximum delay of around 1.2 s with 1000 nodes was reported.

In Fig. 7b, the average delay for the PUT operation is shown, i.e., for storing a value into the DHT. The highest average latency for a payload length of 858 bytes was reported for a network with 600 nodes (4.8 s), even though a larger network of 1000 nodes was also simulated successfully within the discussed constraints by system memory. This payload length of 858 bytes caused a delay of 4.0 s for a network of 30 nodes, which indicates a quite stable scale-out to larger numbers of nodes. With 30 nodes, the average latency for a PUT was 3.6 s, and with 300, still only 3.7 s were reported. The standard value (20 bytes) caused an average latency between 0.7 s with 30 nodes and around 1.1 s with 1000 nodes.

The plot in Fig. 8a reflects the GET success ratio, which indicates how many of the GET requests issued by the nodes could be answered by the DHT. Even if the focus lies on the PUT performance of the DHT, looking at PUT success rates could be misleading, since an entry can get lost. The three graphs show a negative effect of increasing the payload length. For the standard payload of 20 bytes, the success ratio was exactly one for all different network sizes. Increasing the payload size to 858 bytes caused the success ratio to drop by more than 30%. The lowest ratio for that network size was reported as 0.6645 (network size: 600 nodes) and the highest as 0.6855 (network size: 30 nodes).

In order to give an impression of how many entries were stored in the DHT during the simulation, the number of stored values within the measurement phase is shown in Fig. 8b. The two lines represent the measurements within a 1000 node network and different payload lengths of 20 and 858 bytes. Since the DHT already stored entries within the transition phase of 100 s, the lines do not start at zero. Within the simulation, the number of entries grew nearly linearly from 153,674 to 961,641 with a value length

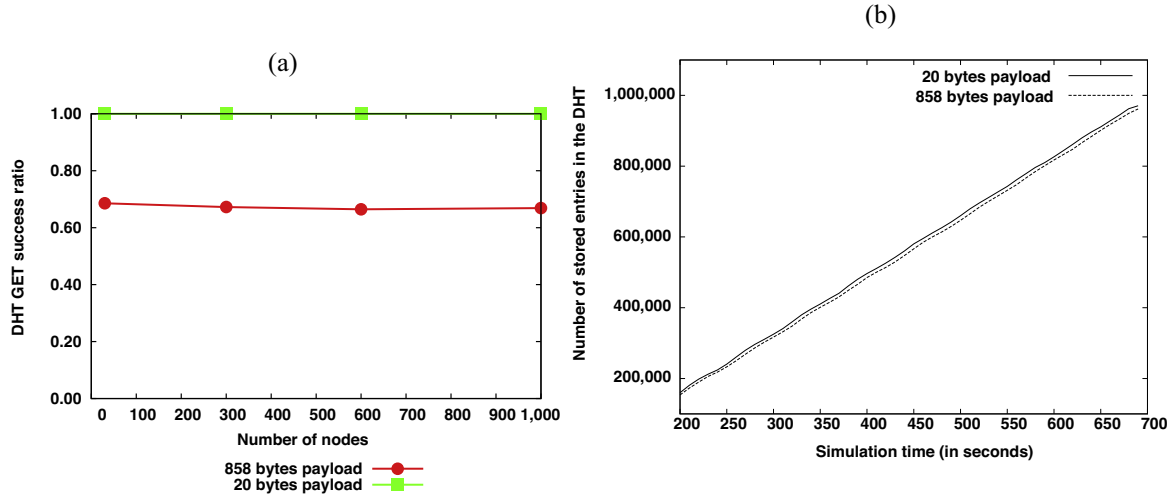


Fig. 8. Chord DHT simulation results. (a) Left: GET success ratio with different payload lengths. (b) Right: Number of stored entries in the DHT in a network with 1000 nodes.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

of 858 bytes, and from 159,876 to 970,700 with a value length of 20 bytes. Values of 858 bytes were stored at a rate of ≈1648.91 records per second simulation time, while the 20 bytes payloads were stored at a rate of ≈1654.74 per second, which is very close.

5. Discussion

5.1. Latency evaluation in the cloud

The centralized and distributed measurements, described in Section 4, represent two different scenarios: the optimal case on the one hand, where all refrigerators (nodes) are in the same town and connected over short distances; on the other hand the worst case, where the nodes are located on different continents. The tolerable waiting time for users, during simple interactive tasks of information retrieval from the Web, is roughly two seconds according to a 2004 study by Nah [50]. In the worst case scenario, the lookup latency meets this requirement in 64% of the cases, and in 97% a lookup is answered within three seconds. Though Nah's study is more than eight years old at the time of this writing, and intuitively, tolerable waiting time is more likely to decrease with technology advancements over time (with the caveat of slower mobile connections), a more recent user study on tradeoffs of latency and anonymity shows that users are willing to wait longer during retrieval tasks from the Web if this latency is the result of a privacy-enhancing technology [42].

Furthermore, the measurements for the centralized deployment show a sufficient overall network performance to be usable inside a real-time system. Though the conditions of the experiment were almost perfect, without any churn or slow Internet connections, they can be seen as a reference value for a real world scenario. A direct comparison with the results of the experiment on PlanetLab by Wang and Borisov [30] where – according to the authors, who kindly supplied it – the same implementation was used, is difficult, since they provide only a plot of a CDF in their paper. Based on this plot, it can be derived that after 2.5 s more than 80% of the lookups were answered by the DHT in their experiment. The authors did not state ambiguous lookups were handled, since the code does not take care of them, and it is possible that two measurements for the lookup of one key are saved in the log files. Another reason why comparability is limited, is the missing outlier-cleaning in the experiment by Wang and Borisov [30]. Moreover, they do not state the rate of failed lookups in their experiment or whether they are expressed in the results. It can be concluded that the performance of the Octopus lookup can be sufficient for a real-time system, if the distance between the nodes is kept relatively short and the connection is stable and fast, such as in a town where all devices are connected via fiber to the home. The average connection speed in Germany was reported as 9 megabit per second (mbps), according to a 2009 study by `SpeedMatters.org`, and it is likely to increase in the future, as it is the goal of the German government to provide broadband access with 50 mbps to 75% of households by 2014 [51]. In a recent study on behalf of the German Federal Network Agency (Bundesnetzagentur), which took place in the second half of 2012 and where the distribution of the sample was close to the population,[5] a mean latency of 23.68 ms for stationary DSL, 15.17 ms for cable, and 44.80 ms for LTE connections is reported [52]. All three values are well below the mean latencies measured between the regions used on Amazon EC2 (see Table A.1 in the Appendix) using the same methodology. Furthermore, only 10.1% of the DSL and 0.3% of the cable users in the study had a broadband connection with 2 mbps or less, while 80.1% of these connections reached at least 50% of the advertised bandwidth. For connections with 2–8 mbps, 70.5% delivered 50% or more of the bandwidth advertised by the provider [52]. Therefore, it is very likely that running Octopus on geographically close nodes with DSL connectivity will cause equal or less latency for lookups than in the distributed experiment.

In order to keep the lookup latencies low, nodes could operate within interconnected Chord rings, as proposed in [53]. This would allow data sharing between different *local* city rings where each ring could keep its autonomy. The interconnection would allow manufacturers to push new product information into city DHTs without owning one (or more) nodes in every city ring, as for privacy reasons our infrastructure would not allow the participation of more than 20% of producer nodes by design.

5.2. Discussion of the network simulation

The results of the network simulation in OverSim show the effect of storing product information inside of a Chord DHT. As the payload size increases, so do the latencies for storing and retrieving values from the DHT. This model does not take into account the additional overhead that results from the Octopus lookup (as discussed in the previous section), so these latencies are expected to increase further in a real implementation. In order to conservatively choose the model settings, a bandwidth of only 1 mbps per node has been chosen. An increase in the bandwidth will most likely result in a decrease of the measured latencies and a higher success ratio.

Using GS1 phase 1 data of up to 858 bytes can be expected to lead to good performance results. However, storing values with 13,105 bytes of information for a key (phase 1 + 2) can be expected to cause too much overhead in this conservative setting of the simulation. An immediate solution could be to use compression in order to decrease the size of the payload, as with 858 bytes the results for different network sizes are relatively stable in terms of the GET success ratio: at least two-thirds of the requests are already effective without parallel requests as used in DNS. Another strategy could be to investigate the usage of the Transmission Control Protocol (TCP) instead of UDP for the DHT. The price for guaranteed delivery of packets and a reliable and ordered transmission offered by TCP would be a higher latency, which however could possibly be mitigated by multiplexing several data exchanges over the same TCP connection.

Regarding the capability of the DHT to store product information, a letter of inquiry in 2012 revealed that the biggest database for product information in Germany, Switzerland and Austria, which is operated by 1WorldSync, a joint venture of GS1 Germany and GS1 US, stores over 1.6 million Global Trade Item Numbers (GTINs), of which 80% are food/non-food consumer products. Therefore, an information system would have to store 1.28 million products in order to store every consumer product in this database. According to the calculation in the appendix (see Table B.4 in the Appendix) that multiplies the number of products with the data storage requirements of the GS1 US Foodservice initiative, the amount of data would fall between ≈1.02 GiB for a payload length of 858 bytes and ≈15.63 GiB for a payload of 13,105 bytes, which reflects the Foodservice implementation phase 1 and 2 combined. Looking at Fig. 8b, which shows the amount of stored entries in the DHT during the simulation, it is plausible to conclude that already with 1000 nodes it would be possible to store 1.28 million values in the DHT when using phase 1 definitions.

The Certificate Authority (CA) should be publicly operated and controlled. This is important for the trust of the end users. An independent agency such as the German Federal Office for Information Security (BSI) could either certify the operator, or even run the whole information system for the CA. Manufacturers

[5] The Bundesnetzagentur states that there were approximately 28 million households with broadband Internet access in Germany in 2012 ([52], p. 28).

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

of refrigerators could then sell higher-priced premium refrigerators with an official certificate, and end-users would profit from an intelligent fridge that takes care of their privacy while providing easy access to product information. The food-producing companies would not need to run their own system, but rather would push the information into the DHT of the refrigerators.

5.3. Contributions and limitations

Our results show that the performance of the Octopus lookup is able to answer queries within 2 s, as long as it is used with nodes that have a stable and fast Internet connection and are geographically close enough. Using the Chord DHT for storing product information is possible, but large data sizes affect the lookup times negatively. Therefore, the data value size should be limited. Here, larger values could be compressed, or split and stored using several keys in the DHT, where one record indicated the key of the next, similar to a linked list. An exact analysis of such a procedure, which would involve a tradeoff between reduced overall load and increased latency as a result of additional lookups, should be studied in future work. Looking at the increased latency due to using the Octopus lookup while storing product information, the overall latency will add up to an approximate value of under ten seconds. This latency is most likely too high for a real-time system, but the refrigerator could query the information as soon as new products are recognized (e.g., when the user is returning from a shopping trip with many products), or within a nightly batch load. This way the device has the information already cached when the user is requesting information and can answer queries immediately.

Concerning other limitations, the evaluation of the infrastructure has focused on the look-up of EPC class-level information for products, similar to the level of granularity of ONS because at this level real-world data was available. Use cases such as product warnings for particular unique serial numbers can already be realized by pushing corresponding critical serial ranges as data values for the product class into the DHT. The main constraint of the network simulation based on OverSim was the limited system memory (see Appendix B for technical details). The simulation was not able to handle more than 1500 nodes with a payload greater than 20 bytes. This can be explained by the architecture of OMNeT++ Network Simulation Framework, which forms the base of OverSim. Increasing the message payload or increasing the amount of messages correlates with the amount of system memory needed, as the messages are modeled exactly as specified by the simulation inside the system. Unfortunately, the paper by Wang and Borisov [30] is not very detailed on the technical aspects of the implementation, and the supplied prototype code for the experiment on EC2 did not allow transfer to simulation code with all necessary details. Therefore, the underlying Chord DHT was chosen for all simulations in order to establish general feasibility and lower latency bounds, whereas exact latency for the full Octopus DHT was measured on EC2.

5.4. Outlook on future research

The OverSim simulation framework offers many options on how further research could be conducted. One aspect would be the full implementation of the Octopus DHT lookup in OverSim, in order to make more accurate estimates about its behavior in very large networks. Another would be the implementation of an EPCIS application layer capable of simulations with a huge number of participants and a sophisticated data model. Since OverSim can integrate real world traffic and is backed by the Karlsruhe Institut für Technologie (KIT), it is a promising framework for conducting further studies on larger and possibly distributed simulation hardware. Experiments on splitting larger data into several records with different keys could be conducted on EC2 and Oversim. Moreover, improving Chord with respect to handling larger data chunks would be a very useful contribution. Hierarchical DHTs should be investigated for their benefits and in particular also for handling a future serial-level lookup of data, as well as for their privacy implications and performance. The Octopus DHT lookup is another promising opportunity for further development. At the time of this writing, there exists no fully functional implementation of it, but the results of this paper indicate that a full prototype would be huge step forward for research based upon anonymous DHT services.

6. Conclusion

This article offers a solution to counter privacy risks of smart home appliances using RFID by presenting a peer-to-peer infrastructure, which provides self-organized data sharing and anonymity of queries between intelligent devices across several homes. The efficiency of this architecture was examined in experiments with several hundred nodes on Amazon EC2. Additionally, corresponding simulations within the OverSim framework were conducted. The proposed architecture is meeting real-time requirements in settings where smart devices are geographically close. It can be used to directly store class-level information of nearly one million different products within a network of a thousand nodes, which is reasonable for a local collaborative infrastructure between smart homes in towns or cities. Future work will focus on modifications for handling larger data items, extended simulations, as well as working toward a real-world implementation.

Acknowledgements

The authors would like to thank Qiyan Wang and Nikita Borisov for providing their implementation of the Octopus DHT lookup.

Appendix A. Evaluation on Amazon EC2

See Tables A.1 and A.2.

Table A.1
Mean latencies (and standard deviations) for the three different regions used in the efficiency evaluation on Amazon EC2. The measurements were conducted with three hosts, each of them in one of the three regions. The ping command was executed every second (30 times in total), targeting a host in one of the other two regions.

From/To	US-East	EU-West	Asia-Pacific
US-East		90.4 ms (0.314 ms)	237.9 ms (18.9 ms)
EU-West	90.3 ms (0.5 ms)		359.3 ms (6.2 ms)
Asia-Pacific	233.6 ms (19.7 ms)	330.392 ms (21.2 ms)	

Table A.2
Specifications of the virtual machines used on Amazon EC2.

Instance type	m1.small
Storage type	instance store (160 GiB)
CPU	1 EC2 Compute Unit
RAM	1.7 GiB
Operating System	Amazon Linux AMI 12.09 (64 bit)

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

Appendix B. Network simulation with OverSim.

See Tables B.3 and B.4.

Table B.3
Specifications of the virtual machine used for OverSim.

CPU	4x Intel Xeon CPU X5460 @ 3.16 GHz
RAM	16 GiB
Operating system	Ubuntu Linux 12.04.2 LTS (64 bit)

Table B.4
Size calculations for GDSN definitions

Number of bits for one char	8
Size of all data types for Phase 1 (in bytes) via sizeof()	858
Size of a char array with 858 values (in bytes)	858
Size of 1.280.000 products stored in this form (in MiB)	1047.36
Size of 1.280.000 products stored in this form (in GiB)	1.023
Size of all data types for Phase 2 (in bytes) via sizeof()	12,250
Size of a char array with 12,250 values (in bytes)	12,250
Size of 1.280.000 products stored in this form (in MiB)	14,953.61
Size of 1.280.000 products stored in this form (in GiB)	14.60
Size of all data types for Phase 1 + 2 (in bytes) via sizeof()	13,105
Size of a char array with 13,105 values (in bytes)	13,105
Size of 1.280.000 products stored in this form (in MiB)	16,000.98
Size of 1.280.000 products stored in this form (in GiB)	15.63

Appendix C. Hardware experiment on computational feasibility

In this appendix, we discuss a small experiment on the computational feasibility of conducting the necessary cryptographic operations on a low-cost chip that can be integrated into smart appliances. The Raspberry Pi is a single-board-computer that has been developed by the Raspberry Pi Foundation in the United Kingdom. It has a Broadcom BCM2835 system on a chip design with ARM processor that is clocked at 700 MHz (factory defaults), 256 MiB of RAM and an 10/100 Ethernet controller (Model B revision 1; http://elinux.org/RaspberryPiBoard). The intention of developing the Raspberry Pi was to teach computer science to pupils in schools. The Model B version costs $ 35 and measures 85.60 mm × 53.98 mm × 17 mm. Therefore it represents a realistic baseline for the hardware of an embedded system inside of a household appliance.

The configuration parameters of the test system are shown in Table C.5. The 800 MHz clock speed of the ARM processor is considered as a conservative setting because the hardware is able to run at clock speeds up to 1 GHz without additional cooling or loosing warranty and most embedded devices are running nowadays at such speeds. Signing and encrypting messages can be considered the most intense operations for the CPU; Wang and Borisov [30] propose AES-128 for encryption in their paper. Table C.6 shows the results of an OpenSSL benchmark conducted on the slightly overclocked Raspberry Pi. It shows that the hardware is capable of processing large chunks of AES encrypted data and is sufficient to sign and verify messages with RSA and a key length of 2048 bit.

Table C.5
Specifications of the Raspberry Pi model B test system.

CPU	ARM1176JZF-S @ 800 MHz
RAM	256 MiB
SD card	16 GB Transcend SDHC Class 10 Memory Card
Network	10/100 wired Ethernet RJ45
Power ratings	700 mA (3.5 W)
Operating system	Arch Linux ARM, Kernel 3.6.11-5-ARCH+

Table C.6
OpenSSL benchmark of the Raspberry Pi model B test system. (The unit for the given measurements is 1000 bytes processed per second. OpenSSL version 1.0.1c (10 May 2012) was used to conduct the benchmark.)

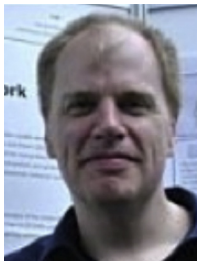
Type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes
AES-128-CBC	16,475.29 k	18,125.96 k	18,551.81 k	18,680.15 k	18,745.71 k
AES-128-IGE	15,877.47 k	17,637.99 k	18,214.87 k	18,281.13 k	17,781.30 k
SHA-1	2924.89 k	8810.20 k	20,562.03 k	30,398.42 k	35,220.14 k
Signing and verifying					
RSA 2048 bits		Sign/s: 15.8			Verify/s: 502.6

References

[1] C. Skrzypczak, The intelligent home of 2010, IEEE Communications Magazine 25 (12) (1987) 81–84.
[2] E. Kaldeli, E.U. Warriach, A. Lazovik, M. Aiello, Coordinating the web of services for a smart home, Transactions on the Web 7 (2) (2013), 10:1–10:40.
[3] A. Juels, RFID security and privacy: a research survey, IEEE Journal on Selected Areas in Communications 24 (2) (2006) 381–394.
[4] LG, LG to Showcase Connected, Easy-to-Control Smart Home Appliances at CES 2013, 2013, http://www.lgnewsroom.com/newsroom/contents/62851.
[5] RFID Journal, Whither the Five-Cent Tag? 2011, February, http://www.rfidjournal.com/article/view/8212.
[6] SpaceCode, RFID SmartFridge Data Sheet, 2012, http://www.spacecode-rfid.com/mobile/documents/RFID-Smart-Fridge_data-sheet.pdf.
[7] Intertek Testing Services, Evolving Smart Technologies Across Home Appliance and Consumer Electronics Markets, 2011, July, http://www.appliancedesign.com/ext/resources/AM/Home/Files/PDFs/EvolvingSmartTechnologies.pdf.
[8] Zpryme, Smart Grid Insights: Smart Appliances, 2010, http://www.smartgridnews.com/artman/uploads/1/2010_Smart_Appliance_Report_Zpryme_Smart_Grid_Insights.pdf.
[9] RFID Journal, Benetton Explains RFID Privacy Flap, 2003, June, http://www.rfidjournal.com/article/articleview/471/1/1/.
[10] B. Fabian, O. Günther, S. Spiekermann, Security analysis of the object name service, in: 1st International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2005), Santorini, (2005), pp. 71–76.
[11] B. Fabian, O. Günther, Security challenges of the EPCglobal Network, Communications of the ACM 52 (7) (2009) 121–125.
[12] B. Fabian, O. Günther, Distributed ONS and its Impact on Privacy, in: IEEE International Conference on Communications (ICC 2007), IEEE, (2007), pp. 1223–1228.
[13] E. Ilie-Zudor, Z. Kemény, F. van Blommestein, L. Monostori, A. van der Meulen, A survey of applications and requirements of unique identification systems and RFID techniques, Computers in Industry 62 (3) (2011) 227–252.
[14] E.W. Schuster, S.J. Allen, D.L. Brock, Global RFID: The Value of the EPCglobal Network for Supply Chain Management, Springer, Berlin - Heidelberg, 2007.
[15] GS1 US, GS1 Standards for Foodservice, 2013, http://www.gs1us.org/industries/foodservice/relevant-standards.
[16] GS1 US, An Introduction to the Global Trade Item Number (GTIN), 2013, http://www.gs1us.org/gtin.
[17] GS1, EPC Information Services (EPCIS) Version 1.0.1 Specification, 2007, http://www.gs1.org/gsmp/kc/epcglobal/epcis/epcis_1_0_1-standard-20070921.pdf.
[18] S. Evdokimov, B. Fabian, S. Kunz, N. Schoenemann, Comparison of discovery service architectures for the Internet of things, in: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2010), Newport Beach, 2010.
[19] Y.-S. Kang, Y.-H. Lee, Development of generic RFID traceability services, Computers in Industry 64 (5) (2013) 609–623.
[20] J. Shi, Y. Li, W. He, D. Sim, SecTTS: a secure track and trace system for RFID-enabled supply chains, Computers in Industry 63 (6) (2012) 574–585.
[21] B. Fabian, Implementing secure P2P-ONS, in: IEEE International Conference on Communications (ICC 09), IEEE, 2009.
[22] N. Schönemann, K. Fischbach, D. Schoder, P2P architecture for ubiquitous supply chains, in: 17th European Conference on Information Systems (ECIS 2009), 2009.
[23] M. Dias De Amorim, S. Fdida, N. Mitton, L. Schmidt, D. Simplot Ryl, Distributed Planetary Object Name Service: Issues and Design Principles, Research Report RR-7042, INRIA, 2009, http://hal.inria.fr/inria-00419496/PDF/RR-7042.pdf.
[24] S. Shrestha, D.S. Kim, S. Lee, J.S. Park, A peer-to-peer RFID resolution framework for supply chain network, in: Second International Conference on Future Networks, 2010.
[25] L. Schmidt, R. Dagher, R. Quilez, N. Mitton, D. Simplot Ryl, DHT-based distributed ALE engine in RFID Middleware, Research Report RR-7316, INRIA, 2010, http://hal.inria.fr/inria-00491795/PDF/RR-7316.pdf.
[26] D.-G. Xu, L.-H. Qin, J.-H. Park, J.-L. Zhou, ODSA: Chord-based object discovery service architecture for the Internet of things, Wireless Personal Communications 73 (4) (2013) 1455–1476.
[27] P. Manzanares-Lopez, J.P. Mu noz-Gea, J. Malgosa-Sanahuja, J.C. Sanchez-Aarnoutse, An efficient distributed discovery service for EPCglobal Network in nested package scenarios, Journal of Network and Computer Applications 34 (3) (2011) 925–937.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001

[28] J.P. Munoz-Gea, J. Malgosa-Sanahuja, P. Manzanares-Lopez, J.C. Sanchez-Aarnoutse, Implementation of traceability using a distributed RFID-based mechanism, Computers in Industry 61 (5) (2010) 480–496.
[29] B. Fabian, T. Ermakova, C. Müller, SHARDIS: a privacy-enhanced discovery service for RFID-based product information, IEEE Transactions on Industrial Informatics 8 (3) (2012) 707–718.
[30] Q. Wang, N. Borisov, Octopus: a secure and anonymous DHT lookup, in: 32nd International Conference on Distributed Computing Systems, 2012.
[31] I. Stoica, R. Morris, D. Karger, M. Kaashock, H. Balakrishman, Chord: a scalable peer-to-peer lookup protocol for Internet applications, in: ACM SIGCOMM 2001, 2001.
[32] A. Pfitzmann, M. Hansen, A Terminology for Talking about Privacy by Data Minimization, 2010, http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.
[33] R. Dingledine, N. Mathewson, P. Syverson, Tor: the second-generation onion router, in: 13th USENIX Security Symposium, 2004.
[34] R. Housley, W. Polk, W. Ford, D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Request for Comments (RFC): 3280, 2002, http://www.ietf.org/rfc/rfc3280.txt.
[35] Y. Rekik, E. Sahin, Y. Dallery, Analysis of the impact of the RFID technology on reducing product misplacement errors at retail stores, International Journal of Production Economics 112 (1) (2008) 264–278.
[36] A. Bayraktar, E. Yılmaz, Ş. Erdem, Using RFID technology for simplification of retail processes, in: C. Turcu (Ed.), Designing and Deploying RFID Applications, InTech, 2011 (chapter 6).
[37] J.R. Douceur, The Sybil attack, in: First International Workshop on Peer-to-Peer Systems (IPTPS 2001), vol. 2429 of LNCS, Springer Berlin-Heidelberg, 2002, pp. 251–260.
[38] C. Díaz, S. Seys, J. Claessens, B. Preneel, Towards measuring anonymity, in: R. Dingledine, P. Syverson (Eds.), Privacy Enhancing Technologies Workshop (PET 2002), vol. 2482 of LNCS, Springer, 2003, pp. 54–68.
[39] A. Kapadia, N. Triandopoulos, Halo: high-assurance locate for distributed hash tables, in: Network and Distributed System Security Symposium (NDSS 08), 2008.
[40] A. Panchenko, S. Richter, A. Rache, NISAN: network information service for anonymization networks, in: 16th ACM Conference on Computer and Communications Security, ACM, (2009), pp. 141–150.
[41] J. McLachlan, A. Tran, N. Hopper, Y. Kim, Scalable onion routing with torsk, in: 16th ACM Conference on Computer and Communications Security, ACM, (2009), pp. 590–599.
[42] F. Brecht, B. Fabian, S. Kunz, S. Müller, Are you willing to wait longer for Internet privacy? in: 19th European Conference on Information Systems (ECIS 2011), 2011.
[43] S. Müller, F. Brecht, B. Fabian, S. Kunz, D. Kunze, Distributed performance measurement and usability assessment of the tor anonymization network, Future Internet 4 (2) (2012) 488–513., http://www.mdpi.com/1999-5903/4/2/488.
[44] G. Wang, T.E. Ng, The impact of virtualization on network performance of Amazon EC2 data center, in: IEEE INFOCOM 2010, IEEE, (2010), pp. 1–9.
[45] J. Janssen, W. Laatz, Statistische Datenanalyse mit SPSS für Windows, 6th ed., Springer-Verlag, Berlin, Heidelberg, New York, 2007.
[46] I. Baumgart, B. Heep, S. Krause, OverSim: a flexible overlay network simulation framework, in: IEEE Global Internet Symposium, 2007, 79–84.
[47] OMNeT++, Website, 2012, http://www.omnetpp.org/.
[48] J. Li, J. Stribling, R. Morris, M.F. Kaashoek, T.M. Gil, A performance vs. cost framework for evaluating DHT design tradeoffs under churn, in: IEEE INFOCOM 2005, vol. 1, IEEE, (2005), pp. 225–236.
[49] I. Baumgart, B. Heep, Fast but economical: a simulative comparison of structured peer-to-peer systems, in: 8th EURO-NGI Conference on Next Generation Internet (NGI), IEEE, (2012), pp. 87–94.
[50] F.F.-H. Nah, A study on tolerable waiting time: how long are web users willing to wait? Behaviour & Information Technology 23 (3) (2004) 153–163.
[51] D.M. West, The Next Wave: Using Digital Technology to Further Social and Political Innovation, Brookings Institution Press, Washington, D.C., 2012.
[52] K. Lukas, A. Marx, B.O. Schöttler, C. Sudhues, Dienstequalität von Breitbandzugängen, 2013, April, http://www.bundesnetzagentur.de/.
[53] Z.L. Kis, R. Szabó, Interconnected Chord-rings, Network Protocols and Algorithms 2 (2) (2010) 132–146.



Benjamin Fabian is currently visiting professor and acting director at the Institute of Information Systems, Humboldt University Berlin. He holds a Diploma degree in Mathematics from the Free University of Berlin, and a Ph.D. with Habilitation in Information Systems from Humboldt University Berlin. His research interests include IT security and privacy, cloud and peer-to-peer computing, Internet science, and complex networks.

Tobias Feldhaus holds a Master's degree in Information Systems from Humboldt University Berlin, and a Bachelor's degree in Information Systems from the University of Mannheim. He currently works as a data engineer at Wooga, one of the most popular developers of mobile games in the world. Over 50 million people play Wooga's games every month across multiple platforms. His work and research interests include databases, networking, security and technologies for handling Big Data.

Please cite this article in press as: B. Fabian, T. Feldhaus, Privacy-preserving data infrastructure for smart home appliances based on the Octopus DHT, Comput. Industry (2014), http://dx.doi.org/10.1016/j.compind.2014.07.001