Journal of Universal Computer Science, vol. 18, no. 9 (2012), 1218-1235
submitted: 1/12/11, accepted: 28/4/12, appeared: 1/5/12 © *J.UCS*

Security-enhanced Search Engine Design in Internet of Things

Xiaojun Qian
(Nanjing Normal University, Nanjing, China
qxj6288991@sina.com)

Xiaoping Che
(Corresponding author)
(Evry University, Paris, France
buptchexp@gmail.com)

Abstract: This paper elaborates the challenges in searching imposed by the burgeoning field of Internet of Things (IoT). Firstly it overviews the evolution of the new field to its predecessors: searching in the mobile computing, ubiquitous computing and information retrieve. Then, it identifies five research thrusts: architecture design, search locality, real-time, scalability and divulging information. It also sketches several presumptive IoT scenarios, and uses them to identify key capabilities missing in today's systems. On top of these challenging issues, we report our undertaking work — a security-enhanced search engine for Internet of Things based on Elliptic Curve Cryptography (ECC) security protocol. We also report our preliminary experimental results.

Key Words: Internet of Things, Mobile Computing, Security, Elliptic Curve Cryptography
Category: H.3.3, H.3.1, J.6

1 Introduction

The Internet of Things (IoT) refers to a networked interconnection of daily objects, thus requires the objects not only for beings to interact, but also for cooperating with each other at anytime or in anyplace [Gershenfeld et al., 2004] [Margery, 2010]. It opens the door of Internet to the physical world such that objects can be managed remotely and act as physical access points to Internet services [Woelffle et al., 2010]. IoT transforms the manner of daily activities by real-time tracking physical objects. Correspondingly, it opens up massive opportunities for economy and individuals, accompanying immense technical challenges and risks.

IoT is established on the basis of proliferation of wireless sensor network, MobiComp (*Mobile Computing*), UbiComp (*Ubiquitous Computing*) and information technologies [Chui et al., 2010]. Thanks to their diminishing size, declining price and falling energy consumption, sensors are being increasingly integrated into everyday objects. Thus, IoT is applicable in a wide spectrum of fields. To get a heightened awareness of real-time events, it deploys sensors in infrastructures [Kortuem et al., 2009]. For achieving an enhanced situational awareness, it employs Radio Frequency IDentification (RFID) to capture object contexts (e.g., location) [Welbourne et al., 2009]. For

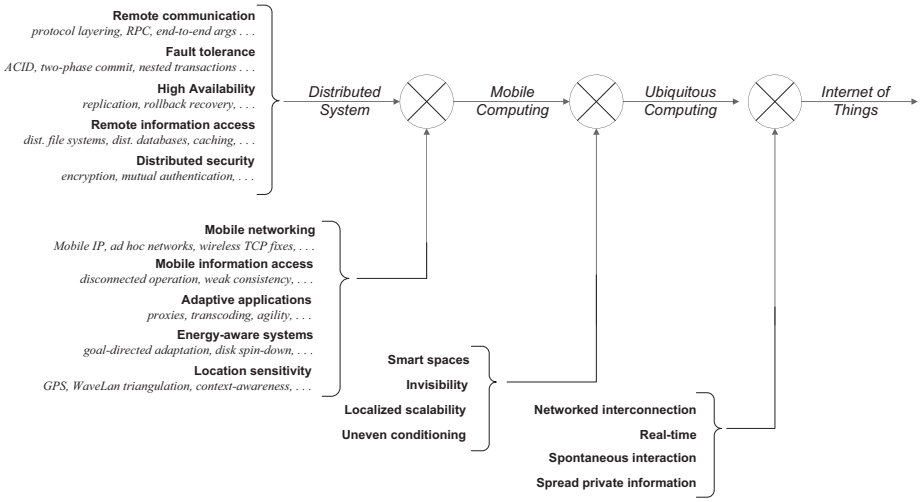

Figure 1: The evolution of Internet of Things from distributed computing, mobile computing and ubiquitous computing

guaranteeing safe driving and green travel, it uses motes to track transportation systems [Puliafito et al., 2010]. For getting user preferences, IoT takes advantage of a recommendation service in recommender systems (i.e., [Aberer et al., 2006] a kind of virtual sensors). As the trend goes, we foresee that IoT eventually links the majority of objects into the virtual space and allows objects to interact in the same place.

Readings from substantive sensors for objects are numerous and extremely dynamics. We will have hundreds of billions of RFID-tagged objects [Das and Harrop, 2010] by 2015, resulting in billions of sensor readings every moment. This indicates we have a massively searching space. Given that billions of sensors – deployed or embedded into infrastructures and objects – are keeping acquiring readings, IoT works in a much larger information space than that of the Internet. Moreover, sensor readings are highly dynamics. They vary with environment and objects. Suppose IoT is acquiring a visitor's location by Global Position Systems (GPS). The raw readings about his / her location keep evolving such that every reading of the location has a short life span. In contrast, most web pages in the Internet are static. They are changed at time intervals which is slower than the update frequency of sensor readings. Another distinction between IoT and Internet is that user behaviors, showing that they are concerned with physical objects in their vicinity, rather than information (e.g., web pages) distributed in faraway places. This is because users usually would like to manually operate physical objects to achieve their goals. Correspondingly, the searching technique suffers from a series of issues in IoT. Besides, these issues will lead the sensors potentially reveal crucial information.

Thus, we find that it is advisable to visit the searching technique in IoT and also to protect the privacy of users. We aim at articulating the challenging issues in this area imposed by IoT and answering how we rapidly design search engines for IoT. We begin by examining predecessors of the searching technique in MobiComp, UbiComp and Information Retrieve(IR). Then, we propose a series of measurement dimensions to decompose and analyze the state-of-the-art search engines to identify what missing in satisfying requirements of IoT's searching. Finally, we propose our development of searching engine and security framework. In summary, the main contributions of this paper are two-fold.

– We thoroughly investigate the searching problem in IoT in a following manner. Firstly, we look back to the evolution of the searching from MobiComp, UbiComp to IoT. Then, we characterize the IoT's searching by five features: architecture design, search locality, real-time, scalability and divulging information. Finally, we point out a couple of future research directions, and introduce our ongoing project *ISE* (IoT Search Engine) that is designed based on IoT requirements.

– By incorporating the security control and the identified challenging issues, we design an security-enhanced IoT search engine (ISE) based on the Identification-based cryptosystem using the Elliptic Curve Cryptography (ECC) technique. Our preliminary results are provided in this paper.

The rest of this paper is organized as follows. Section 2 introduces the background and overviews the evolution of IoT. Section 3 identifies challenging issues of the searching technique in IoT. Section 4 proposes a series of measurement dimensions, by which it overviews existing efforts from MobiComp, UbiComp and IR. Then, it puts forward several considerations how to develop IoT's search engines. Finally, Section 5 introduce our ongoing project ISE and IBC-based security framework. Section 6 concludes the paper.

2 Background

To clearly figure out our research background, we give a brief introduction to Internet of Things (IoT). Consider that IoT evolves from MobiComp and UbiComp; we also introduce these two computing paradigms. We discuss the evolution of IoT and identify its distinct characteristics that have significant impact on the design of the relevant search engines.

2.1 Evolution of computing paradigm

IoT represents a major evolutionary step in a line of work dating back to the mid-1980s. Two distinct earlier steps in this evolution are MobiComp and UbiComp. Some of the technical problems in IoT correspond to problems already exist and studied in

the evolution. Meanwhile, IoT brings new problems that have to obvious mapping to problems studied earlier. We try to figure out the relationship in this evolution and to develop a taxonomy of issues characterizing each phase of the evolution.

Figure 1 illustrates the evolution of computer systems from MobiComp, UbiComp to IoT. MobiComp came into being when mobile clients appeared in 1990s such that building applications required the support of mobile networking [Bhagwat et al., 1996] [Royer and Toh, 1999], mobile information access [Tait and Duchamp, 1992] and application adaptation [Fox et al., 1996]. Meanwhile, applications were capable of tracking users with a wireless localization function. Consider that mobile devices usually have limited life span thus applications were designed to minimize energy consumption. MobiComp imposed several distinct problems in developing mobile applications. These problems consist of unpredictable variation in network quality, lowered trust and robustness of mobile applications, limited computation and communication capability from handheld and heterogeneous devices, and concern for energy consumption [Satyanarayanan, 1996] [Zhang et al., 2011c].

The flourish of wireless technologies, mobile and embedded devices has fostered an increasing interest in a new computing paradigm — ubiquitous computing. This paradigm aims at building an intelligent space such that users share services provided by ubiquitous environment without awareness of underlying technologies. UbiComp — evolving from MobiComp — incorporates four distinct research thrusts into it — smart spaces, invisibility, localized scalability and uneven conditioning [Weiser, 1999] [Satyanarayanan, 2001]. Smart spaces, e.g., meeting rooms, cars and hospitals, are well-defined areas that are saturated with various tiny sensors and handheld devices such that they smartly serve users. Invisibility refers to UbiComp meets user requirements with minimal user involvement. Localized scalability denotes that the intensity of interactions between a user's personal computing space and his/her surroundings increases as smart spaces grow in sophistication. [Zhang et al., 2011a] Uneven conditioning refers to nontechnical factors such as organizational structure, economics and business models that hides the applicability of UbiComp.

Research in UbiComp in the last twenty years has grown dramatically. A variety of schemes and experimental demos have been proposed regarding effectively using smart spaces, hiding enabling technologies and masking uneven conditioning. Recently, as increasingly number of sensors and handheld devices are being connected to the Internet, a new computing paradigm, i.e., Internet of Things, appears. IoT extends the UbiComp into the physical world, and hence provides new perspective on the characteristics [Chen, 2012].

2.2 Characteristics of Internet of Things

IoT is a computing paradigm shift from UbiComp and has achieved widespread success in object tracking, asset management and intelligent transportation system. It is characterized by four features: networked interconnection, real-time, spontaneous interaction

and spread privacy information. shown in Figure 1.

– **Networked interconnection**. Daily objects are assigned with unique identification by using RFID and sensors in IoT, and they are linked into the Internet. Thus, IoT mapped all physical objects into the Internet.

– **Real-time**. Sensor readings keep evolving, leading to the acquired information rapidly changing. Moreover, these readings may vary with users and their movement. Nevertheless, conventional search engines fail due to the reason that they implicitly assume that web content changes slowly, so that it is sufficient to index at a frequency of days, weeks, months even years. As a matter of fact, IoT requires a real-time searching technique.

– **Spontaneous interaction**. IoT is a loosely-coupled system, where devices with limited computing and communication capability may spontaneously interact with each other. Therefore, IoT are required to handle each interaction in an efficient manner such that the entire system is scalable and real-time, irrespectively of how many objects and their interaction.

– **Spread private information**. All of the subjects are connected in the IoT, which can meet the individual need, it may expose privacy information to the public at the same time. IoT presents significant challenge in terms of who can see what with which credentials. Thus, it requested the guaranty of data, information security and authentication of the users.

3 Challenging Issues of the Searching in IoT

Traditional search engines fall short of in IoT due to the challenging issues imposed by the characteristics of IoT [Zhang et al., 2011b]. These issues hamper the information sharing and processing. They also show directions that rapidly building IoT search systems should take into consideration [Jin et al., 2011]. In this section, we identify these challenging issues.

– **The prominent challenging issue is the architecture design of search engines.** The goal of searching physical objects distinguishes that of commercial search engines is that users are willing to operate objects locally or remotely. Thus, searching in IoT requires a methodology of architecture design of search engines. However, designing an appropriate search engine for IoT is non-trivial. This is because the design process requires a new technique of search engines ranging from crawling, indexing, storing and querying.

– **Another challenging issue is search locality**. Users in the World Wide Web or Internet would like to get information of the query items in the search process. They obtain knowledge by assimilating the information retrieved in the search results.

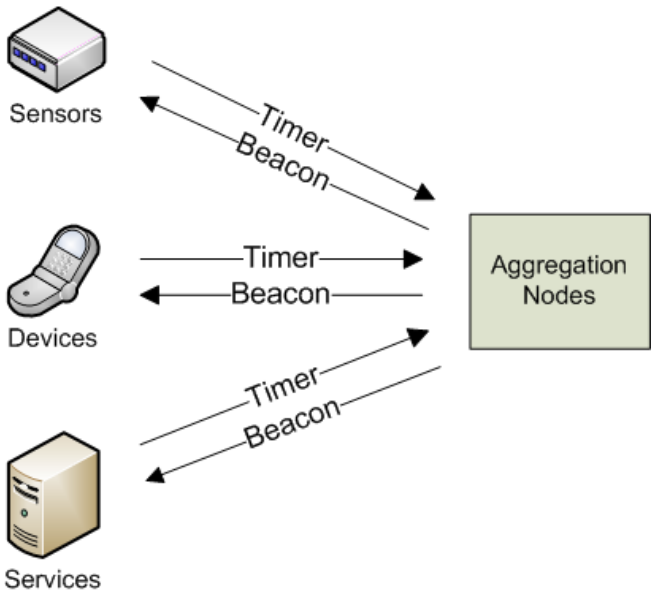

Figure 2: Communication paradigms of the searching in IoT, consisting of timer and beacon schemes that assist aggregation nodes in communicating with sensors, devices and services.

Nevertheless, in most cases users in IoT are likely to operate or control physical objects. For instance, Jim in his car queries where his key to the home is. He aims at using his key to open the door. Existing searching techniques are based on remote information sharing such that they fail to smoothly support local search, specifically, local search of physical objects.

– **The third crucial challenging issue is scalability**. In IoT, environment is saturated with a huge number of sensors. Take RFID-tagged objects for example. According to [Das and Harrop, 2010], there will be hundreds of billions of RFID-tagged objects by 2015. Sensors and RFID keep capturing objects and generating an extremely large number of readings. Moreover, these readings usually have limited timeliness. For instance, a reading of a RFID passive tag may be expired within 1 or 2 seconds. Suppose we adopt the indexing technique of traditional search engines, the size of the index in IoT will increase significantly, which leads to the clumsy performance. Actually, most sensor readings indexed are out of date and thus are less insignificant to response to queries than current readings.

– **The fourth challenging issue is real-time**. One of the crucial function of IoT is real-time tracking. Notice that sensor readings from a tremendous number of sensors are easily to be expired. Therefore, IoT needs to guarantee the real-time of IoT

system in the presence of local search and non-local search. The real-time search lies on the communication paradigms. Only real-time communication paradigms offer the possibility of real-time index and search [Wu et al., 2011]. In general, there are two communication paradigms available in information aggregation: ***timer*** or ***beacon*** schemes. In the former scheme, sensor nodes report their readings to aggregation nodes when they detect an event. Once aggregation nodes do not receive any ***alive*** message from a specific sensor within a period, the sensor may move or be in failure. Timer scheme guarantees the up-to-date information of every sensor, following with much energy consumption and a large amount of redundant readings to aggregation nodes. In contrast, in the latter scheme, aggregation nodes periodically broadcast message to every individual sensors. Once they cannot get reply from an individual sensor, they can easily detect the failure or movement of this sensor. The beacon scheme is energy-efficient, but it cannot always get the latest readings. Figure 2 illustrates the two communication paradigms above. In order to keep up-to-date information of physical objects, IoT has to consider the combination of these communication paradigms. Due to the randomness of user movement and the quick out-of-date characteristics of a large number of sensor readings, it is difficult to find an ideal integrating way.

– **The last challenging issue is divulging information**. An object of the internet of things may contain personal and critical information. The owner of the object will not accept to exchange or share these data with any person. However in IoT, a malicious reader is able to collect data and use it without the authorization of the user [Coetzee and Eksteen, 2011]. For example: Your identity card can be an object of the IoT. As a result, an attacker can read your personal information such as name, address, birthday, age, etc. Besides, the traceability can be another problem. If an object O_1 is used to exchange encrypted data and it is identified by an address ID_1 to send and receive messages, any reader will be able to detect this address then it can threaten the personal privacy. Taking your transport card ID for example, it can be used by a group of malicious readers. They can have information about your last plan in the weekend by consulting the visited station and also your time of arriving to home T_h (Time of your arriving to the station T_a plus an estimation,$T_h = T_a + \varepsilon$).

4 Prototypes: Case study

Research community in MobiComp, UbiComp and IR has put forward a couple of systems that partially support the search of physical objects. Note that these systems may not call to mind the searching characteristics in IoT, but their underlying mechanisms would also apply to search real objects in IoT. To distinguish existing schemes, we propose a series of measurement dimensions. In this section, we introduce the dimensions. Then, we overview existing efforts in detail. Finally, we propose several considerations for developing IoT's search engines.

4.1 Measurement dimensions

Searching in IoT involves a number of different dimensions, which forms the design space of developing corresponding search engines. In the context of our discussion, we account for eight crucial design dimensions. In the rest of this section, we will refer to these dimensions to characterize existing efforts that would be partially available in IoT.

Table 1: Measurement dimensions of search engines in IoT

Dimensions	Explanation
architecture	the structure of a search engine, i.e. *centralized* or *distributed*
aggregation type	the type of aggregating sensor readings, *beacon* or *timer* or *hybrid*
index type	the method to index sensor readings
prototype	whether a prototype exists
query type	the type of query, either *ad hoc* or *continuous* query
query mode	the supported manner of the search engine which can be interpreted as *keyword-based*, or *semantics-based*
query scope	the scope of the query, either *local* or *global*
security support	whether the search engines include the security protection

Table 1 illustrates the measurement dimensions that are extracted to evaluate existing efforts. These dimensions consist of architecture, aggregation type, index type, prototype, query type, query mode, query scope, and security support.

4.2 Investigation

We investigate existing efforts based on the measurement dimensions that are listed in Table 1. We select Snoogle/Microsearch [Wang et al., 2010, Tan et al., 2010], MAX [Yap et al., 2008], OCH (Object Calling Home) [Frank et al., 2008], GSN (Global Sensor Networks) [Aberer et al., 2007], RTS (Real-time Search) [Corley, 2010] and Dyser [R*ö*mer et al., 2010] from MobiComp, UbiComp and IR as baselines. We firstly review the baselines and then discuss their superiorities.

Table 2 illustrates our insights to existing efforts from the selected measurement dimensions.

- This table shows that most search systems in MobiComp, UbiComp and IR are designed by following hierarchical structure in a centralized or distributed manner. Note that RTS is a file-system-based search engine that is not designed in the same way as that of the remaining systems.

- It also shows that current search systems support *local* or *global* search. Note that local refers to that search systems only work locally, i.e., they are used to specific areas or regions, e.g., parking, rooms and buildings. *Global* denotes that the search

Table 2: Overview of existing efforts that would be available for searching in IoT

Dimensions	Snoogle Microsearch	MAX	OCH	GSN	RTS	Dyser
architecture	distributed, two-tier	centralized, three-tier	distributed, two-tier	centralized, container-based	file system	centralized, two-tier
aggregation type	hybrid	beacon	timer	beacon	n/a	hybrid
index type	inverted	no	no	MySQL-style	inverted	inverted
prototype	AP-based, smart buildings	mica-based, smart rooms	phone-based, GPS-based	mica-based, framework	no	no
query type	ad hoc	ad hoc	continuous	ad hoc	ad hoc	ad hoc
query mode	keyword	keyword	keyword	keyword	keyword	keyword
query scope	local	local	local	global	global	global
security support	yes	no	no	no	no	yes

scope of search engines crosses all the working space. However, this kind of systems does not account for search locality or are not tailored for local search.

– The table indicates that all search engines only support keyword search. They do not consider the object contexts and dynamic changes of environment, e.g., user location, device switching and limited capability of communication and computation.

– Besides, the current search engines rarely support the security issues, even some engines support the security, they do not take into account the privacy problems.

– Finally, existing search engines offer various support to the searching in IoT, but do not systematically assist programmers in developing a specialized search engine for IoT. By following the experience from current work, we partially meet the requirements in search of IoT.

5 ISE: IoT Search Engine

In Section 2, Section 3 and Section 4, we have reviewed the state-of-the-art of search engines mainly in MobiComp, UbiComp and IR. A large amount of work has been conducted in the literature, though there are still many open issues. On the basis of discussion of existing efforts, we point out some directions that may be a potential way of developing search engines in IoT.

In this section, we introduce *ISE* – an IoT Search Engine, our ongoing project of a security-enhanced search engine for IoT. To the best of our knowledge, *ISE* is the first work on searching in IoT. Note that RFID is one of most important enabling technique to achieve IoT; we conduct our project only limited to RFID-based scenarios from the time being. We are using RFID EPCIS technique (*h*ttp://wiki.aspire.ow2.org/

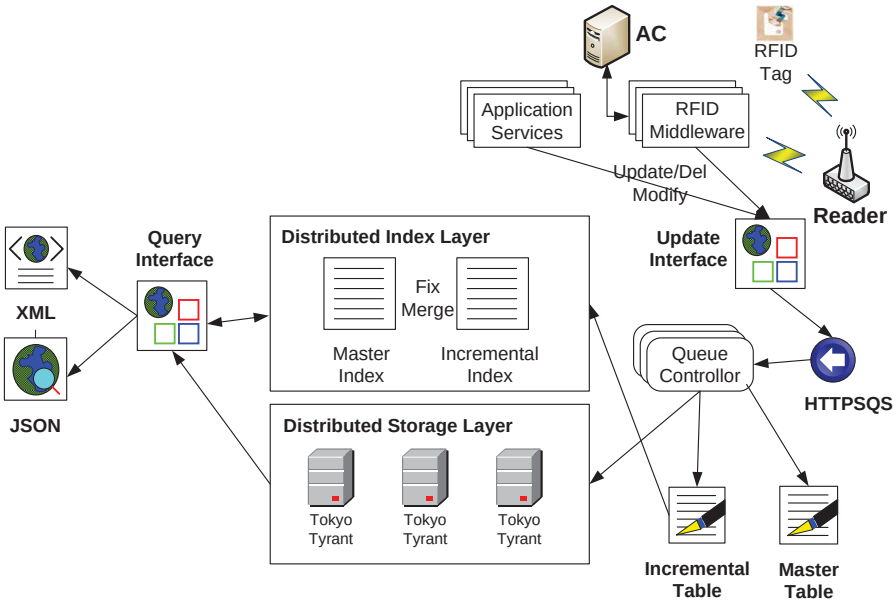

Figure 3: ISE Architecture

xwiki/bin/view/Main.Documentation/EpcisRepository) to build our RFID middleware that acquires readings and disseminates query requests. Later, we will gradually complete support to sensor-based scenarios.

5.1 System Overview

Figure 3 illustrates the architecture of ISE, consisting of index module (extracted from Sphinx project, *http://sphinxsearch.com/*), update module ,query module and security module. The index module provides two kinds of functions — index and storage — as distributed index layer and distributed storage layer. The update module receives data DML operations (i.e., update, delete and modify) from applications, services or RFID EPCIS middleware. Whereas the query module handles requests and return query results in XML, HTML or JSON formats. What sort of format is taken relies on the query contexts, e.g., screen sizes and communication capabilities of handheld devices. The IBC-based security module provides the protection of crucial information and the authentication of the readers. During the process of encryption and decryption, ECC-based algorithm has been used to generate public key and secret keys.

5.2 The design of ISE for Internet of Things

We have mentioned several challenging issues in searching in IoT. When designing and implementing ISE, we have accounted for these issues.

Firstly, we have adopted two measures to solve the problem of huge searching space. One is that we employ RFID EPCIS middleware to deal with tag readings and queries. Each middleware takes full responsibility for a small area, e.g., 40 square meters. The other is that we have taken distributed index and storage techniques. Thus, the searching space is significantly decreased.

Secondly, we have used three measures to achieve real-time searching. We enrich the function of RFID EPCIS middleware by [Jeffery et al., 2006] so that it is capable of filtering redundant readings and removing abnormal readings. Then, we thoroughly investigate most IoT scenarios and find that only partially sensor readings keep evolving with evidently change. Therefore, we design two kinds of table – master and incremental tables. We update master table at a long intervals (e.g., 20 minutes in our experiments), while incremental table with a short interval (e.g., 3 minutes) to keep sensor readings up-to-date. We also employ a lightweight indexer and a database to index and store data as quickly as possible. This part is described as follow.

Thirdly, we have utilized several policies to ensure the scalability of searching. We select Nginx as the web server whose efficiency is 10 times that of the MySQL. Furthermore, Nginx allows more concurrent requests than MySQL, i.e., from 300 to more than 10, 000. At the same time, we use Tokyo Cabinet (*http://fallabs.com/tokyocabinet/*) as data storage servers, and Tokyo Tyrant to access it. Extensive experiments show that Tokyo Cabinet substantially outperforms MySQL concerning building index time, query response and workload (More information in *http://www.MySQLperformanceblog.com*).

Finally, we designed and implemented a security framework for ISE to protect the privacy, especially the crucial private information. We used IBC-based cryptosystem for the framework and the Elliptic Curve Cryptography for the encryption and decryption. Some previous work illustrated the possibility of using IBC and ECC in mobile environment [Bradai and Afifi, 2011]. Compared to RSA, the prevalent public-key scheme of the Internet nowadays, elliptic curve cryptography offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings and is thus better suited for small devices such as RFIDs. The comparison between RSA and ECC on small devices has been done by [Creado et al., 2009], and also the power consumption has been studied by [Nils Gura, 2004]. These works proved the possibility to apply ECC algorithms on the devices of IoT to protect the privacy issues. We will use the following subsection to describe the details of the security framework.

5.2.1 Security design of ISE

Under specific conditions, the privacy of identity of subjects, the location messages and any other related information should be guaranteed during the searching. Accord-

ing to this point of view, we proposed an IBC-based cryptosystem which uses Elliptic Curve Cryptography (ECC) [Malhotra et al., 2007]. The Identity-Based cryptosystem was first proposed in 1984 [A.Shamir, 1984]. The IBC is built on top of the well-known public key cryptography, but the user can use their identity (name, address, ID) as his public key instead of using the public keys generated by the system. However, all the existing IBC security frameworks are using Private Key Generator (PKG) to assign the private/public keys to the devices, which could cause an attack [Zhang and Wang, 2008]. Therefore, in order to provide a strong security framework, we apply Elliptic Curve Cryptography for the enhancement. The Elliptic Curve Cryptography (ECC) is a technique to public key cryptosystem based on the algebraic structure of elliptic curves over finite fields [Koblitz, 1987] [Miller, 1986].

5.2.1.1 Model

Bearing the security requirement and the characteristic of IoT in mind, we present a network security model, illustrated in Figure 4. This model consists of three major components: RFID Tags, RFID Readers and Authentication Center (AC). RFID Tags represent

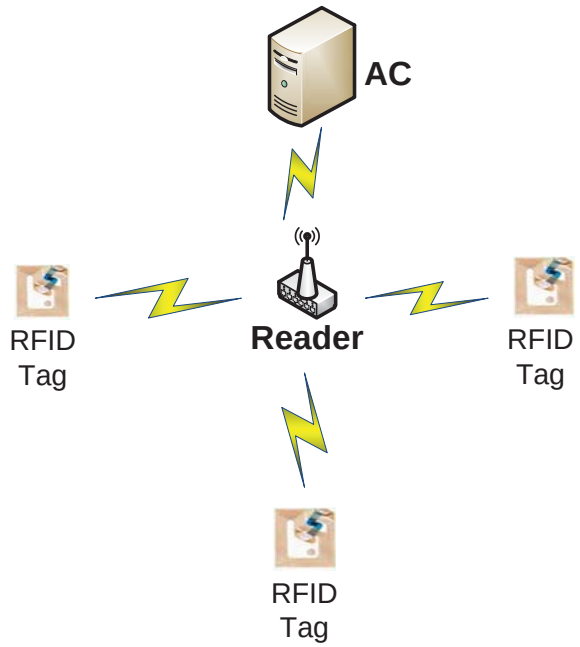

Figure 4: Model of Security Network

the tags with crucial information which are required for protection. The Authentication

Center in ISE is used to setup the cryptography and authenticate the authorization of Readers. The process of cryptography will be illustrated in the following "Steps of Protocol" part.

5.2.1.2 Algorithm

In order to configure ECC, we firstly select a particular elliptic curve E over GF(p), where p is a big prime number. We denote P as the base point of E, and a big prime q as the order of P. Then, we pick a secret key x (*x is the unique ID of the RFID Tags*), the corresponding public key y where $y = x.P$, and a hash function $h(.)$. Finally, we will have the public parameter set (y, P, p, q, h(.)) and a secret key x. For encrypting a message, we need to use public parameters.

- Generate a random number $r \in GF(p)$, encrypt message M with r to get M_r($M_r = M \bigoplus r$)
- Calculate $T_r = h(r).y$
- Calculate $D_r = h(r).P$
- Calculate $t_r = r \bigoplus T_r$
- Get cipher text $C = (t_r, D_r, M_r)$

For decrypting a message, we need to use the secret key x.

- Calculate $x.D_r = x.h(r).P = h(r).y = T_r$
- Derive symmetric key r, $t_r \bigoplus T_r = r \bigoplus T_r \bigoplus T_r = r$
- Use r to decrypt M_r, $M_r \bigoplus r = M$

The random number r can used to ensure the distinctive among different encryption. For the write-once tags, if any malicious attacker get a value of r, it can not be used to decrypt the other tags. For the rewritable tags, the value r can be used to update the encryption at any time.

5.2.1.3 Steps of Protocol

Our protocol includes four steps: Initialization, KeyGen, Encryption and Decryption.

- AC: Initially, in the AC side, it will start the Initialization and then execute the KeyGen.
  – Initialization: The AC choose an elliptic curve E over GF(p), to get the public parameters (P, p ,q, h(.))
  – KeyGen: AC Generate the set of public and secret key pairs (x_1, y_1),(x_2, y_2), ... ,(x_n, y_n).

– RFID tags: After the AC finish the KeyGen, the Encryption will be applied on the RFID tags.
– Encryption: Then we use these public parameters to encrypt the crucial information of each RFID tag, and write/rewrite into it.

– RFID Readers: When the readers need to read the information of the tags, it will ask AC for authentication and begin the Decryption procedure to get the crucial information.
– Decryption: For each searching request, the Reader will ask AC for authentication to read RFID tags, once authorized, it will receive the secret key to read and decrypt the message.

5.3 Experiments

In order to evaluate the performance of the proposed search engine framework, we carried out a series of experiments in order to test the common searching time, concurrency searching time and security execution time. These experiments are performed over 1,000,000 RFID records by Lucene 2.0, ISE, MySQL 5.0, running a Linux system with 2 GB memory, 2.4 GHz dual CPU and 1 TB disks. Every search gets top 20 results return. Particularly, we would like to check the performance of ISE in the following perspectives. One is that how the concurrency of ISE for large-scale IoT is. The other is the execution time of encryption in the runtime.

5.3.1 Concurrency of ISE

Figures 5 and 6 illustrate the results of common search and 10-user concurrent search, indicating that ISE outperforms Lucene and MySQL in scalability. During the common search, ISE performed 10 times faster than MySQL; when a concurrent search occurred, the ISE had significantly performance than the other two, 6 times faster than Lucene and 20 times than MySQL. To handle and respond requests sequentially, we adopt HTTPSQS as request queue processing protocol that is a very simple (less than 900 lines source code), but extremely high efficient (*http://code.google.com/p/httpsqs/*). Then we offer a queue controller to assist HTTPSQS. With respect to locality search, we currently rely on region division, which means that we firstly check local ISE database components to respond to requests.

5.3.2 Execution time of Security Cryptography

We implement our framework by using ”Miracl library” and c++. Initially, the equation $y^2 = x^3 + 1$ (where $a = 0$, $b = 1$) has been used for our GF(p), and the unique IDs of each RFID tag (160-bit) are used as secret keys. We separately measured the average speed of each steps in order to check whether it satisfying the real-time environment, and a 1024-bit RSA cryptography is used for the comparison.

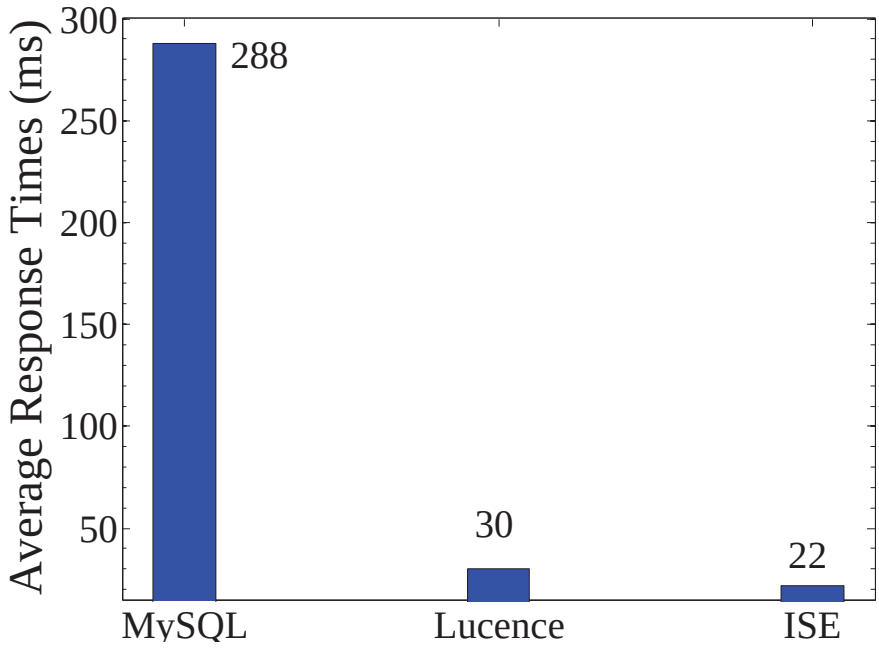

Figure 5: 1000 common search

Number	Initialization(ms)		KeyGen(ms)		Encryption(ms)		Decryption(ms)	
	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
1	7	12	28	50	3	3	2	2
2	6	11	30	48	4	2	3	2
3	7	13	31	45	3	5	2	3
4	8	15	25	50	3	2	3	2
5	10	12	27	48	4	2	3	2
6	6	11	30	49	4	3	3	3
Average	6.5	12.3	28.5	48.3	3.5	2.8	2.6	2.3

Table 3: Execution Time

From the table 3, we can see the ECC performed better than RSA in the Initialization and KeyGen process, and the ECC total execution time is smaller than RSA ($6.5 + 28.5 + 3.5 + 2.6 = 41.1ms$ compared with $12.3 + 48.3 + 2.8 + 2.3 = 65.7ms$). Due to the real-time characteristic of IoT, it does not tolerate high execution time, and the execution time of ECC in our experiments certainly satisfies such environment. Besides, the 160-bit secret key in ECC can offer the similar security as the 1024-bit secret key

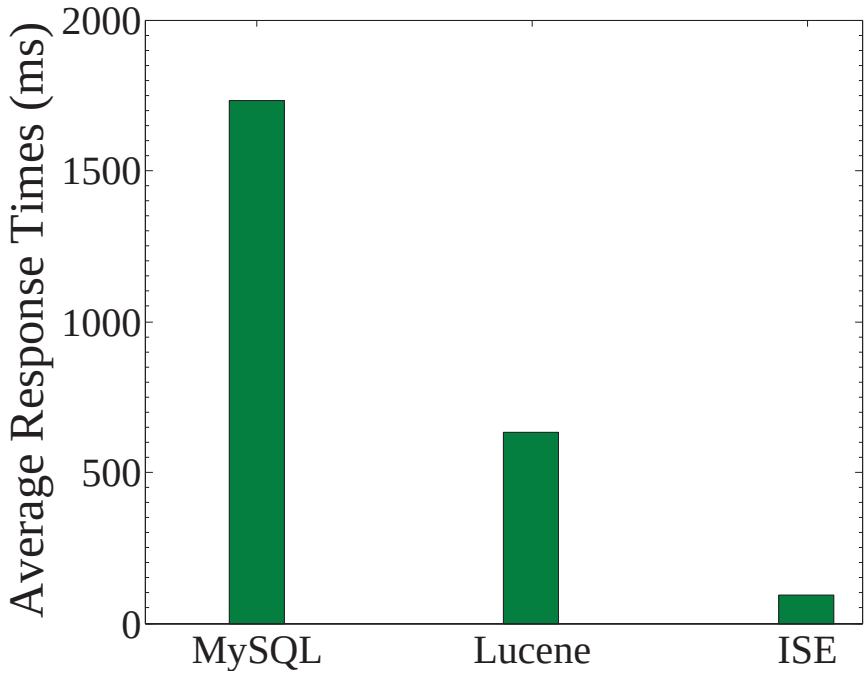

Figure 6: 1000 10-user concurrent search

in RSA, but with a much smaller key length.

6 Conclusion

Searching in Internet of Things is fundamental, which enables users to quickly locate physical objects and thus to control and operate them locally or remotely. More and more sensors are adopted across a multitude of fields. Correspondingly, the searching function becomes increasingly important.

In this paper, we have studied the searching problem in IoT, which imposes five challenging issues — architecture design, search locality, real-time, scalability and divulging information. We also put forward a set of measurement dimensions, involving eight dimensions that cover information aggregation, index and query perspectives. According to these dimensions, we overview the state of the art search engines that are designed in mobile computing, ubiquitous computing and information retrieve. We have briefly delineated our search effort ISE — a security-enhanced IoT Search Engine and reported our early results. As an ongoing project, ISE could be further improved in terms of locality search, privacy control and extensive evaluation.

References

[Aberer et al., 2006] Aberer, K., Hauswirth, M., and Salehi, A. (2006). Middleware support for the Internet of Things. In *Proceedings of the 5th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze"*. EPFL.

[Aberer et al., 2007] Aberer, K., Hauswirth, M., and Salehi, A. (2007). Infrastructure for data processing in large-scale interconnected sensor networks. In *Proceedings of the 2007 International Conference on Mobile Data Management*, pages 198–205, Washington, DC, USA.

[A.Shamir, 1984] A.Shamir (1984). Identity-based cryptosystems and signature schemes. *Advances in Cryptology-Crypto 84*, pages 47–53.

[Bhagwat et al., 1996] Bhagwat, P., Perkins, C., and Tripathi, S. (1996). Network layer mobility: an architecture and survey. *IEEE Personal Communications*, 3:54–64.

[Bradai and Afifi, 2011] Bradai, A. and Afifi, H. (2011). A framework using ibc achieving non-repudiation and privacy in vehicular network. In *Network and Information Systems Security (SAR-SSI), 2011 Conference on*, pages 1 –6.

[Chen, 2012] Chen, Y.-K. (2012). Challenges and opportunities of internet of things. In *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pages 383 –388.

[Chui et al., 2010] Chui, M., Löffler, M., and Roberts, R. (2010). The Internet of Things. *McKinsey Quarterly*, (2):1–9.

[Coetzee and Eksteen, 2011] Coetzee, L. and Eksteen, J. (2011). The internet of things - promise for the future? an introduction. In *IST-Africa Conference Proceedings, 2011*, pages 1 –9.

[Corley, 2010] Corley, A. (2010). Real-time search stumbles out of the gate. *IEEE Spectrum*.

[Creado et al., 2009] Creado, O. M., Wu, X., Wang, Y., and Le, P. D. (2009). Probabilistic encryption–a comparative analysis against rsa and ecc. In *Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, ICCIT '09, pages 1123–1129, Washington, DC, USA. IEEE Computer Society.

[Das and Harrop, 2010] Das, R. and Harrop, P. (2010). RFID forecasts, players and opportunities 201 – 2021. Technical report, IDTechEx.com.

[Fox et al., 1996] Fox, A., Gribble, S. D., Brewer, E. A., and Amir, E. (1996). Adapting to network and client variability via on-demand dynamic distillation. In *Proceedings of the 7th International Conference on Architectural Support For Programming Languages and Operating Systems*, pages 160–170.

[Frank et al., 2008] Frank, C., Bolliger, P., Mattern, F., and Kellerer, W. (2008). The sensor internet at work: locating everyday items using mobile phones. *Pervasive and Mobile Computing*, 4(3):421 – 447.

[Gershenfeld et al., 2004] Gershenfeld, N., Krikorian, R., and Cohen, D. (2004). The Internet of Things. *Scientific American*, 291(4):76–81.

[Jeffery et al., 2006] Jeffery, S. R., Garofalakis, M., and Franklin, M. J. (2006). Adaptive cleaning for RFID data streams. In *Proc. of the 32nd VLDB*, pages 163–174.

[Jin et al., 2011] Jin, X., Zhang, D., Zou, Q., Ji, G., and Qian, X. (2011). Where searching will go in internet of things? In *Wireless Days (WD), 2011 IFIP*, pages 1 –3.

[Koblitz, 1987] Koblitz, N. (1987). Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209.

[Kortuem et al., 2009] Kortuem, G., Kawsar, F., Fitton, D., and Sundramoorthy, V. (2009). Smart objects as building blocks for the Internet of Things. *Internet Computing, IEEE*, 14(1):44–51.

[Malhotra et al., 2007] Malhotra, K., Gardner, S., and Patz, R. (2007). Implementation of elliptic-curve cryptography on mobile healthcare devices. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pages 239–244, London, UK. IEEE.

[Margery, 2010] Margery, C. (2010). Sensors empower the "Internet of Things". *Electrical Design News*, pages 32–38.

[Miller, 1986] Miller, V. S. (1986). Use of Elliptic Curves in Cryptography. *Advances in Cryptology - CRYPTO '85: Proceedings*, pages 417+.

[Nils Gura, 2004] Nils Gura, A. P. (2004). Comparing elliptic curve cryptography and rsa on 8-bit cpus. *CHES2004*.

[Puliafito et al., 2010] Puliafito, A., Cucinotta, A., Minnolo, A. L., and Zaia, A. (2010). *Making the Internet of Things a reality: the WhereX Solution*, pages 99–108. Springer New York.

[Römer et al., 2010] Römer, K., Ostermaier, B., Mattern, F., Fahrmair, M., and Kellerer, W. (2010). Real-time search for real-world entities: A survey. *Proceedings of the IEEE*, (99):1–16.

[Royer and Toh, 1999] Royer, E. M. and Toh, C.-K. (1999). A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 6:46–55.

[Satyanarayanan, 1996] Satyanarayanan, M. (1996). Fundamental challenges in mobile computing. In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing*, PODC'96, pages 1–7, New York, NY, USA. ACM.

[Satyanarayanan, 2001] Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 8(4):10–17.

[Tait and Duchamp, 1992] Tait, C. and Duchamp, D. (1992). An efficient variable-consistency replicated file service. In *Proceedings of the USENIX File Systems Workshop*, pages 111–126.

[Tan et al., 2010] Tan, C. C., Sheng, B., Wang, H., and Li, Q. (2010). Microsearch: a search engine for embedded devices used in pervasive computing. *ACM Transactions on Embedded Computing Systems*, 9(4):1–29.

[Wang et al., 2010] Wang, H., Tan, C. C., and Li, Q. (2010). Snoogle: a search engine for pervasive environments. *IEEE Transactions on Parallel and Distributed Systems*, 21:1188–1202.

[Weiser, 1999] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3:3–11.

[Welbourne et al., 2009] Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., Balazinska, M., and Borriello, G. (2009). Building the Internet of Things using RFID: The RFID ecosystem experience. *Internet Computing, IEEE*, 13(3):48–55.

[Woelffle et al., 2010] Woelffle, H. S., Guillemin, P., Friess, P., and Sylvie (2010). *Vision and challenges for releasing the Internet of Things*. Publications Office of the European Union, Luxembourg.

[Wu et al., 2011] Wu, Y., Sheng, Q., and Ranasinghe, D. (2011). P2p object tracking in the internet of things. In *Parallel Processing (ICPP), 2011 International Conference on*, pages 502 –511.

[Yap et al., 2008] Yap, K., Srinivasan, V., and Motani, M. (2008). Max: wide area human-centric search of the physical world. *ACM Transactions on Sensor Networks*, 4(4):26.

[Zhang et al., 2011a] Zhang, D., Huang, H., Lai, C.-F., Liang, X., Zou, Q., and Guo, M. (2011a). Survey on context-awareness in ubiquitous media. *Multimedia Tools and Applications*, pages 1–33. 10.1007/s11042-011-0940-9.

[Zhang et al., 2011b] Zhang, D., Yang, L., and Huang, H. (2011b). Searching in internet of things: Vision and challenges. In *Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on*, pages 201 –206.

[Zhang et al., 2011c] Zhang, D., Zhou, J., Guo, M., Cao, J., and Li, T. (2011c). Tasa: Tag-free activity sensing using rfid tag arrays. *Parallel and Distributed Systems, IEEE Transactions on*, 22(4):558 –570.

[Zhang and Wang, 2008] Zhang, G. and Wang, S. (2008). A certificateless signature and group signature schemes against malicious pkg. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pages 334 –341.