

# A Distributed ePedigree Architecture

Dijiang Huang, Mayank Verma, Archana Ramachandran, Zhibin Zhou  
Arizona State University

**Abstract—** Current ePedigree creation and discovery services rely on a centralized framework, i.e., EPCglobal network. The centralized system has several restrictions to prevent it from being widely adopted. For example, it is un-scalable when ePedigree service requests are increased dramatically due to the item-level product tracking; it has little privacy protection since the product historical information can be easily derived from the Object Name Service (ONS) provided by the centralized EPCglobal network; it is cumbersome since the ePedigree information will be amplified in the local databases along with the product transportation stops; and so on. To overcome the above mentioned problems, we propose a distributed EPC Information Service (EPC-IS), which makes the ePedigree creation and discovery more robust, scaleable, and secure. Using our approach, the ePedigree historical records of a product is created and stored in the ePedigree creating parties' EPC-IS servers; in addition, each EPC-IS server maintains a look up table that stores the EPC-IS providers' one-hop up/down stream information. In this way, the ePedigree service creation and discovery are processed following a chain of processes with a distributed manner. The distributed ePedigree architecture and a set of EPC-IS service protocols are described in this paper.

## I. INTRODUCTION

An ePedigree is the historical records of a product distributed along the distribution system in the supply chain from its manufacture to the final customer. Electronic Product Code (EPC) realized by RFID technology transfers traditional Pedigree creation and verification from cumbersome paper processing to efficient digital handling. In healthcare domain, identifying counterfeit drugs becomes very critical. Thus, it is highly desired that the ePedigree should provide item-level tracking instead of massive-level tracking (e.g., pallet and container). Existing ePedigree discovery system relies on a centralized framework, i.e., EPCglobal network [3]. It brings visibility of assets,

reduction in inventories, just-in-time inventory handling, reduction in labor, and many other benefits.

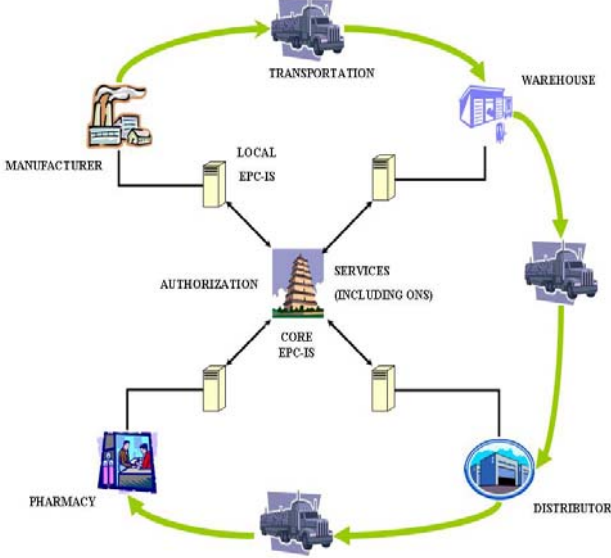
The EPCglobal Network premises a lot to manufactures, retailers, healthcare providers, hospitals, and many more industries in the ecosystem. However, it does bring up several concerns to prevent it from being widely deployed: (a) it is vulnerable to single point failure, (b) it is not scalable to handle a large number of requests, such as it involves heavy load of communication overhead and administrative overhead such as registration, records verification, updating, and processing; (c) when a particular object moves from one level to the other in the supply chain, more information about the object would be added for its usage at the next level and thus create additional overload in local databases; and (d) the EPC discovery to its ePedigree service mapping is maintained by a third party, and thus the centralized framework has potential privacy exposure problems.

To address the above mentioned problems, we propose a distributed EPC Information Service (EPC-IS) architecture for ePedigree creation and discovery, which makes the ePedigree services more robust, scaleable, and secure. The distributed architecture consists of various components such as Distributed Hash Table (DHT), ePedigree certificate services, ePedigree creation parties, and the EPC-IS service providers. Using our approach, the ePedigree historical records of a product is created and stored in the ePedigree creating parties' EPC-IS servers; in addition, each EPC-IS server maintains a look up table that stores the one-hop away EPC-IS providers' DHT *ids* and corresponding IP addresses; the ePedigree discovery service is via DHT searching. Note that all the operations of ePedigree creation and discovery require mutual authentication via certificate services. Thus, only legitimate parties are able to perform ePedigree related operations.

Our research goal is to construct a reliable and secure ePedigree creation and discovery infrastructure. Thus, the sharing of pedigree information common to organizations will ensure better security and data exchange mechanisms, resulting in smoother operations and eradicating the storage of repetitive data.

The rest of the paper is arranged as follows: in section II, we provide the background of EPCglobal network; the system components to build up our distributed EPC-

IS services are presented in section III; in section IV, we describe the proposed distributed ePedigree architecture



**Figure-1: Centralized Architecture.**

in details; the protocols involved in our ePedigree architecture are presented in section V; finally, we summarize our research and provide the research directions in section VI.

## II. BACKGROUND

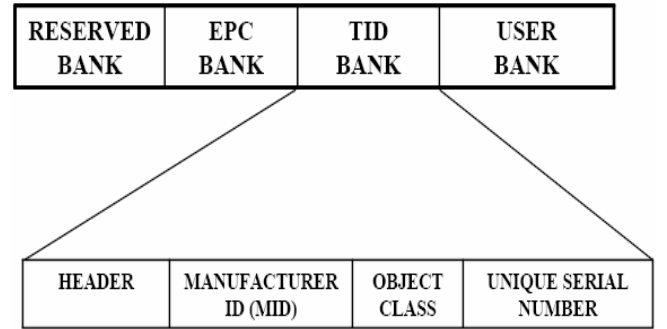
Research in creating ePedigree has increased extensively in recent years. Most of existing work is relying on a centralized approach [2]. EPC and ISO have been working in this area, but the major contribution has been done by EPCGlobal. The supply chain and ePedigree overview along with demonstration of EPCglobal network is described in [1] and [2], respectively. The architectural framework of ePedigree has been described in [3]. It explains the various components that combine to form the entire framework with the technical principles and their role and interfaces along with the data flow among relationship between various interfaces.

The centralized architecture of ePedigree is shown in Figure-1. It contains a centralized authority, which includes ONS services [4]. There exists a direct relationship between every participant in the network with the centralized server. Every participant in EPCglobal network maintains an EPC-IS (Electronic

Product Code) server, which stores relevant information related to specific EPC numbers. In centralized approach, when a query is submitted to EPCGlobal network, it is directed to root ONS, which returns the address of EPC-IS server containing the requested information. The major drawback of the existing ePedigree discovery and verification service is its centralized infrastructure. The centralized infrastructure imposes three restrictions: (1) It is

vulnerable to single point failure, (2) It requires complicated registration and processing procedures, and (3) The ePedigree owner has little control on the ePedigree discovery, which creates potential privacy exposure problems. Thus, it is highly desirable to replace the centralized ONS by distributed EPC discovery service. A distributed peer-to-peer network infrastructure is a promising candidate.

ePedigree makes use of RFID tags [5] for identification of products. There exist various types of tag formats. They fall under two basic categories. General, EAN.UCC system and DoD identity type [5]. We base our protocol on generation2 (gen2) tag (see Figure-2), as there is no standard tag format. The use of generation-2 (gen-2) tags makes our protocol more robust. In gen-2 tags, tag the memory is separated into four distinct banks, each of which comprises one or more memory words, where each word is 16 bits long. These memory banks are described as “Reserved”, “EPC”, “TID” and “User”. The “Reserved” memory



**Figure-2: Generation2 RFID tag.**

bank contains kill and access passwords. The “EPC” memory bank contains data used for identifying the object to which the tag is or will be attached, the “TID” memory bank contains data that can be used by the reader to identify the tag’s capability, and “User” memory bank is intended to contain user-specific data. As ePedigree uses RFID tag to identify tag by reading it, problem of unauthorized reading can result in information leak. So we provide a mechanism by which, the information in the tag can only be read and modified by authenticated participant. This is attained by encrypting information in user bank of the tag.

As we plan to make ePedigree architecture distributed, we use concept of distributed DNS as basis of our protocol. A distributed DNS service using dHash (distributed hash table - DHT) was proposed in [7]. dHash [8] is a Chord [10] based distributed hash table. The basic idea of a distributed hash table is to map the domain name of a node to its IP address using a hash function. The IP addresses are stored and retrieved using a distributed hash table protocol. The key generated by hashing the domain name of a node is used as an index

to place the node's IP address in Chord. In our approach, we combine the information from RFID tag [5] and distributed ONS using the idea of distributed DNS to perform the discovery services. Apart from Chord, there are many DHT protocols such as Pastry [9] and Tapestry [11] which can be used to implement distributed hash table. In this paper, we use Chord to explain our idea.

### III. SYSTEM COMPONENTS

There are four major components in our distributed Electronic Product Code – Information Service (EPC-IS) infrastructure, Object Name Service (ONS) based on Distributed Hash Table (DHT), EPC-IS databases, RFID encryption and decryption and certificate authorities. They are described in details below.

#### A. Object Name Services Using DHT

In current ONS service, a customer submits a query to EPCGlobal network [2], which is processed by a centralized ONS maintained in the EPCGlobal architecture. The ONS performs lookup service to find the address of the responsible EPC-IS server that contains the requested information [3]. Several difficulties exist in using this approach: (a) it is vulnerable to single point failure and is not scalable to handle massive requests; (b) it involves heavy load of administrative overhead such as registration, records verification, updating, and processing; and (c) the participants who create ePedigree have little control on the ePedigree discovery service. Since the EPC to EPC-IS mapping is maintained by a third party, the centralized framework has potential privacy exposure problems.

We create a distributed ONS structure by combining ONS and DHT. We first construct a distributed ONS structure using any of the DHT protocol. In this paper, we choose Chord [10] to explain the working of our protocol. The ID of the EPC-IS server is hashed to get a key. The location of an EPC-IS server in Chord is determined by using this key. Each of the EPC-IS servers maintains a hash table and store a list of mappings between a key and corresponding EPC-IS servers IP address in it. To find the IP address of an EPC-IS server requested by an EPC query, the EPC number is hashed to generate a key. This key is used for searching the DHT to locate the corresponding EPC-IS server's IP address. The creation and searching of the DHT is explained in detail in section V.

#### B. EPC-IS Databases

The EPC-IS is maintained by each participant in EPCGlobal network [1, 2] and it contains information mapping between a National Drug Code (NDC) and

corresponding electronic product code (EPC). In addition, the EPC-IS server stores each ePedigree processing history and corresponding signatures. The local database (EPC-IS) provides information for a submitted query and if it fails, distributed ONS is used to lookup the address of appropriate EPC-IS server, containing the requested information. Thus, in our approach there is no centralized server to which the queries are submitted. Instead, the distributed EPC-IS server is used to redirect the EPC query to an appropriate EPC-IS server, making the system more robust and scalable.

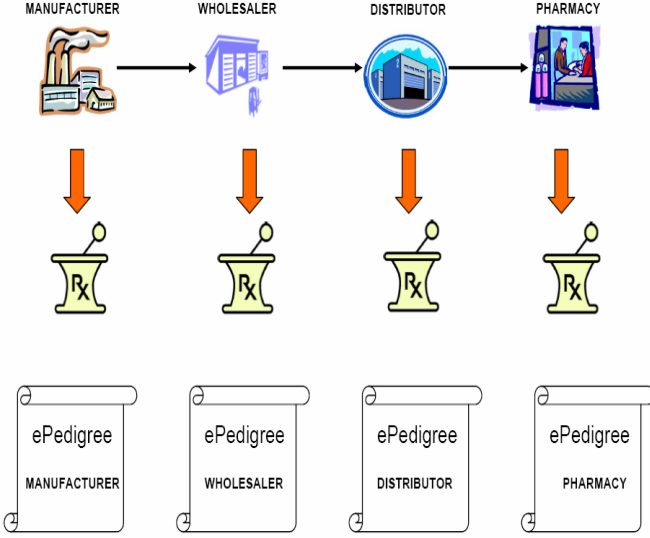
#### C. RFID tag encryption and decryption

ePedigree uses tag for identification and storing information about the product. This information is confidential and should only be accessed by authorized user. So we provide encryption and decryption mechanism for tag to prevent unauthorized read of information. Unauthorized reading can raise various concerns like an unauthorized user can claim the goods by read information in the tag and updating its EPC-IS.

In our mechanism, when the drugs are ready to be shipped from one participant to another participant downstream, the information in the user bank of the tag is encrypted. This encrypting is performed on entire lot with same key. The purpose of encrypting entire lot is to prevent the overhead of computation of encrypting every item independently. When tag information is to be read, decryption of tag is done using decryption key. The encryption and decryption is performed at each level by every participant in the supply chain. Every participant generates encryption and decryption for specific lot and stores that key pair in its EPC-IS along with the certificate associated with that lot. The encryption and decryption mechanism is explained in detail later.

#### D. Certificate Authorities

A certificate authority provides authentication services. All participants need to obtain a certificate from CA, to become a part of ePedigree. When a participant approaches CA to obtain a certificate, its identity is verified and after complete verification, the certificate is issued. These certificates play important role in ePedigree. Communicating participants use them for mutual authentication. Apart from that, by using certificates at each level (as discussed in section V. B) the certificate chain is build, which helps in traversing the chain and discovering level in ePedigree, where a possible drug counterfeit occurred.



**Figure-3: ePedigree Work Flow**

#### IV. A DISTRIBUTED EPEDIGREE ARCHITECTURE

Our distributed ePedigree architecture is divided into various components: ePedigree work flow, creating distributed ONS using DHT, ePedigree creation, ePedigree discovery, and verification service. The following sections describe them in detail.

##### A. ePedigree Work Flow

ePedigree work flow illustrates the working and relationship among various participants in network like manufacturer, distributor, wholesaler and pharmacy, which combine to form ePedigree. Figure-3 shows the flow of drugs from manufacturer to pharmacy, through the supply chain in ePedigree.

After the drugs have been produced by the manufacturer, a unique RFID tag is attached to every product. Manufacturer adds some information about himself in the user bank so that later it can be identified and authenticated by next level participant. The user bank of RFID tag is encrypted by manufacturer by using an encryption key before they are shipped to next level supplier. This encryption is done at lot level i.e. entire lot is encrypted using the same key. A certificate for the lot is created which contains information of the drugs that are being sent, along with information of sender and receiver of the drug. Manufacturer updates its EPC-IS server with information (lot number, respective EPC number, certificate and the key pair) and ships the drugs to next level participant downstream along with the information related to the drug. Before downstream participant can read the tag, it has to be decrypted. A request is made to the manufacturer for a decryption key with which the tag can be decrypted. Manufacturer verifies the identity of downstream supplier and provides

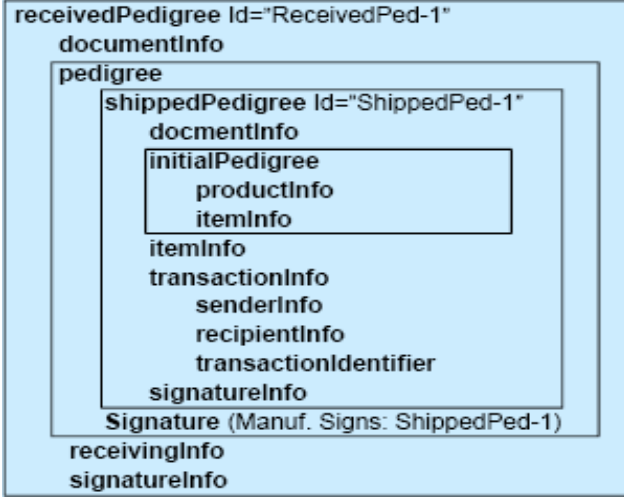
the decryption key. After decryption, user bank and tags are read and verified.

Downstream supplier uses the information send by manufacturer through secure channel, to authenticate the drug received. After authentication, drugs are accepted. Same procedure is repeated by every participant in ePedigree. Encryption and decryption is performed by every participant at each level. A new certificate for the drugs to be delivered is created by each participant. This ensure that the downstream supplier has information about previous hop only and not of the hop before that. This helps in increasing privacy. This procedure is performed until drug reaches the consumer, thus creating the work flow.

##### B. Creating Distributed ONS using DHT

We create distributed ONS in our approach to replace the existing centralized ONS. In centralized ONS, all the information about a specific EPC-IS was stored in one centralized entity. Every query is submitted to this entity, which provided the address of appropriate EPC-IS to the query generator. But using a centralized ONS have various disadvantages, so we propose a distributed ONS as its replacement.

In distributed ONS, the EPC-IS servers of all the participants take part in the address location service. The EPC-IS servers arrange themselves in a distributed manner using any of the DHT protocols. We explain our idea using Chord [8] protocol. Every EPC-IS server hashes its ID to get a key. The key is used to arrange them in the chord structure. Each of the EPC-IS servers stores the IP addresses of its successors apart from their own IP address in a hash table (i.e.) The hash table consists of a key and the IP address mapped to the corresponding key. Every EPC-IS server is a node in the chord and each of them contains a hash table thus forming a distributed hash table structure. In this way, the participants take part in the address discovery service and centralized server is removed. Once the structure is built, distributed ONS can be used to find an IP address of the EPC-IS server in the network. Before the downstream participant accepts the drug from the manufacturer, they do mutual authentication. For this mutual authentication, both of them should know each other's IP addresses. If they already know the peer's IP address, they go ahead with the authentication. Otherwise they need to send a request to the distributed ONS service to find the IP address of a responsible EPC-IS server. Also distributed ONS is contacted by a participant to find the IP address of the previous or the next level participant if the IP address is not know. This is done to verify the ePedigree and move back and forth



**Figure -4: ePedigree Format.**

in the chain as explained in V.C. In both cases distributed ONS is used to discover the IP address of an EPC-IS server in the network. Construction of the Distributed ONS is explained in detail in section V.

#### C. ePedigree Creation

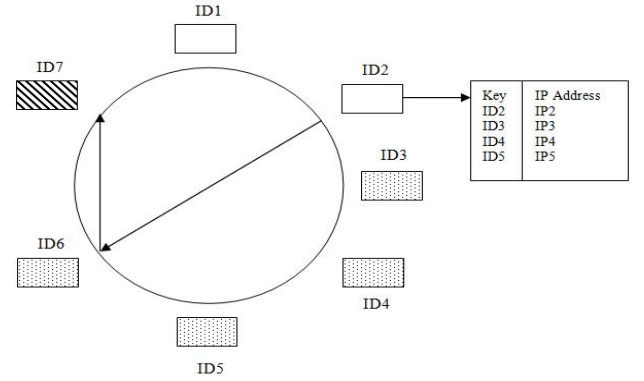
When these drugs arrive at specified receiver, they are decrypted and authenticated. If any mismatch occurs then upstream organization is contacted and after verification drugs are accepted. Thus, builds a link from manufacturer to first supplier.

The certificate received from sender is stored in the receiver EPC-IS and a new certificate is created instead of forming a layer on previous certificate. Format of certificate can be seen in Figure-4. Reason for creating a new certificate is to hide the details of previous level supplier from the 2 hop downstream supplier. For example if a distributor receives a certificate, it will create a new certificate in order to hide the details of his supplier i.e. wholesaler from his next downstream receiver i.e. pharmacy. This procedure keeps on repeating until the drug reaches the pharmacy, from where they are sold to consumer. Thus a complete ePedigree is build, along with a certificate chain from manufacturer to last supplier. This certificate chain is used for traversing the supply chain by performing discovery and verification services.

#### D. ePedigree Discovery and Verification Service

Discovery and verification service is an integral part of ePedigree. These services are required for locating EPC-IS servers and verifying whether the request, that has been made is by authenticated user or not. If the request is genuine, the address of appropriate EPC-IS server is found and provided to the requested user. Traditionally, these services were provided by ONS [4] which worked in centralized manner, thus was not very

efficient. Our discovery service works in a distributed manner and enables to find appropriate EPC-IS server for the requested information. It also provides verification services by authenticating the participant making the request by checking their certificate. Another benefit, which this discovery and verification services provide is, traversing the ePedigree back and forth for detecting the presence of counterfeit drug and the level at which it got counterfeited. This prevents further introduction of counterfeit drug and removal of sham participants from the supply chain.



**Figure-5: Constructing and Searching the DHT.**

### V. DISTRIBUTED EPEDIGREE PROTOCOLS

Our distributed ePedigree protocol consists of three phases: phase I includes creation protocol for distributed ONS; phase II deals with creation protocol for ePedigree; and phase III includes discovery and verification procedures.

#### A. PHASE I: Distributed ONS creation protocol

Before creating and verifying the ePedigree, the distributed ONS has to be constructed involving the EPC-IS servers of all the participants. The distributed ONS creation protocol is somewhat similar to “Serving DNS using a Peer-to-Peer Lookup Service” [7] but our protocol can be adapted to any DHT protocol like chord, Pastry, Tapestry, etc. Whenever an EPC-IS server joins the network, its unique ID (e.g., name, IP address, etc) is hashed to generate a key. Let’s call this key as Mkey. Based on this Mkey value, the servers are arranged in the DHT. This can be done by using any DHT protocols. For example let’s consider the chord [8] protocol. Chord has a unique m-bit ID based on the number of participant in Chord, i.e., if there are K participants then the ID’S range from 0 to  $2^k - 1$ . The structure of chord is shown in Figure-5. Here ID1, ID2, etc denotes the keys of the chord. Mkey is mapped to one of the ID’s in the chord. For example, if Mkey is between ID1 and ID2, it will be mapped to ID2. Every server knows its own IP address and apart from that each server stores the ID and IP addresses of  $r$  ( $r = 2 \log_2 N$ , where  $N$  is the number of



severs) successors clockwise in chord. The EPC-IS server with the immediate highest ID is said to be the immediate successor of a server.

Now the question is how the server knows the IP addresses of its successor? When a server (say ID1) joins the chord, it sends request to an existing server (say ID5) to find the ID and IP address of its successor. The ID5 finds ID1's successors by searching the DHT [6] which is explained in V.B. Once the successor ID2 is identified, ID1 requests ID2 for its successor and so on until it identifies  $r$  successor's IP address. Before the IP address is stored in a server, it is verified and signed by the certificate authority [6]. This is done to certify that this IP address belongs to the corresponding EPC-IS server. When a new EPC-IS server joins the network and say it's mapped to ID2, the successors of ID1 become the successors of ID2. Moreover IP addresses are stored in redundant nodes (servers) so that even if one node holding the IP address fails, then the other nodes which store the replicas can be approached. The redundant nodes are chosen using pseudorandom function. The number of replicas varies according to the DHT protocol.

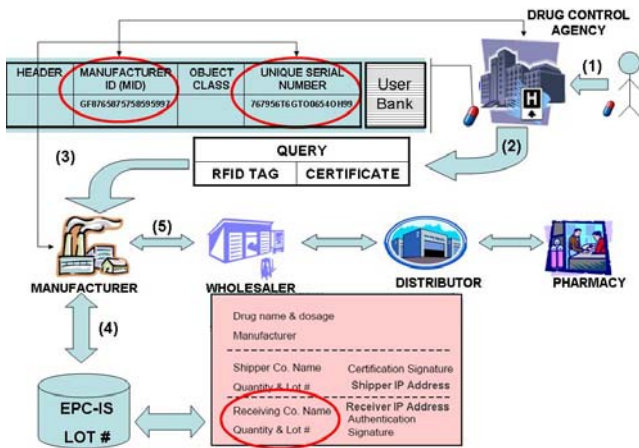
Searching the DHT is explained with chord as the DHT protocol. To find the IP address of a EPC-IS server we need to hash the ID of the server to generate a key (say) Mkey and map Mkey with one of the ID's in the chord as shown in Figure-5. Note that the ID of the EPC-IS server is encrypted. The requester of the IP address should first do the necessary authentication and get the decryption key as explained in section III.C. Once the decryption key is obtained, the user bank can be decrypted to yield the EPC-IS server's ID. The Mkey is given as a query to one of the existing servers in the chord say ID2. ID2 compares its ID with the Mkey. If the key falls between this node and its successor, then the successor is the intended node (7). If not, the node forwards the key to its successor whose ID is closest to this key. Each node maintains a hash table containing its successors' IDs and IP addresses. The same process continues until the ID closest to the key is found. Once the node is found, the IP address can be retrieved from the node. For example, if one of the participants in the ePedigree submits a query to ID2 asking to find the ID7, node ID2 compares its ID2 with ID7. From this comparison it knows that it is not the intended node. Now it looks into its hash table, and sees which of its successor is closest to ID7. In Figure-5, there are seven nodes in chord and each of the node stores  $r = (8\log_2 N)$  successors' IDs and IP addresses. The hash table of ID2 storing the IP addresses and keys of its successors is shown in Figure-5. The successors of ID2 are indicated in dotted squares. ID2 knows that ID6 is closer to ID7 so it forwards the request to ID6. ID6 will have ID7 in its

hash table and so it forwards the request to ID7. ID7 knows that it is the intended node by looking at the request and it retrieves the IP address and returns it. Before the IP address is retrieved, the signature of the certificate Authority on the IP address is verified. Thus the IP address of any EPC-IS is retrieved from the DHT.

### B. PHASE II: ePedigree Creating Protocol

This phase consist of creating ePedigree. As described in section IV.B, after the drugs have been produced and tagged by manufacturer at item level, they are placed in big cartons which have RFID tag of their own. These cartons combine together and form pallets, which in turn have their own RFID tag. These pallets form a lot and all the tags in this lot are encrypted using same encryption key. All information about the drug (i.e. all EPC numbers, certificates and the key pair) are stored in EPC-IS server of the manufacturer. The manufacture also adds some information in RFID user bank, which helps in his identification. All information added by manufacturer in user bank is encrypted by using encryption key. Manufacture initiates an ePedigree and signs a certificate which contains information about the drug, shipper co. name, the receiver co. name and their IP addresses. Before manufacturer ships out this drug to next level supplier, a mutual verification of each other is performed. The certificates which they obtained from Certificate authority are verified. After mutual authentication is complete, drugs are shipped to specified receiver. A list of all RFID tags along with their lot numbers is sends from manufacturer to next level receiver before drugs reach downstream supplier. This information is send through a secure channel. It is done to aid next level receivers to authenticate the drugs at the time of reception, by reading their EPC number and checking their validity by mapping it with the information previously received through secure channel from the manufacturer.

When drugs are received by next level supplier, a request is made to manufacturer for decryption key. After obtaining this key, user bank is decrypted to get identification of manufacturer. This information helps in authenticating sender of the drug (in this case, manufacturer). In case of mismatch, upstream organization (i.e. manufacturer) is contacted. Each participant before distributing the drug further down the chain performs same operations. It adds his identifiable information in tag user bank and encrypts it. A new certificate is signed, which contains information of sender and next level receiver only. The same procedure is repeated at every level. The main purpose of insertion information in RFID tag at each level is used to get the address of last supplier in the chain.



**Figure-6: Forward tracing.**

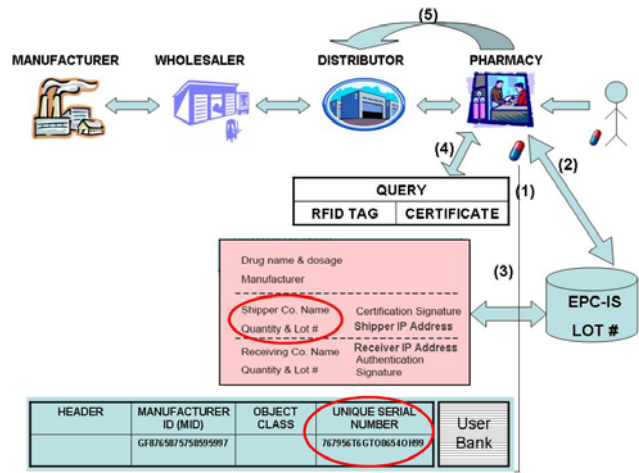
### C. PHASE III: ePedigree Discovery and Verification Protocol

The main goal of ePedigree is to prevent introduction of counterfeit drugs into the supply chain and also to trace back to the level at which they got counterfeited. Conventionally, discovery and verification services were performed by root ONS. Our distributed protocol provides various benefits over centralized services and mitigates its disadvantages.

Discovery services are used to locate the appropriate EPC-IS server, which contains information related to specific EPC number. By using discovery service, we can also traverse the ePedigree chain in both directions, forward (i.e. from manufacturer to last distributor in “chain of custody”) and reverse (i.e. from last distributor to manufacturer) to determine the location of a drug, in the supply chain. With the help of certificate chain build during the creation phase, discovery and verification protocol can trace down to the level at which counterfeit occurred. Traversing can be done in two ways, forward and reverse which are described below:

**Forward Tracing:** Suppose a consumer purchases a drug and after some time realizes that drug is counterfeited. But he doesn’t remember the pharmacy from where the drug was purchased. With the use of discovery and verification protocol, we can trace to the level at which this drug counterfeit occurred. Figure-6 describes step by step working of discovery and verification protocol.

**Step 1:** Consumer reports to a drug controlling agency that the drug he has is counterfeit. The drug controlling agency reads the RFID tag and obtains the manufacture ID, as it was not encrypted. Then it will hash the manufacture ID to get the Mkey. All we need to do is to map this Mkey with one of the ID in the chord. The IP address of the server can be obtained by searching the DHT which is explained in V.B. In this case the agency queries one of the existing servers in Chord to find the IP address of the EPC-IS server by giving the generated Mkey. Generally the result of DHT searching is the IP



**Figure-7: Reverse tracing.**

address of the EPC-IS server of the participant. Note that the IP address is signed by a Certificate Authority before it is stored in a node. So before the IP address is retrieved the signature on the IP address is verified. Then manufacture is located by using this IP address and his signature can be verified. **Step 2:** upon manufacturer authentication, the query is submitted to the manufacturer.

**Step 3:** After receiving the query from agency, manufacturer reads the tag and get the unique serial number (EPC number), since it cannot read the information in user bank as it is encrypted.

**Step 4:** Manufacturer performs lookup in his EPC-IS server to get the information (lot number) associated with that tag. After lot number is found, corresponding certificate is obtained. This certificate contains the information about the next level receiver of the drugs. From this certificate next level receiver information is retrieved.

**Step 5:** The query is forwarded to next level receiver in downstream, whose address was obtained from the certificate. The next level supplier also performs the same operations. The EPC number is read from the tag and lookup service is used to get the associated information available in the downstream EPC-IS server. If searching the EPC number in downstream EPC-IS does not result in any corresponding entry, then we conclude that drug got counterfeited between these two participants. As the drugs were present in supply chain and EPC number information was available with upstream supplier but was missing in downstream supplier, we can deduce that downstream supplier was involved in counterfeit. The main reason of us concluding this is because when downstream supplier received the counterfeit drug, their information would not have been present in the certificate send by upstream supplier. So authentication should have failed but still drugs were accepted and forwarded downstream. This makes it obvious that particular downstream supplier is involved in counterfeit. If specific entry, to particular EPC number is not found, we conclude that drug got counterfeited at next level supplier. But if the

corresponding entry exists in EPC-IS server of downstream supplier, then related information is retrieved from downstream supplier EPC-IS server and the query is further forwarded to next level downstream supplier, which performs the same operation. The certificate chain is traversed till the point, where the information mismatch occurs. Once the forged participants are found they can be questioned and put under surveillance, to find out who performed the counterfeit.

There exists a possibility, when two communicating participants are involved in counterfeit. They both can modify their own EPC-IS to store the false EPC number, but even then our protocol will be able to trace them. As the two participants do not have an agreement with the upstream authority, so the entries in upstream EPC-IS won't match. The probability that all participants in ePedigree chain are forged is very low.

**Reverse Tracing:** Reverse tracing is done in the scenarios when the consumer bought drugs from a particular pharmacy and discovers the drug is counterfeit. Consumer goes back to the pharmacy and reports the counterfeit. We use our discovery and verification protocol to determine the level at which counterfeit occurred explains it with Figure-7.

Step 1: The participant (pharmacy) read RFID tag with reader and obtains the unique serial number (EPC number).

Step 2: Participant performs lookup service in EPC-IS to retrieves the decryption key to verify the information (his own information) which it stored in user bank before the drugs were sold. After verification is complete, lot number is retrieved, which was maintained and updated in EPC-IS of participant at the time of reception of drugs from upstream supplier.

Step 3: if the particular EPC number is found in EPC-IS, the certificate associated with it is obtained. This certificate contains the information (Shipper number, Co. number, IP address, Lot # and the quantity) about the upstream supplier.

Step 4: Query is generated by the participant, which contains the RFID tag and the certificate.

Step 5: This query is reverted back to upstream supplier which on receiving the query, performs same operation. If upstream supplier is unable to find the EPC number in its EPC-IS, then we can conclude that the downstream participant is forged and is involved in counterfeit of the drug. If the EPC number is found, the associated certificate is retrieved from EPC-IS and the query is forwarded to next upper level supplier. At any level if the information mismatch occurs, we can conclude the point at which counterfeit took place.

## VI. CONCLUSION

In this paper, we propose a distributed EPC Information Service (EPC-IS) to make the ePedigree creation and discovery more robust, scaleable, and secure. Using our approach, the ePedigree historical records of a product is created and stored in the ePedigree creating parties' EPC-IS servers; in addition, each EPC-IS server maintains a look up table that stores the EPC-IS providers' one-hop up/down stream information. Our approach brings three-fold benefits for existing healthcare industry: (a) it relieves the single point failure due to the centralized EPCglobal network infrastructure, (b) it reduce the processing and storage overhead due to distributed processing and storage architecture, (c) it ensure the security and privacy via a chain of ePedigree operations where only authorized parties can conduct the ePedigree creation and searching functions.

It is a vital research area of ePedigree in healthcare industry. Many research problems and implementation issues are still unsolved and require more efforts from both academia researchers and industrial practitioners. Our research points out one of viable directions. We plan to involve more efforts from industrial side and set up testing bed to evaluate our solutions.

## REFERENCES

- [1] T. Applebaum, "ePedigree: Be Aware, Be prepared" in *Proceedings of Maxiom Group*, pages, 11-33, OCT 5, 2006.
- [2] EPCGlobal, "The EPCGlobal network demonstration", *EPCGlobal draft*, 2004.
- [3] EPCGlobal, "The EPCGlobal Architecture Framework", *EPCGlobal draft*, Final version, July 1, 2005.
- [4] EPCGlobal, "Object Name Service", *EPCGlobal draft, version 1*, October 4, 2005.
- [5] EPCGlobal, "EPCGlobal Tag Data Standards, version 1.3", *EPCGlobal draft*, March 8, 2006.
- [6] EPCGlobal "Drug Pedigree messaging interface JWG Requirement document, version 1", *EPCGlobal draft*, November 7<sup>th</sup>, 2005.
- [7] R. Cox, A. Muthitacharoen, and R. T. Morris, "Serving DNS using a Peer-to-Peer Lookup Service", in *proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, March, 2002, Cambridge, MA.
- [8] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS", in *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, Chateau Lake Louise, Ban.Canada, October 2001.
- [9] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-



- scale peer-to-peer systems”, in *proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, pages 329-350, Nov 2001.
- [10] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications”, in *IEEE/ACM Transactions on Networking (TON)*, Vol 11, Issue 1, Pages 17 - 32, 2003
- [11] B. Zhao and L. Huang and J. Stribling and S. Rhea and A. Joseph and J. Kubiatowicz “Tapestry: A Resilient Global-scale Overlay for Service Deployment”, in *IEEE Journal on Selected Areas in Communications*, 2003.