A Modified Approach for the Domain Name System Security

(DNSSEC)

Ms. Sneha S. Shahane,
Ms. Priyanka B. Shivgunde,
Ms. Jyoti P. Dalvi,
Ms. Madina M. Attar,
Ms. Poonam V. Kabra,
Department of Computer Science and Engineering,
Walchand Institute of Technology, Solapur.

Ms. R.A. Attar,
Assistant Professor
Department of Computer Science and Engineering,
Walchand Institute of Technology, Solapur,
Maharashtra, India, Pin: 413001.

***Abstract*—In this paper, we are presenting a modified approach to DNS Security using both Asymmetric and Symmetric cryptography. Domain Name System (DNS) is the Distributed database structure in which various domains are arranged hierarchically starting from the root. It allows storing and retrieving of the resource records (RR), resolving host names to IPs (Internet Protocol) and vice versa. The existing system provides the symmetric cryptography of the DNS protocol. In this paper, we are presenting a modified approach to DNS Security using both Asymmetric and Symmetric cryptography. This strategy helps to build a secure channel from the root servers to other authoritative servers. Our new and efficient mechanism is able to provide the protection against replay attack. It also gives confidentiality that leads to minimum exposure of the information. This approach preserves existing features of the DNSSEC protocol with additional security levels. This strategy helps to build a secure channel from the root servers to other authoritative servers.**

***Keywords*— public key;private key; secret key;hmac**

I. INTRODUCTION

Domain Name System (DNS) is the Distributed database structure in which various domains are arranged hierarchically starting from root [1]. It consists of following:

A. Plaintext

Plaintext message is a simple text message that can be easily read by the user.

B. Ciphertext

Cipher text is formed by encrypting the plaintext message by using encryption algorithm.

C. Encryption

In cryptography, encryption is the course of encoding data like messages, mail, etc., using an encryption algorithm in such a way that eavesdroppers or hackers cannot read it, but that message only read by authorized parties.

D. Decryption:

Decryption is the process of decoding the ciphertext into plaintext. Decryption of ciphertext is done by using various decryption algorithms.

E. SK-DNSSEC

In Symmetric-key schemes, the encryption and decryption keys are the same.

F. PK-DNSSEC:

In public-key encryption schemes, encryption of messages is done by using one key and decryption of messages is done by using the other key. These separate keys are generated in which one key is validated and freely shared called public key and one which is kept undisclosed is private key. The messages are encrypted and decrypted with help of above mentioned keys.

II. LITERATURE REVIEW

A. Domain Name System with Security Extensions

This paper presented DNS Wrapper, used to prevent spoofing. The DNS Wrapper examines incoming & outgoing messages. They have implemented one DNS prototype named as ‘Security Wrapper’. The Security Wrapper is piece of software that encapsulates component such as name server to improve security. The Symmetric Key Cryptography uses the concept of master keys. In this each node shares master key with its parent. The root has its own Master key & pair of Public and secret keys. To communicate with server it uses Symmetric Certificates. In this method, master key is used to create the certificates which provide safe transfer of keys between levels. The master key is created by root server [2].

B. Network Security via Private-Key Certificates

In this paper, DNS Security solely based on Public key cryptography is used. In public key cryptography, public key, is validated and shared freely with the world, but the private key is kept secret. The key pair is used to encrypt and decrypt data. The data encrypted with the public key can be decrypted only by the corresponding private key. Public key security system trusts its user to validate each others' public keys & to manage its own private keys security. Public key is widely accepted because it doesn't need trusted key management. This technique is based on validating public key of other. The public cryptography is best suited for securing communication between servers, between sites & organization [3].

C. *A new Approach to DNS security (DNSSEC)*

In this approach, symmetric key cryptography is used. Minimum messages are send through the network. As it combines two messages i.e. certificate request for authentication and the query messages. Further they have also used hmac (Hash Message Authentication Code) also secret keys, master keys for encryption and decryption of messages [4].

III. DNS System and Architecture

DNS stands for Domain Name System, by which all Internet service addresses are created, maintained and used. The DNS (Domain Name System) is a massive network of servers that comprises the largest digital database on the planet and this database managed , , managed and regulated by several internet authorities, including the IANA (Internet Assigned Numbers Authority) and ICANN (Internet Corporation for Assigned Names and Numbers). ICANN coordinates the addressing system to ensure all the addresses are unique [5]. The Domain Name System is an essential component of the functionality of the Internet. The internet would cease without the DNS, which is the central database of the internet, to survive as we know it. DNS also stands for Domain Name server. Domain Name System is that it serves as the "phone book" for the Internet by translating easily memorized domain names to the numerical IP addresses.For example, the domain name www.solapur.com translates to the addresses 199.192.168.11(IPv4) and 2606:2820:228:6d:26bf:1447:1197:aa7 (IPv6).If you purchase a Domain Name, DNS servers are given IP address and corresponding domain name to your web server. When someone comes to your domain, then your DNS server translates that domain name into corresponding IP address, so client's browser knows where to send request [6]. Domain Name Server is also known as Domain Name Service. Domain Name System allows user to find a system without knowing its IP address. Domains are logically organized as an inverted tree. In DNS tree, domains are divided into zones and delegating responsibilities to the corresponding zone. Name Server maintains database of host information for its zone. For getting host information such as Internet protocol address, it needs to contact the authoritative Name Server. When information at host changes in the zone, then that information needs to be updated [7]. (See fig 1.1)

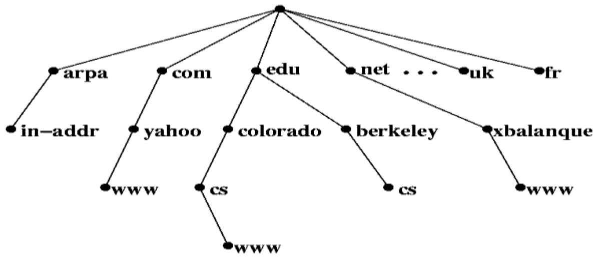


Fig 1.1

Root Node is called as Empty Domain. In the above figure, Root Node is denoted by (.). Domain name needs to be globally unique. A domain name refers to a node in the tree which called as Domain Name Space. Domain name consists of strings separated by (.) Each sub tree is called as Domain. For example, www.yahoo.com, where .com is sub domain of root domain (.com) which includes all domain names ending with (.com).The nodes that are direct children of the root node are called as Top level Domain. The top level domain include [8]

1. Generic or Organizational domains: These include three character domains such as .com for commercial organizations, .edu for educational institutions in the U.S.

2. Country domains: These include two character domains ranging from Ascension Island .ac to Zimbabwe .zw. The United States is also listed as .us.

3. ARPA domain: A separate .arpa domain controls the translation of IP addresses into domain names.

The root server is run by the Internet Corporation for Assigned Names and Numbers (ICANN). Similarly the second-level domain servers are under the control of the first-level domain and so on [9].

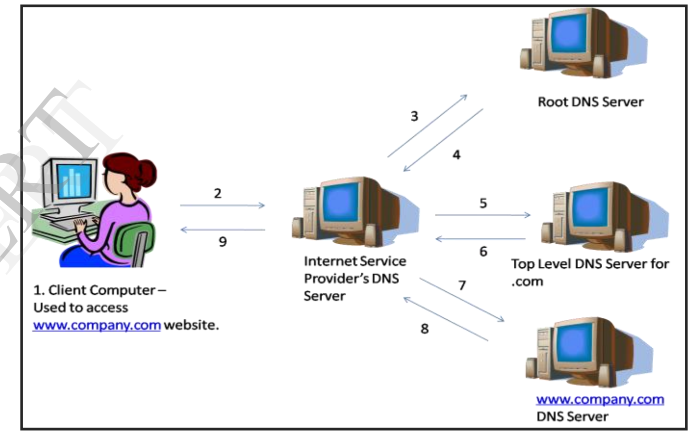

Root DNS Server
3
4
2
9
5
6
Internet Service Provider's DNS Server
Top Level DNS Server for .com
7
8
1. Client Computer – Used to access www.company.com website.
www.company.com DNS Server

Fig 1.2

A. *How Domain Name System works ?*

1. Client enters 'www.company.com' where Client computer wants the IP address version of 'company.com' and the first checks its own DNS cache for this information. It cannot find the IP address here if this is the first time using this website or the cache has been cleared [10].

2. The client computer request for the IP address of www.company.com is then redirected to the Internet Service Provider's DNS Server. The ISP's DNS server verifies its own cache. If the site has not been accessed before, it will be absent.

3. Every DNS server has a file that contains a list of all of the root DNS servers; at this the ISP's DNS server redirects the query to the Root DNS Server [11].

4. The root DNS server maintains information about where a top-level DNS server (.com) is located and returns this information to the ISP's DNS Server.

5. The ISP's DNS server redirects the query to a top-level DNS server (.com).

6. The top-level (.com) DNS server knows the IP address of the DNS server for the 'company.com' domain and returns that information to the ISP's DNS server.

7. The ISP's DNS server redirects the query to the actual DNS server for the 'company.com' domain.

8. The DNS server for 'www.company.com' returns the IP address of the host of 'www.company.com' to the ISP's DNS server.

9. Lastly, the ISP's DNS server sends the IP address to the client computer so the client can access 'www.company.com'.

IV. NEED OF DNSSEC

Recently vulnerabilities in the DNS were discovered that allow an attacker to hijack the process of looking someone up or looking a site up on the internet using their name. The purpose of the attack is to hijack the session and copy the authentication information of account like number and password. These vulnerabilities have increased interest in introducing a technology called DNS Security Extensions (DNSSEC) to secure this part of the Internet's Infrastructure [12]. IP addresses in DNS database are changed by unauthorized hosts to point traffic destined for one domain to another. By addressing DNS security weaknesses, the purpose of Domain Name System Security Extensions is to increase the security of the Internet as a whole. To make the system more secure, DNSSEC adds authentication to DNS. The security weakness leaves the system vulnerable to a number of attacks [13].

There are several DNS attacks are follows:

1. DNS cache poisoning

In a DNS cache poisoning attack, rogue address is placed by intruder at valid IP address cached in a DNS. Requests for the valid address are redirected accordingly, and malware , such as a worm, spyware or browser hijacker may be downloaded to the user's computer from the rogue location.

2. DNS Spoofing

DNS spoofing is a computer hacking, whereby the data is introduced into a Domain Name System (DNS) name server's cache database, causing the name server to return an incorrect IP address diverting traffic to another computer (often the attacker's) [14].

V. PROPOSED WORK

We have developed a secured DNS system in which even when the request from resolver has been received by the attacker, it would not be able to reply for the same. We have also shown fake server, which tries to enter into the communication between the resolver and servers & grant the access. Our system also maintains log files for all the connections established in the network.

A. Equations

1. Notations:

   a) SK: Secret Key
   b) PUR: Root's Public key
   c) ESK: Encryption using Secret Key
   d) EPUR : Encryption using Root's Public Key

2. Resolver and DNSRootServer

   a) Resolver → DNSRoot: EPURi(SK, hostname)

   b) DNSRoot → Resolver: ESK[hmac (EPURi(SK, hostname)), IPi)]

3. Resolver and Destination Server

   a) Resolver → DestServer:E_{SK}[hmacE_{PURi+2}(SK, hostname))]

4. DNSRootServer and DNSServer

   a) EPURi+1[SK, hmac (EPURi+1(SK, hostname))]

VI. METHODOLOGY

A. DNSRoot Server

DNS Root Server is at the top of the DNS hierarchy, every request first goes to DNS Root Server. DNS Root Server database maintains two files i.e. Public key and Private Key. It first checks whether these two files exist or not. If the files do not exist, it creates new files and generates a key pair using 'Encryption Manager'. Encryption Manager is the class which used for encryption purpose. All public and private keys related to Root Server are stored in the above files. After this, DNS Root receiving secret key and hostname from DNS Resolver, decrypt it using own private key. Obtain top level domain from receiving hostname then retrieving IP of the same from the database. The received IP and port are encrypted using a secrete key. Now the next step is reading public key of top level domain. Connect to that top level domain using the IP and port number. Encrypting token ,secrete key and IP using public key of top level domain and then send encrypted secrete key, encrypted IP and encrypted token to DNS Server. It creates HMAC of above encrypted messages using an algorithm MD5 and encrypts HMAC using secret key. Send this HMAC to top level domain. This procedure is repeated for each request from Resolver. The screenshot shows the output when it gets the request from the resolver. It has been implemented using virtual machines. DNS Root Server creates HMAC and sends this HMAC to next DNS Server and Resolver. (See Fig 1.4)

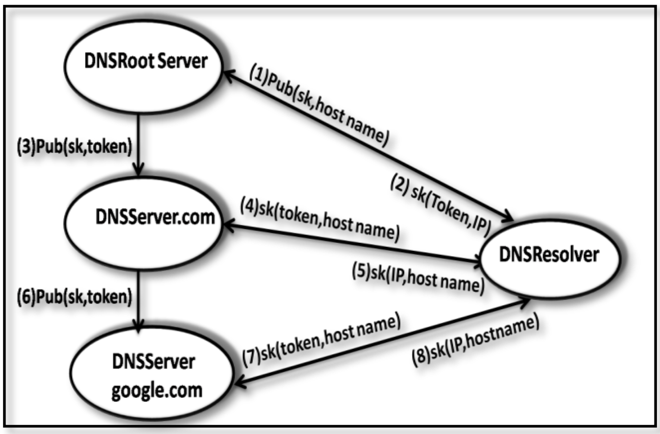


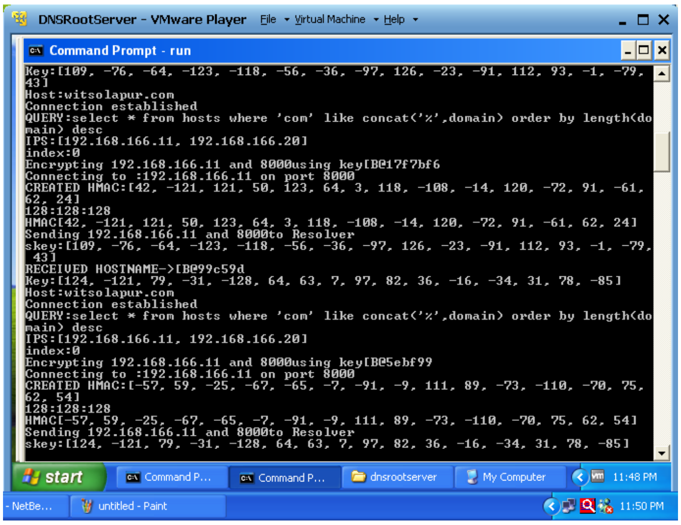
Fig. 1.3

B. *DNSResolver*

The main task of DNS Resolver is to take host name from client and returns it’s IP or vice versa. When it receives a host name it will first connect to the DNS Root Server and then reads the public key of DNS Root Server through this connection. AES algorithm is used for the generation of secret keys. DNS Resolver create Secret key using Encryption Manager. It will encrypt the host name and secret key using the public key of DNS Root Server then send this encrypted host name and secret key to DNS Root server. After receiving host name and secret key DNS Root server, send the top level domain’s port; address (IP) and HMAC to the DNS Resolver. DNS Resolver checks to see whether the desired hostname is retrieved or not, otherwise it iteratively sends IP and hostname to DNS Server, read the next domain name and IP address. Reply contains encrypted HMAC and IP of witsolapur.com which is in encrypted format. (See Fig 1.5)

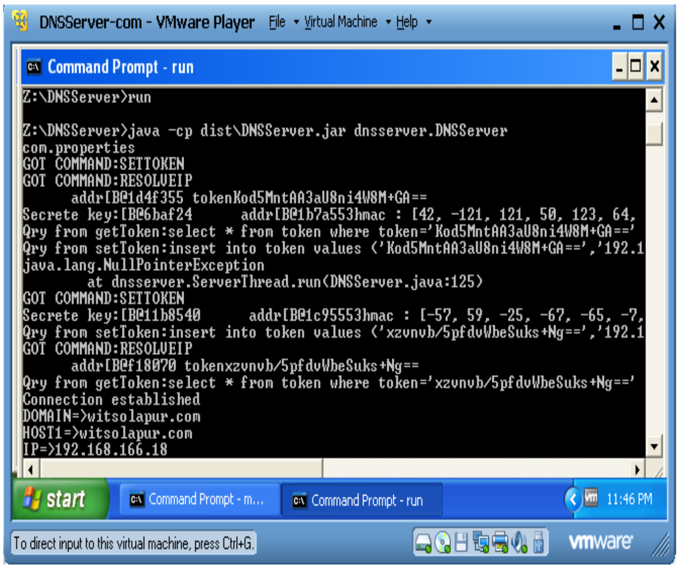
C. *DNSServer*

DNS Server is middle level servers which provide services. DNS Root Server forwards data i.e. HMAC and secret key to DNS Server using DNS Server public key. Received data is decrypted using its own private key. It stores the secret key in its own database. Then it compares HMAC received from the DNS Root Server with HMAC received from DNS Resolver, if it matches then it Obtains next hostname and corresponding IP from its database, else the request is rejected. After obtaining the next hostname and IP, it forwards secret key encrypted with next server’s public key and HMAC generated from this encrypted message. Every time the parent domain calculates HMAC for its child domain by using child’s public key. This output shows the working at the virtual machine which acts as DNS Server (.com). It starts its processing when it gets token (HMAC) from DNS Root Server which in the encrypted format. It also receives encrypted token (HMAC) from Resolver and host name (address: witsolapur.com), and checks these two tokens matched or not. If matching founds send the IP (192.168.166.18) to the resolver. The snapshot shows reply received by Resolver from DNS Server (.com). (See Fig 1.6)



```
Key:[109, -76, -64, -123, -118, -56, -36, -97, 126, -23, -91, 112, 93, -1, -79,
43]
Host:witsolapur.com
Connection established
QUERY:select * from hosts where 'com' like concat('%',domain) order by length(do
main) desc
IPS:[192.168.166.11, 192.168.166.20]
index:0
Encrypting 192.168.166.11 and 8000using key[B@17f7bf6
Connecting to :192.168.166.11 on port 8000
CREATED HMAC:[42, -121, 121, 50, 123, 64, 3, 118, -108, -14, 120, -72, 91, -61,
62, 24]
128:128:128
HMAC[42, -121, 121, 50, 123, 64, 3, 118, -108, -14, 120, -72, 91, -61, 62, 24]
Sending 192.168.166.11 and 8000to Resolver
skey:[109, -76, -64, -123, -118, -56, -36, -97, 126, -23, -91, 112, 93, -1, -79,
 43]
RECEIVED HOSTNAME->[B@99c59d
Key:[124, -121, 79, -31, -128, 64, 63, 7, 97, 82, 36, -16, -34, 31, 78, -85]
Host:witsolapur.com
Connection established
QUERY:select * from hosts where 'com' like concat('%',domain) order by length(do
main) desc
IPS:[192.168.166.11, 192.168.166.20]
index:0
Encrypting 192.168.166.11 and 8000using key[B@5ebf99
Connecting to :192.168.166.11 on port 8000
CREATED HMAC:[-57, 59, -25, -67, -65, -7, -91, -9, 111, 89, -73, -110, -70, 75,
62, 54]
128:128:128
HMAC[-57, 59, -25, -67, -65, -7, -91, -9, 111, 89, -73, -110, -70, 75, 62, 54]
Sending 192.168.166.11 and 8000to Resolver
skey:[124, -121, 79, -31, -128, 64, 63, 7, 97, 82, 36, -16, -34, 31, 78, -85]
```

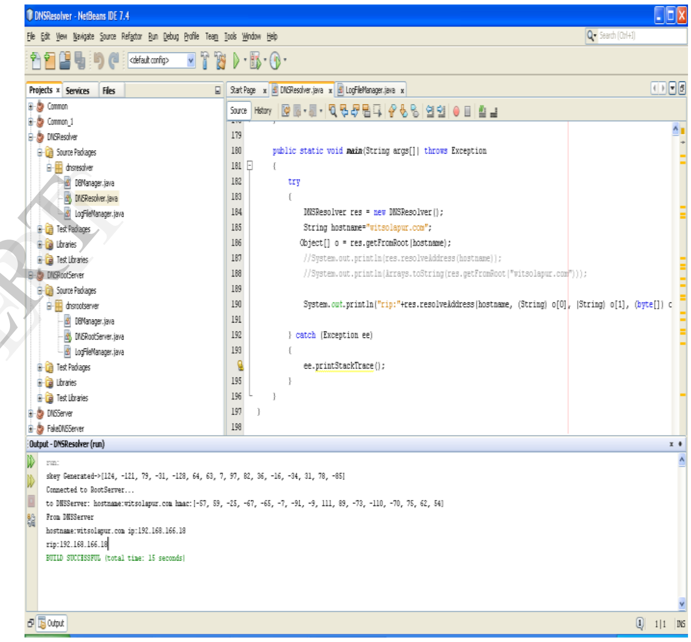
-Fig 1.4



```
Z:\DNSServer>run

Z:\DNSServer>java -cp dist\DNSServer.jar dnsserver.DNSServer
com.properties
GOT COMMAND:SETTOKEN
GOT COMMAND:RESOLVEIP
      addr[B@1d4f355 tokenKod5MntAA3aU8ni4W8M+GA==
Secrete key:[B@6baf24      addr[B@1b7a553hmac : [42, -121, 121, 50, 123, 64,
Qry from getToken:select * from token where token='Kod5MntAA3aU8ni4W8M+GA=='
Qry from setToken:insert into token values ('Kod5MntAA3aU8ni4W8M+GA==','192.1
java.lang.NullPointerException
        at dnsserver.ServerThread.run(DNSServer.java:125)
GOT COMMAND:SETTOKEN
Secrete key:[B@11b8540      addr[B@1c95553hmac : [-57, 59, -25, -67, -65, -7,
Qry from setToken:insert into token values ('xzvnvb/5pfdvWbeSuks+Ng==','192.1
GOT COMMAND:RESOLVEIP
      addr[B@f18070 tokenxzvnvb/5pfdvWbeSuks+Ng==
Qry from getToken:select * from token where token='xzvnvb/5pfdvWbeSuks+Ng=='
Connection established
DOMAIN=>witsolapur.com
HOST1=>witsolapur.com
IP=>192.168.166.18
```

Fig 1.5



```
    public static void main(String args[]) throws Exception
    {
        try
        {
            DNSResolver res = new DNSResolver();
            String hostname="witsolapur.com";
           Object[] o = res.getFromRoot(hostname);
            //System.out.println(res.resolveAddress(hostname));
            //System.out.println(Arrays.toString(res.getFromRoot("witsolapur.com")));

            System.out.println("rip:"+res.resolveAddress(hostname, (String) o[0], (String) o[1], (byte[]) c

        } catch (Exception ee)
        {
            ee.printStackTrace();
        }
    }
}
```

```
run:
skey Generated->[124, -121, 79, -31, -128, 64, 63, 7, 97, 82, 36, -16, -34, 31, 78, -85]
Connected to RootServer...
to DNSServer: hostname:witsolapur.com hmac:[-57, 59, -25, -67, -65, -7, -91, -9, 111, 89, -73, -110, -70, 75, 62, 54]
From DNSServer
hostname:witsolapur.com ip:192.168.166.18
rip:192.168.166.18
BUILD SUCCESSFUL (total time: 15 seconds)
```

Fig 1.6

VII. SCOPE

Now a day, there is been a lot of threat to the existing DNS protocol due to hackers. It has become a risk to redirect the current page to any desired page, because one may be redirected to any unsafe page which may cause to lower the trust in security measures. So our objective is to develop such a system which provides more security while we are surfing on the Internet and make the path from source page to requested page more secure. This paper provides an idea through which we can overcome the disadvantages of secret key cryptography. Our paper provides greater convenience as it solves the problem of key distribution. Public key encryption allows the use of digital signatures which enables the recipient of a message to verify that the message is truly from a particular sender. In our paper, the use of digital signatures in public key encryption allows the receiver to detect if the message was altered in transit.

VIII. DISCUSSION AND FUTURE WORK

Java is an open platform hence it is useful for implementation. It was originally developed by Sun Microsystems which was initiated by James Gosling. With the advancement of Java and its widespread popularity, multiple configurations were built to suite various types of platforms. Ex: J2EE for Enterprise Applications, J2ME for Mobile Applications. Sun Microsystems has renamed the new J2ME versions as Java SE, Java EE and Java ME, respectively. Java is guaranteed to be Write Once, Run Anywhere. Java is Object Oriented, Platform independent, Simple, Secure, Portable and Multithreaded.

For a back end MySQL is the most popular and efficient Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

VMware SERVER is a free virtualization product for Microsoft Windows and Linux servers. It enables users to quickly provision new server capacity by partitioning a physical server into multiple virtual machines.

We have performed our project using a virtual platform on a single machine, so in future we can implement this approach within LAN on different machines. Also there is still the improvement to be done regarding the speed as we have used asymmetric cryptography in communication within servers which lead to slower execution. There are various advantages of using asymmetric encryption, but the problem faced was the security of private key file with respective server. So in future we would work on how to increase the speed of this system and how to secure the private key file.

IX. CONCLUSION

We believe DNS would make a good distribution point of application keys and certificates for large scale systems. The main reason is that DNS is a unique provider of bindings between commonly used names. We presented a proposal for DNSSEC that, when properly implemented, offers the highest level of security while reducing network traffic. In addition, it reduces storage requirements and enables efficient mutual authentication. In particular, for highly critical parts of the DNS, like root servers or other servers near the root, our service can provide increased security. We rely on DNSSEC to provide authenticated delegation, while keeping the functional overhead of key distribution outside the critical DNS infrastructure. This strategy allows us to use the name service infrastructure to guarantee authenticity.

X. REFERENCES

[1] Giuseeppe Ateniese, Stefan Mangard,"A New Approach to DNS Security (DNSSEC)"

[2] Herbert Schildt,"Java Complete Reference", Fifth Edition.

[3] http://java.sun.com/j2me/learning/tutorial/index.html

[4] D. Davis and R. Swick, \Network Security via Private-Key Certificates", USENIX 3rd Security Symposium

[5] James M. Galvin, "Public Key Distribution with Secure DNS", in 6thUNIX Security Symposium

[6] Charishma G Shivaratri, "Domain Name System with Security Extensions"

[7] D. Eastlake, and C. Kaufman, "Domain Name System Security Extensions". RFC 2065.

[8] Eastlake, D., Gudmundsson, "Storing Certificates in the Domain Name System (DNS)", RFC 2538.

[9] RamaswamyChandramouli, Scott Rose, "Secure Domain Name System (DNS) Deployment Guide"

[10] http://en.wikipedia.org/wiki/Domain_Name_System.

[11] http://whatis.techtarget.com/definition/DNS-Security-Extensions-DNSSEC.

[12] [http://answers.oreilly.com/topic/2892-how-does-public-key-cryptography-work-in-dnssec/

[13] DNSSEC – What Is It and Why Is It Important _ ICANN.htm

[14] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034

[15] Guo, F.,Chen, J.andChiueh, T."Spoof Detection for Preventing DoSAttacks againstDNS Server