SECURITY SYSTEM FOR DNS USING CRYPTOGRAPHY

N.DEVI M.Sc., M.Ed., M.Phil.,
Assistant prof, Mangayarkarasi college of arts and science for women

P.Ramya^1 M.Sc(CS), P.Hema^2 M.Sc(CS)
Student of Mangayarkarasi college of arts and science for women

ABSTRACT
The mapping or strap of IP addresses to host names became a major problem in the rapidly growing Internet and the higher level binding effort went through different stages of development up to the currently used Domain Name System (DNS). The DNS Security is designed to provide security by combining the concept of both the Digital Signature and Asymmetric key (Public key) Cryptography. Here the Public key is send instead of Private key. The DNS security uses Message Digest Algorithm to compress the Message (text file) and PRNG (Pseudo Random Number Generator) Algorithm for generating Public and Private key. The message combines with the Private key to form a Signature using DSA Algorithm, which is send along with the Public key. The receiver uses the Public key and DSA Algorithm to form a Signature. If this Signature matches with the Signature of the message received, the message is Decrypted and read else discarded.

1. INTRODUCTION
The Domain Name System (DNS) can be considered one of the most important components of the modern Internet. DNS provides a means to map IP addresses (random, hard-to-remember numbers) to names (easier to remember and disseminate). Without DNS, we would have to remember that www.amazon.com is actually the IP address 72.21.207.65, and that would be hard to change. DNS is really the most successful, largest distributed database. In recent years, however, a number of DNS exploits have been uncovered. These exploits affect the system in such a way that an end user cannot be certain the mappings he is presented with are in fact legitimate. The DNS Security (DNSSEC) standard has been written in an attempt to mitigate some of the known security issues in the current DNS design used today. Finally, we will analyse the impacts of DNSSEC on embedded platforms and mobile networks.

2. LITERATURE SURVEY
The DNS was designed as a replacement for the older "host table" system. Both were intended to provide names for network resources at a more abstract level than network (IP) addresses (see, e.g., [RFC625], [RFC811], [RFC819], [RFC830], [RFC882]). In recent years, the DNS has become a database of convenience for the Internet, with many proposals to add new features. Only some of these proposals have been successful. Often the main (or only) motivation for using the DNS is because it exists and is widely deployed, not because its existing structure, facilities, and content are appropriate for the particular application of data involved. This document reviews the history of the DNS, including examination of some of those newer applications. It then argues that the overloading process is often inappropriate. Instead, it suggests that the DNS should be supplemented by systems better matched to the intended applications and outlines a framework and rationale for one such system. To connect to a system that supports IP, the host initiating the connection must know in advance the IP address of the remote system. An IP address is a 32-bit number that represents the location of the system on a network. The 32-bit address is separated into four octets and each octet is typically represented by a decimal number. The four decimal numbers are separated from each other by a dot character ("."). Even though four decimal numbers may be easier to remember than thirty-two 1’s and 0’s, as with phone numbers, there is a practical limit as to how many IP addresses a person can remember without the need for some sort of directory assistance. The directory essentially assigns host names to IP addresses. The Stanford Research Institute’s Network Information Center (SRI-NIC) became the responsible authority for maintaining unique host names for the Internet. The SRI-NIC maintained a single file, called hosts.txt, and sites would continuously update SRI-NIC with their host name to IP address mappings to add to, delete from, or change in the file. The problem was that as the Internet grew rapidly, so did the file causing it to become increasingly difficult to manage. Moreover, the host names needed to be unique throughout the worldwide Internet. With the growing size of the Internet it became more and more impractical to guarantee the uniqueness of a host name. The need for such things as a hierarchical naming structure and distributed management of host names paved the way for the creation of a new networking protocol that was flexible enough for use on a global scale [ALIU]. What evolved from this is an Internet distributed database that maps the names of computer systems to their respective numerical IP network address(es). This Internet lookup facility is the DNS. Important to the concept of the distributed database is delegation of authority. No longer is one single organization responsible for host name to IP address mappings, but rather those sites that are responsible for maintaining host names for their organization(s) can now regain that control.

Fundamentals of DNS

DNS not only supports host name to network address resolution, known as forward resolution, but it also supports network address to host name resolution, known as inverse resolution. Due to its ability to map human memorable system names into computer network numerical addresses, its distributed nature, and its robustness, the DNS has evolved into a critical component of the Internet. Without it, the only way to reach other computers on the Internet is to use the numerical network address. Using IP addresses to connect to remote computer systems is not a very user-friendly representation of a system's location on the Internet and thus the DNS is heavily relied upon to retrieve an IP address by just referencing a computer system's Fully Qualified Domain Name (FQDN). A FQDN is basically a DNS host name and it represents where to resolve this host name within the DNS hierarchy.

3. PROBLEM FORMULATION

1. Threats to the Domain Name System
Original DNS specifications did not include security based on the fact that the information that it contains, namely host names and IP addresses, is used as a means of communicating data [SPAF]. As more and more IP based applications developed, the trend for using IP addresses and host names as a basis for allowing or disallowing access (i.e., system based authentication) grew. Unix saw the advent of Berkeley "r" commands (e.g., rlogin, rsh, etc.) and their dependencies on host names for authentication. Then many other protocols evolved with similar dependencies, such as Network File System (NFS), X windows, Hypertext Transfer Protocol (HTTP), et al. Another contributing factor to the vulnerabilities in the DNS is that the DNS is designed to be a public database in which the concept of restricting access to information within the DNS name space is purposely not part of the protocol. Later versions of the BIND implementation allow access controls for such things as zone transfers, but all in all, the concept of restricting who can query the DNS for RRs is considered outside the scope of the protocol. The existence and widespread use of such protocols as the r-commands put demands on the accuracy of information contained in the DNS. False information within the DNS can lead to unexpected and potentially dangerous exposures. The majority of the weaknesses within the DNS fall into one of the following categories: Cache poisoning, client flooding, dynamic update vulnerability, information leakage, and compromise of the DNS server "s authoritative database. The answer received was related to the original question.

**METHODOLOGY PROPOSED SYSTEM**

Taking the above prevailing system into consideration the best solution is using Pseudo Random Number Generator for generating Key Pair in a quick and more secured manner. We use MD5 (or) SHA-1 for producing Message Digest and Compressing the message. Signature is created using Private Key and Message Digest which is transmitted along with the Public Key. The transfer of the packets from each System to System is shown using Graphical User Interface (GUI). Each time the System get the message, it verifies the IP Address of the sender and if no match is found it discards it. For verification, the Destination System generates Signature using Public Key and DSA Algorithm and verifies it with received one. If it matches it Decrypts otherwise it discards. The Following functions avoids the pitfalls of the existing system.

- Fast and efficient work
- Ease of access to system
- Manual effort is reduced

**4. WORK DONE**

Vulnerabilities in the DNS have frequently been exploited for attacks on the Internet. One of the most common ways of “defacing” a web server is to redirect its domain name to the address of a host controlled by the attacker through manipulation of the DNS. DNSSEC [9] eliminates some of these problems by providing end-to-end authenticity and data integrity through transaction signatures and zone signing. Transaction signatures are computed by clients and servers over requests and responses. DNSSEC allows the two parties either to use a message authentication code (MAC) with a shared secret key or public-key signatures for authenticating and authorizing DNS messages between them. The usefulness of transaction signatures is limited since they guarantee integrity only if a client engages in a transaction with the server who is authoritative for the returned data, but do not protect against a corrupted server acting as a resolver. For zone signing, a public-key for a digital signature scheme, called a zone key, is associated with every zone. Every resource record (it is the basic data unit in the DNS database) is complemented with an additional SIG resource record containing a digital signature,
computed over the resource record. Zone signing also protects relayed data because the signature is created by the entity who owns the zone.

Key Generation
Careful generation of all keys is a sometimes overlooked but absolutely essential element in any cryptographically secure system. The strongest algorithms used with the longest keys are still of no use if an adversary can guess enough to lower the size of the likely key space so that it can be exhaustively searched. Technical suggestions for the generation of random keys will be found in RFC 4086 [14]. One should carefully assess if the random number generator used during key generation adheres to these suggestions. Keys with a long effectively period are particularly sensitive as they will represent a more valuable target and be subject to attack for a longer time than short period keys. It is strongly recommended that long-term key generation occur off-line in a manner isolated from the network via an air gap or, at a minimum, high-level secure hardware.

Encryption and Decryption, Signature Creation, Signature Verification.

5. CONCLUSION

The DNS as an Internet standard to solve the issues of scalability surrounding the hosts.txt file. Since then, the widespread use of the DNS and its ability to resolve host names into IP addresses for both users and applications alike in a timely and fairly reliable manner, makes it a critical component of the Internet. The distributed management of the DNS and support for redundancy of DNS zones across multiple servers promotes its robust characteristics. However, the original DNS protocol specifications did not include security. Without security, the DNS is vulnerable to attacks stemming from cache poisoning techniques, client flooding, dynamic update vulnerabilities, information leakage, and compromise of a DNS server’s authoritative files.

In order to add security to the DNS to address these threats, the IETF added security extensions to the DNS, collectively known as DNSSEC. DNSSEC provides authentication and integrity to the DNS. With the exception of information leakage, these extensions address the majority of problems that make such attacks possible. Cache poisoning and client flooding attacks are mitigated with the addition of data origin authentication for RR Sets as signatures are computed on the RR Sets to provide proof of authenticity. Dynamic update vulnerabilities are mitigated with the addition of transaction and request authentication, providing the necessary assurance to DNS servers that the update is authentic. Even the threat from compromise of the DNS server ”s authoritative files is almost eliminated as the SIG RR are created using a zone ”s private key that is kept off-line as to assure key ”s integrity which in turn protects the zone file from tampering. Keeping a copy of the zone ”s master file offline when the SIGs are generated takes that assurance one step further.

DNSSEC cannot provide protection against threats from information leakage. This is more of an issue of controlling access, which is beyond the scope of coverage for DNSSEC. Adequate protection against information leakage is already provided through such things as split DNS configuration.

DNSSEC demonstrates some promising capability to protect the Internet infrastructure from DNS based attacks. DNSSEC has some fairly complicated issues surrounding its development, configuration, and management. Although the discussion of these issues is beyond the scope of this survey, they are documented in RFC 2535 and RFC 2541 and give some interesting insight into the inner design and functions of DNSSEC. In addition to keep the scope of this paper down, many topics such as secure zone transfer have been omitted but are part of the specifications in RFC 2535. The first official release of a DNSSEC implementation is available in BIND version 8.1.2.
REFERENCES