International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/22312803/IJCTT-V68I4P101

Volume 68 Issue 4, 1-4, April 2020

Original Article

Security Systems for DNS using Cryptography

Satyam Akunuri[1], Sanjeev Bandru[2], Chandu Naik Azmera[3]

[1,2,3]*Assistant Professor & Computer Science and Engineering & SNIST, Hyderabad, India*

Received Date : 23 February 2020
Revised Date: 09 April 2020
Accepted Date: 11 April 2020

Abstract *- The mapping or binding of IP addresses to hostnames became a major problem in networking systems. 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 sent instead of the Private key.*

The DNS security uses Message-Digest Algorithm to compress the Message (textfile) and PRNG(Pseudo Random Number Generator) Algorithm for generating Public and Private keys.

Keywords *- DNS, Digital Signature, Cryptography, ECC, ECDSA.*

I. 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.

II. SCOPE

The Domain Name System (DNS) has become a critical operational part of the Internet Infrastructure, yet it has no strong security mechanisms to assure Data Integrity or Authentication. Extensions to the DNS are described that provide these services to security-aware resolves are applications through the use of Cryptographic Digital Signatures. These Digital Signatures have included zones as resource records.

The extensions also provide for the storage of Authenticated Public keys in the DNS. This storage of keys can support general Public key distribution services as well as DNS security. These stored keys enable security-aware resolvers to learn the authenticating key of zones, in addition to those for which they are initially configured. Keys associated with DNS names can be retrieved to support other protocols. In addition, the security extensions provide for the Authentication of DNS protocol transactions.

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 sent instead of the 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 keys. The Message combines with the Private key to form a Signature using DSA Algorithm, which is sent along with the Public key.

The receiver uses the Public key and DSA Algorithm to form a Signature. If the Signature matches with the Signature of the Message received, the Message is Decrypted and read else discarded.

III. DOMAIN NAME SPACE (DNS)

As a tree is traversed in an ascending manner (i.e., from the leaf nodes to the root), the nodes become increasingly less specific (i.e., the leftmost label is most specific, and the rightmost label is least specific). Typically in an FQDN, the leftmost label is the hostname, while the next label to the right is the local domain to which the host belongs. The local domain can be a subdomain of another domain. The name of the parent domain is then the next label to the right of the subdomain (i.e., local domain) name label, and so on, till the root of the tree is reached

.

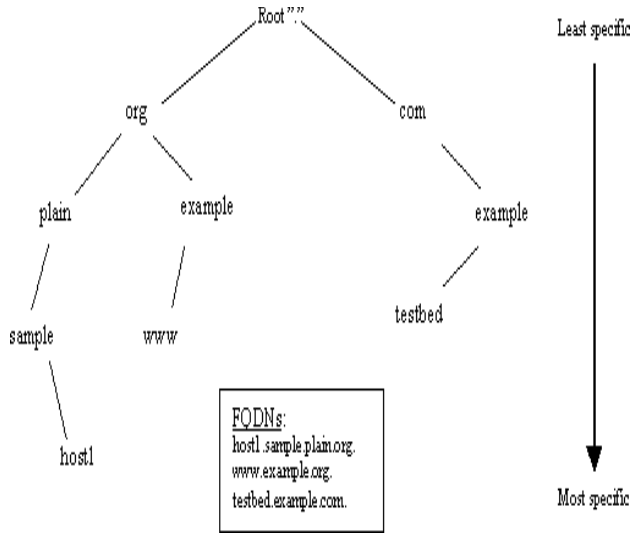


Fig. 3.1 Domain name space example

The DNS is a hierarchical tree structure hoserootnodeis known as the root domain. A label in a DNS name directly corresponds with a node in the DNS tree structure. A label is an alphanumeric string that uniquely identifies that node from its brothers. Labels are connected together with a dot notation, ., and a DNS name containing multiple labels represents its path along the tree to the root. Labels are written from left to right. Only one zero-length label is allowed and is reserved for the root of the tree. This is commonly referred to as the root zone. Due to the root label being zero-length, all FQDNs end in a dot [RFC 1034].

When the DNS is used to map an IP address back into a hostname (i.e., inverse resolution), the DNS makes use of the same notion of labels from left to right (i.e., most specific to least specific) when writing the IP address. This is in contrast to the typical representation of an IP address whose dotted decimal notation from left to right is least specific to most specific.

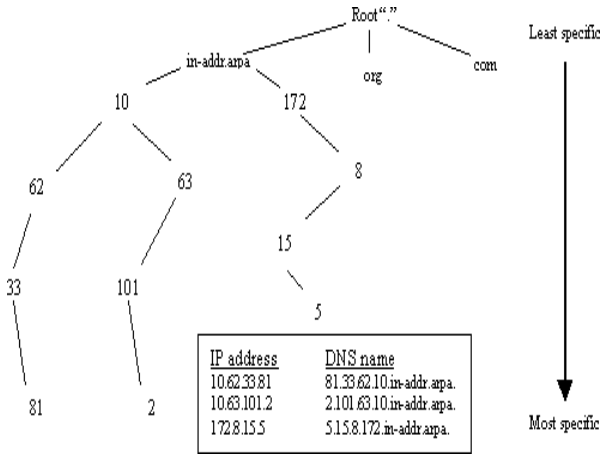


Fig. 3.2. Example of inverse domains and the Domain Name Space

To handle this, IP addresses in the DNS are typically represented in reverse order. IP addresses fall under a special DNS top-level domain (TLD), known as the in-and. Arpa domain. By doing this, using IP addresses to find DNS hostnames are handled just like DNS hostname lookups to find IP addresses.

A. DNS Components

The DNS has three major components, the database, the server, and the client [RFC 1034]. The database is a distributed database and is comprised of the Domain Name Space, which is essentially the DNS tree, and the Resource Records (RRs) that define the domain names within the Domain Name Space. The server is commonly referred to as a name server. Name servers are typically responsible for managing some portion of the Domain Name Space and for assisting clients in finding information within the DNS tree. Name servers are authoritative for the domains in which they are responsible. They can also serve as a delegation point to identify other name servers that have authority over sub-domains within a given domain.

B. DNS Transactions

DNS transactions occur continuously across the Internet. The two most common transactions are DNS zone transfers and DNS queries/responses. A DNS zone transfer occurs when the secondary server updates its copy of a zone for which it is authoritative. The secondary server makes use of the information it has on the zone, namely the serial number, and checks to see if the primary server has a more recent version. If it does, the secondary server retrieves a new copy of the zone.

A DNS query is answered by a DNS response. Resolvers use a finite list of name servers, usually not more than three, to determine where to send queries. If the first name server in the list is available to answer the query, then the others in the list are never consulted. If it is unavailable, each name server in the list is consulted until one is found that can return an answer to the query. The name server that receives a query from a client can act on behalf of the client to resolve the query. Then the name server can query other name servers one at a time, with each server consulted being presumably closer to the answer. The name server that has the answer sends a response back to the original name server, which then can cache the response and send the answer back to the client. Once an answer is cached, a DNS server can use the cached information when responding to subsequent queries for the same DNS information. Caching makes the DNS more efficient, especially when under heavy load. This efficiency gain has its tradeoffs; the most notable is insecurity.

IV. DNS SECURITY

A. Security Need

As originally designed, DNS has no means of determining whether the domain name data comes from the authorized domain owner or has been forged. This weakness in security leaves the system to be vulnerable to a number of attacks, like DNS cache poisoning, DNS spoofing etc. Due to weak authentication between DNS servers exchanging updates, an attacker may predict a DNS message ID and manage to reply before the legitimate DNS server, thus inserting a malicious record into the DNS database. The exploit forces a compromised DNS server to send a request to an attacker's DNS server, which will supply the wrong host to IP mapping.

DNS Security Extensions (DNSSEC) is a set of IETF (Internet Engineering Task Force) standards that have been created to address the vulnerabilities in the DNS and to protect from online threats. The main purpose of DNSSEC is to basically increase Internet security as a whole by addressing and resolving DNS security weaknesses. Essentially, DNSSEC adds authentication feature to DNS that make the system more secure DNSSEC core elements were specified in the following three IETF Requests for Comments which have been published in March 2005: RFC 4033 - DNS Security Introduction and Requirements RFC 4034 - Resource Records for the DNS Security Extensions RFC 4035 - Protocol Modifications for the DNS Security Extensions Existing proposals for securing DNS are mainly based on public-key cryptography. The public key algorithms used for authentication in DNSSEC are MD5/RSA (Rivest Shamir Adleman Algorithm) and DSA (Digital Signature Algorithm). Digital signatures generated with public-key algorithms have the advantage that anyone having the public key can verify them.

The idea behind it is that every node in Domain Name Space has a Public Key, and each Message from DNS Servers is signed using Private Key. Since DNS is Public, Authenticated DNS root Public Keys are known to all, which are used to generate Certificates/Signatures to combine the identity information of Top Level Domain. So, in Domain Name Space, each parent signs the Public Keys of all its Children in the DNS tree.

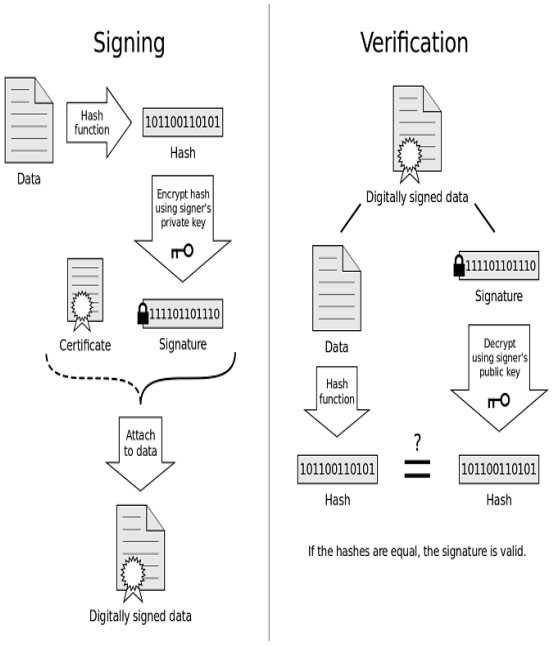
B. Securing DNS with ECC

With technology growing faster, everyone accesses the Internet through mobile phones. Whether it is used to check Emails or visit any secure sites, ECC (Elliptic Curve Cryptography) can be implemented. ECC provides the same level of Security as RSA[5] with benefits of small key sizes, faster computation, and memory and energy savings[6]. Small Key Size and Faster Computation: The security level of 160-bit ECC and 1024- bit RSA is the same. RSA operations are based on modular exponentiations of large integers, and security is based on factoring these large integers. On the other hand, ECC operations are based on groups of points over elliptic curves and security is based on discrete logarithm problems (ECDLP). This allows ECC to have the same level of security with smaller key sizes and higher computational efficiency. Memory and Energy savings: ECC requires less power for its functioning, so it is more suitable for low power applications such as handheld and mobile devices. On small processors, multiple precision multiplications of large integers (done in RSA) not only involves arithmetic operations but also a significant amount of data transport to and from memory due to limited registers space. While in ECC, the scalar multiplications involve additions with no intermediate results to be stored, thereby requiring less use of registers. So, ECC provides less memory space and also, the energy required to perform additions is much less than performing multiplications done in RSA.

Table 4.1 ECDSA vs RSA

PARAMETERS	RSA	ECDSA
Key Size	1024 bit length	192 bit length Smaller
Encryption	Fast	Slow
Decryption	Slow	Fast
Key Exchange	Fast	Slow
Signature Generation	Slow	Fast
Signature Verification	Fast	Slow

V. SYSTEM ARCHITECTURE



VI. ECDSA IMPLEMENTATION

The key parameters are taken as same as recommended by NIST, but we are introducing a change in the signing and verification process.

A. Key Parameters

Some predefined parameters for the ECDSA implementation used, as follows:

1. Select a prime number (p) of large size.
2. Choose constants (a and b) such that (4a3+27b2) modulo p is not equal to 0.
3. Generate elliptic curve points Ep (a, b), where Ep (a, b) is a generalized term for elliptic curve points (x, y).
4. Choose generator point (G) of order n, where an order is a number of points in the elliptic curve.
5. Select d such that 1 < d < n-1. This is used as a private key. These parameters are recommended by NIST for federal government use and include elliptic curves of various bit lengths (e.g., 192, 224, 256, 384, 521 etc.)[8].
6. Generate public key Q such that Q = d.G, where '.' Is point multiplication for ECDSA and is represented as G+G+G……d times which can be calculated using elliptic curve arithmetic.

B. Signature Generation

1. Select a random number k to be used only once. That is, for every new signature generation of a message, a new k is selected, such that 1 < k < n-1.
2. Generate (r, s) component of signature such that
a. k.G = (x, y) r = x modulo n if r = 0 then repeat 2 again
b. Calculate hash of message (M) whose signature is to be generated, i.e., e = h (M). c. s = d(r*k – e)-1modulo n // (modified)

C. Signature Verification

1. Calculate u1 = e*r-1 modulo n // (modified)
2. Calculate u2 = (r*s)-1 modulo n // (modified)
3. Calculate T = u1.G + u2.Q = (x1, y1), where '.' Is point multiplication and '+' is point addition and can be calculated using elliptic curve arithmetic.
4. Calculate v = x1 modulo n
5. If v = r, a signature is valid.

The above-proposed algorithm is a variant of the algorithms as described in, providing less complexity in signing.

VII. CONCLUSION

The purpose of this work is to show the simulation of how this software system works, but with the ECDSA algorithm implemented in it. ECDSA is fast at verifying the signatures, uses a small key size as compared to RSA, and also provides the same level of security as given by RSA. ECC is a growing field of the future. So, this work involves DNS security using ECC. ECC being very secure, smaller key sizes, less in power and memory consumption gives better security to small portable devices.

VIII. REFERENCES

[1] Hu Junru, The Improved Elliptic Curve Digital Signature Algorithm, International Conference on Electronic & Mechanical Engineering and Information Technology, IEEE, (2011).

[2] Casey Deccio, Jeff Sedayao and Krishna Kant, Prasant Mohapatra, Quantifying and Improving DNSSEC Availability', IEEE, (2011).

[3] Ghanmy Nabil, Khlif Naziha, Hardware implementation of Elliptic Curve Digital Signature Algorithm (ECDSA) on Koblitz Curves 8th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing, IEEE, (2012).

[4] A.Sakthivel, R. Nedunchezhian, Improved The Execution Speed of Ecdsa Over Gf(2 n) Algorithm for Concurrent Computation Journal of Theoretical and Applied Information Technology, (2013).

[5] Aqeel Khalique, Kuldip Singh, Sandeep Sood, Implementation of Elliptic Curve Digital Signature Algorithm, International Journal of Computer Applications (0975 – 8887), 2(2) (2010).

[6] Vivek Kapoor, Vivek Sonny Abraham, Ramesh Singh, Elliptic Curve Cryptography, ACM Ubiquity, 9(20) (2008).

[7] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang, High-speed high-security signatures, (2011).

[8] HONG Jingxin, A New Forward-Secure Digital Signature Scheme, IEEE, (2007).

[9] El hadj youssef wajih, Machhout Mohsen, A Secure Elliptic Curve Digital Signature Scheme for Embedded Devices, International Conference on Signals, Circuits and Systems, IEEE, (2008).

[10] Xue Sun, Mingping Xia, An Improved Proxy Signature Scheme Based on Elliptic Curve Cryptography, International Conference on Computer and Communications Security, IEEE, (2009).

[11] Jonathan Petit, Analysis of ECDSA Authentication Processing in VANETs, IEEE, (2009).

[12] Qingkuan Dong, Guozhen Xiao, A Subliminal-Free Variant of ECDSA Using Interactive Protocol, IEEE, (2010).

[13] Jalel Ben-Othman, Yesica Imelda Saavedra Benitez, A lightweight security scheme for HWMP protocol using Elliptic Curve Technique, 11th IEEE International Workshop on Wireless Local Networks, IEEE, (2011).

[14] M. Janagan, M. Devanathan, Area Compactness Architecture for Elliptic Curve Cryptography, International Conference on Pattern Recognition, Informatics and Medical Engineering, IEEE, (2012).

[15] Zhang Youqiao, Zhou Wuneng, An ECDSA Signature Scheme Designs for PBOC 2.0 Specifications, 9th International Conference on Fuzzy Systems and Knowledge Discovery,IEEE, (2012).