3rd National Conference on Green Technology and Science for Sustainable Development
© 2020 IJSRST | Volume 5 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
International Journal of Scientific Research in Science and Technology

DNS Security Using Cryptography

P. A. Deshmane[1], Nilima R. Bharambe[2], Shrutika R. Tarmale[2]
[1.]Lecturer, Department of Information Technology, Anuradha Engineering College,Chikhli, Maharashtra, India
priyanka.deshmane8218@gmail.com[1]
[2]Department of Information Technology, Anuradha Engineering College,Chikhli, Maharashtra, India

ABSTRACT

The Domain Name System (DNS) is an essential part of the internet on where function many other protocols rely. DNS is a protocol that resolves host-names to IP addresses over the internal DNS, being an open source, it is less source and it has no means of determining whether domain name or data comes from an authorized domain owner. This security weakness leaves the system vulnerable to a number of attacks such as DNS cache poisoning, DNS spoofing. Hence, there is a need of securing DNS. The digital signatures generated with public key algorithms have the advantage that anyone having the public key able to verify the Digital Signature. The public key is used for encryption and private key is use for decryption. Cryptographic algorithms (i.e. ECDSA) is used for security DNS. Now a days technology growing day by day so there is a need of same level of security along with smaller key sizes. Everyone accesses internet through mobile phones whether it is used to check E-mails or visiting other secure sites, ECDSA involving ECC (Elliptic Curve Cryptography) concepts having less key sizes as compared to RSA be implemented to provide security to DNS.

Keywords: DNS, ECC, ECDSA, ECDLP, Cryptography

I. INTRODUCTION

Domain Name System DNS, Domain Name System is a protocol that resolves host names to IP Addresses. Since, human can easily grasp host names but to learn IP Addresses in a large amount is a difficult task for human, which is therefore, is performed by DNS. The Domain Name System (DNS) is a vital part of the Internets infrastructure. It maps names to machine-readable information, e.g., mapping www.example.com to the IP address 93.184.216.34. The original DNS has a critical vulnerability, called cache poisoning t thousands of Internet users to malicious sites. The DNS Security Extensions (DNSSEC) were introduced to address this vulnerability. Using digital signatures, DNSSEC guarantees the authenticity and integrity of DNS data, thus thwarting attackers that seek to falsify DNS responses. However, while DNSSEC solves this flaw in the DNS, it introduces a serious new vulnerability. The digital signatures DNSSEC adds to the protocol make DNS responses much larger. This makes DNSSEC an attractive vector to abuse in amplification denial-of-service attacks. The root cause for why DNSSEC makes responses so much larger, is the use of RSA for digital signatures. Current official recommendations suggest using 2048-bit RSA keys. If followed, this means every signature added to a DNS response requires 256 octets. Earlier work showed there are attractive alternatives to RSA. In particular, Elliptic Curve Cryptography (ECC) offers better security with smaller signatures, with only minor drawbacks . Thus, use of ECC can reduce the attack potential in DNSSEC. Two algorithms, ECDSA P-256 and P-384, were standardized for use in DNSSEC in 2012.Yet when was published, in 2015, there was next to no adoption of these algorithms, with 99.99% of signed domains in .com, .net and .org still using RSA.

There is an increasing interest in use of ECC for DNSSEC .Finally, there has been active evangelization of the use of ECC in DNSSEC in recent conferences frequented by operators, such as ICANN meetings, the IETF and NANOG. This raises a question whether these efforts are paying off. In this paper we able to study the adoption of ECC by DNS operators. Our main contributions are that we provide the first detailed insight into operational adoption of a new cryptographic algorithm in DNSSEC[5].

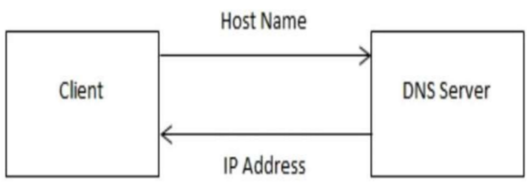

Fig 1. Basic DNS Functionality

show how adoption of ECC in DNSSEC grows, based on longitudinal data sets that span up to 1.5 years, show evidence of hurdles to deployment in our data sets, illustrate how adoption by a single operator can potentially be a game changer for the use of ECC in DNS.

II. DNS Security

A. Need Of Security

Since DNS is an open source system, there can be fakers who can change the information served by the DNS to the clients. As originally designed, DNS has no meaning of determining whether the domain name data comes from the authorized domain owner or it 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 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.

In DNS cache poisoning attack, an intruder replaces a valid IP address cached in a DNS table with a rogue address. Requests for the valid address are redirected accordingly, and malware (e.g., worm, spyware, browser hijacker etc.) .may be downloaded to the user's computer from the rogue location. DNSSEC employs cryptographic keys and digital signatures to ensure that lookup data is correct and that connections are legitimate servers.

A scenario for DNS Spoofing is shown in figure. where, when the client sends a request to the DNS server to know the IP Address, it sends its ID with the Query. By identifying that ID, an Attacker can send malicious data to the client or redirect the client to an unwanted site.

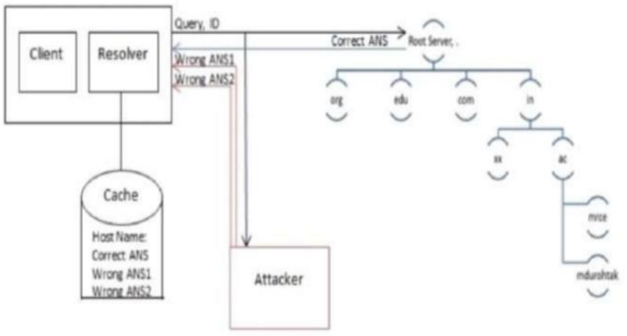

Fig 2. DNS Spoofing

III. CRYPTOGRAPHY

Cryptography involves creating written or generated codes that allow information to keep it secret [6]. Cryptography converts the data into a format that is unreadable for an unauthorized user and allowing it to be transmitted without unauthorized entities decoding it back into a readable format, thus compromising the data.

Information security uses cryptography on several levels. The information cannot able to read without a key to decrypt it. The information maintains its integrity during transit while being stored. Cryptography also aids in non-repudiation. This means that the sender and the delivery of a message can be verified. Cryptography is also known as cryptology.

Cryptography is the science of Secret Writing . The process of changing the plain information called the plain text into some sort of code called the cipher text is Encryption. The reverse of Encryption is Decryption. Cryptography includes various methods for providing security like Symmetric Key Cryptography, Public Key Cryptography and the Elliptic Curve Cryptography. Cryptography has been expanded to provide the following information security requirements:

- Non-repudiation: Preventing an entity from denying previous commitments and actions.
- Integrity: Ensuring no unauthorized alteration of data.
- Authentication: Verifying an entitys identity
- Confidentiality: Protecting the data from all but intended receiver.

IV. TECHNOLOGY USED

Through the use of key pairs Cryptography also allows senders and receivers to authenticate each other. There are various types of algorithms for encryption. There are some common algorithms include [1]:

B. *Symmetric Key Cryptography (SKC):*

There is only one key is used for both encryption and decryption. This type of encryption is also referred to as symmetric encryption [2].This type of cryptography has a common shared key for both encryption and decryption. The key must be shared between the two parties in a secure manner before starting exchanging the data. This can be easily implemented and is faster. The concept is well explained by the following diagram:

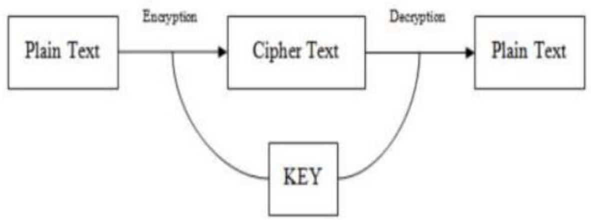

Fig 3. Symmetric key cryptography

C. *Public Key Cryptography (PKC):*

Here two keys are used. This type of encryption is called asymmetric encryption. One key is the public key that anyone can access .This requires a key pair to share the data. For non-repudiation, the sender encrypts plain text using a private key, while the receiver uses the senders public key to decrypt it. Thus, the receiver knows who sent it. The public key (known to all) is used for encryption and the private key (only with the user) is used for decryption. It requires larger resources to produce the key pair, so it is an slower algorithms. The advantage here is that no key has to be shared before starting the conversation. The concept is well explained by the following diagram

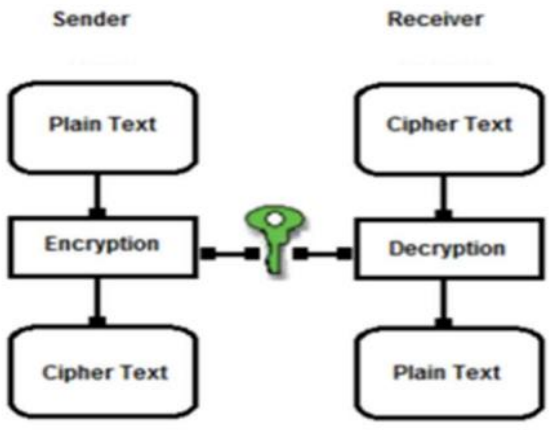

Fig 4. Public key cryptogrphy

D. *Elliptic Curve Cryptography*

Elliptic Curve Cryptography (ECC) [4] is similar to public key cryptography based on the concept of elliptic curves. Elliptic curves are basically cubic equations of two variables, with coefficients. ECC uses only those elliptic curves, wherein the variables and coefficients are restricted to elements of a finite field.

V. ELLIPTIC CURVE DISCRETE

The ECDLP is the basis for the security [8] and is based on the intractability of Scalar Multiplication products. Given points P and Q in the elliptic curve group, then find k such that, P.k = Q. k is a discrete logarithm of Q to the base P.

E.g.: Consider the elliptic curve y2=x3+5x+7 defined over F19 and let P=(6,4) and Q=(5,0). Then k.P = Q can be computed as:
P = (6, 4)
2P = (14, 16)
3P = (0, 2)
4P= (5, 8)
5P = (5, 0) = Q
So, k = 5

In reality the value of k would be large, making it infeasible to determine k. Given a point R = k*P, where R and P are known, then there is no way to find out what the value of k . There is no point of subtraction or point of division to resolve k = R/P. Also computing k requires 2n/2 operations. If the key size is 192 bits, then 296 operations are to be perform which would take millions of years. This thing where the multiplicand can t be found even when the original and destination points are known is the whole basis of the security behind the ECDSA algorithm, and the principle is called a trap door function or ECDLP.

VI. ECDSA

The elliptic curve digital signature algorithm is the elliptic curve analogue of DSA and also serves the same purposes of key generation, signature generation, and signature verification .The key parameters are taken as same as recommended by NIST but we are introducing a change in signing and verification process.[3]

E. *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 order is number of points in the elliptic curve.
5. Select d such that 1 < d< n-1. This is used as private. These parameters are recommended by NIST for federal government use and includes elliptic curves of various bit lengths (e.g., 192, 224, 256, 384, 521 etc.).
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.

F. *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 < n1.
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)

G. *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, signature is valid.

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

VII. COMPARISON OF ALGORITHMS

The complexity comparison of four ECDSAs is shown in table 2[1]. The four ECDSA are:-

1. Original ECDSA
2. ECDSA proposed by Hu Junru[1] (E-1)
3. ECDSA proposed by Hu Junru[1] (E-2)
4. ECDSA proposed implementation

Algorithm		Doubling O(n^2-logn)	Multiplication O (n^2)	Inverse O (n^2)	Total
ECDS A	SIGN	1	2	1	(logn+11)n 2
	VERIFY	2	2	1	(2logn+11) n^2
E-1	SIGN	1	2	1	(2logn+3)n 2
	VERIFY	2	2	1	(2logn+11) n^2
E-2	SIGN	1	2	1	(logn+11)n 2
	VERIFY	2	2	0	(2logn+2)n 2
PROPOSED IMPLEMENTATION	SIGN	1	1	1	(logn+2) n^2
	VERIFY	2	2	2	(2logn+4)n 2

Table 1.Algorithm Complexity Comparison

VIII. ADVANTAGES

DNSCrypt aims to protect the DNS traffic from not just 'sniffing' (passive) attacks but also active attacks - where attacker changes the DNS response and may be point it to the attackers server.

You say MITM is not possible. But it depends where in the process you think you are at risk. The VPN tunnel is protected (assuming the client is configured to route DNS through the VPN and not the default gateway). In that case you may be correct in assuming you are safe enough for your needs but Remember though the VPN terminator will (in this example) query the DNS for you may that be that could be a point of weakness[7].

IX. CONSLUSION

There are various security measures adopted in DNS using public key cryptography. As the technology growing day by day, there is a need of same level of security with smaller key sizes. ECDSA being fast at verifying the signatures. It uses small key size as compared to RSA and also, provides same level of security as given by RSA. Now, everyone uses mobile to retrieve data from internet and mobile being small and portable device needs security with less power consumption. This can be done with the help of ECC by implementing ECDSA in DNS.

X. REFERENCES

[1] William Stallings, Cryptography and Network Security, 4th Edition, Pearson Education, Inc., 2011

[2] https://simple.wikipedia.org/wiki/Symmetric key_algorithm

[3] Aqeel Khalique Kuldip Singh Sandeep Sood, Implementation of Elliptic Curve Digital Signature Algorithm , International Journal of Computer Applications (0975 8887), Volume 2 No.2, May 2010.

[4] Vivek Kapoor, Vivek Sonny Abraham, Ramesh Singh, "Elliptic Curve Cryptography , May 20-26, 2008. ACM Ubiquity, Volume 9, Issue 20.

[5] Preeti Thareja, Securing DNS using ECC, 2014 5th International Conference- Confluence The Next Generation Information Technology Summit (Confluence)

[6] https://www.techopedia.com/definition/1770 /cryptography

[7] https://security.stackexchange.com/questions /162601/what-are-the-privacy-advantages-ofa-dns-encryption-service-such-as-dnscrypt

Naveen Kumar Framework for Using Cryptography for DNS Security International Journal of Computer Science and Mobile Computing Vol.4 Issue 6,June-2015 .pg 882- 891