International Journal of Computer Applications (0975 – 8887)
Volume 120 – No.17, June 2015

Security System for DNS using Cryptography

Naveen Kumar Tiwari
M.Tech. (C.S.E)
Galgotia College Of engineering & Technology,
Greater Noida

Sanjay Khakhil
Assistant Professor
Galgotia College Of engineering
& Technology, Greater Noida

ABSTRACT
DNS, Domain Name System is a protocol that resolves hostnames to IP Addresses over the Internet. DNS, being an open source, it is less secure and it has no means of determining whether domain name data comes from an authorised domain owner. So, these vulnerabilities lead to a number of attacks, such as, cache poisoning, cache spoofing etc. Hence, there is a need of securing DNS. Digital Signatures are a good way of authyenticating the domain owners. The paper presents the Domain Name System security concept,. Digital Signature algorithms helps in providing good level of security to DNS. Software like OpenDNSSEC, BIND, Secure64 etc. It involves the signing of DNS using cryptographical algorithms (e.g., RSA, DSA etc.). Further, ECDSA is one way that provides same level of security, as security provided by RSA for low power and portable devices. So, here we proposing a new ECDSA implementation that can be used to secure DNS.

General Terms
Elliptic curve cryptography, Digital Signature Generation Algorithm

Keywords
DNS, RSA, ECDSA, ECDLP,DNSSEC, DSA and ECC.

1. INTRODUCTION
The Domain Name System is a protocol for locating domain names and mapping them to IP addresses. DNS is a hierarchical, distributed database, which provides mapping between easy to remember hostnames, such as www.uptu.ac.in, and IPv4 or IPv6 network addresses, for example, 117.211.115.134. In DNS tree, each node represents a DNS name. A DNS domain is a branch under the node. When a hostname is translated into its numeric representation, this allows the network to trace a path from a user to a particular server. Correct and timely DNS translations are vital for networks such as the Internet and thus are an interesting target for attackers. As originally designed, DNS has no means 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.

1.1 Elliptic Curve Cryptography
Elliptic Curve Cryptography (ECC) is a kind of 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.

1.2 Elliptic Curve Discrete Logarithm Problem (ECDLP)
The ECDLP is the basis for the security[3]. 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‘is. Since, there is no point subtraction or point division, to resolve k = R/P. Also, computing k requires roughly 2n/2 operations. If the key size is 192 bits, then 296 operations are to be done 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.

2. DNS BACKGROUND
The DNS system consists of following main components:

- Domain Name Space and resource records (RRs) which are used to identify hosts and extract its properties.
- Name servers having information on a subset of the domain tree.
- Resolvers or programs able to extract information from a name server after a client request and follow query referrals from one DNS server to another.
- Zones are certain portions of the DNS namespace. This portion is what for which the server is authoritative. An authority for server can be possible for one or more zones.
- Zone files are files that contain resource records about zones for which the server is authoritative. Zones are mostly implemented as text files in DNS implementations

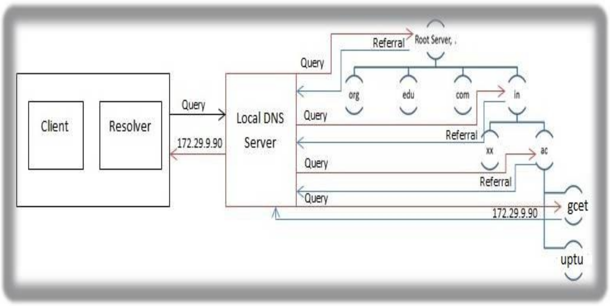

Fig . 1: DNS Working

Each host is identified by the name and resource information combined into RRs. RRs includes information such as owner of the domain, type of the database record and the Time to Live (TTL) value. DNS also includes a feature for one host to posses several names, this is done with help of a canonical name (CNAME) RRs. DNS messages can be carried over UDP or TCP. TCP version is mostly used for traversing stateful firewalls. DNS is also capable of performing inverse queries, which resolve an IP address into a DNS name, needed for some network-enabled applications.

Name servers store information about only a particular segment or zone of the DNS database. When a name server answers a query, it can use either a local database or reply with a referral to another server. DNS server containing all information about a zone is called an authoritative server for that zone. It is recommended that data on authoritative servers is replicated to secondary servers to ensure availability. The name server can also contain cached data from other DNS servers for records requested by the local resolver.

Resolver is an interface for programs to communicate with DNS servers. Resolver transforms subroutine calls into DNS requests and queries various DNS servers. Resolvers can reside either on a local PC or a DNS server. The latter option is called a stub resolver.

3. DNS SECURITY

3.1 Security Need

As originally designed, DNS has no means 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.

DNS Security Extensions (DNSSEC) is a set of IETF (Internet Engineering Task Force) standards which 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 the 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 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.

3.2 Securing DNS with ECC

With the technology growing faster everyone accesses internet through mobile phones whether it is used to check E-Mails or visiting any secure sites, ECC (Elliptic Curve Cryptography) can be implemented. ECC provides 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 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 problem (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 multiplication 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 energy required to perform additions is much less than performing multiplications, done in RSA.

Table 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

4. ECDSA IMPLEMENTATION

The key parameters are taken as same as recommended by NIST but we are introducing a change in 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 order is number of points in the elliptic curve.
5. Select d such that 1 < d < n-1. This is used as private key.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.)[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, signature is valid.

The above proposed algorithm is a variant of the algorithms as described in [1], providing less complexity in signing.

4.2 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

5. EXPERIMENTAL OUTCOMES

Here the experimental outcomes are listed in form of detailed table corresponding to traditional ECDSA algorithms and its operations.

Table 2 Algorithm Complexity Comparison

ALGORITHM		DOUBLING $O(n^2 \log n)$	MULTIPLICATION $O(n^2)$	INVERSE $O(n^2)$	TOTAL
ECDSA	SIGN	1	2	1	$(\log n + 11)n^2$
	VERIFY	2	2	1	$(2\log n + 11)n^2$
E-1	SIGN	1	2	1	$(2\log n + 3)n^2$
	VERIFY	2	2	1	$(2\log n + 11)n^2$
E-2	SIGN	1	2	1	$(\log n + 11)n^2$
	VERIFY	2	2	0	$(2\log n + 2)n^2$
PROPOSED IMPLEMENTATION	SIGN	1	1	1	$(\log n + 2)n^2$
	VERIFY	2	2	2	$(2\log n + 4)n^2$

5. CONCLUSION

The purpose of this work is to show the simulation of how these software system works, but with ECDSA algorithm implemented in it. ECDSA being fast at verifying the signatures and uses small key size as compared to RSA and also, provides same level of security as given by RSA. ECC is a growing field of 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 portable small devices.

6. ACKNOWLEDGEMENTS

I am extremely indebted to my guide Asst. Prof. Sanjay Kumar, Department of Information Technology, Galgotias College of Engineering & Technology, Greater Noida. I am very grateful to him for continual encouragement, motivation for literature, and continuous hours of sitting together and discussing the problems, which helped me to understand the subject and methodology to complete my Dissertation.

I would like to express me deep and sincere gratitude to Head of Department, Prof. (Dr.) Bhawna Mallick, Computer Science & Engineering, Galgotias College of Engineering & Technology, for her consultation, encouragement and personal guidance, which has provided me a good foundation for under taking to complete the Dissertation.

7. 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*, 10th April 2013.

[5] 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

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

[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 light weight 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, March 21-23,* 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 (FSKD 2012),* IEEE, 2012.

[16] Ravi Kishore Kodali, "Implementation of ECDSA in WSN", *International Conference on Control Communication and Computing (ICCC),* IEEE, 2013.

[17] Nabil GHANMY, Lamia CHAARI FOURA TI, Lotfi KAMOUN, "Enhancement security level and hardware implementation of ECDSA", IEEE, 2013.

[18] Soumya Basu, M.Pushpalatha, "Analysis of Energy Efficient ECC and TinySec Based Security Schemes in Wireless Sensor Networks", IEEE, 2013.

[19] Shweta Lamba, Monika Sharma, "An Efficient Elliptic Curve Digital Signature Algorithm (ECDSA)", *International Conference on Machine Intelligence Research and Advancement,* IEEE, 2013.

[20] Noura Ben Hadjy Youssef, Wajih El Hadi Youssef , Mohsen Machhout, Rached Tourki, "A Low-Resource 32-bit Datapath ECDSA Design for Embedded Applications", IEEE, 2014.