Volume 3, Issue 2, February 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Indepth Understanding Of Digital Signature Algorithm: A Case Study

Anjana S.Chandran
Research Scholar, School Of Computer Sciences,
Mahatma Gandhi University
Kottayam, India

Dr.Varghese Paul
Associate Professor, Department Of IT
CUSAT, Kalamassery
Kochi, India

Abstract – **This case study discusses on why Digital Signature Algorithm specifies that if the signature generation process result in a value of s=0, a new value of k should be generated and the signature should be recalculated .**

Keywords- ***Digital Signature Algorithm, Modular Arithmetic***

I. INTRODUCTION

The most important work from the work on public- key cryptography is the Digital Signature. The Digital Signature provides a set of security capabilities that would be difficult to implement in any other way. [1]
The National Institute of Standards and Technology (NIST) have published Federal Information Processing Standard FIPS 186, known as the Digital Signature Standard (DSS). DSS makes use of Secure Hash Algorithm (SHA) and presents Digital Signature Algorithm. [1] Which means DSS is the standard and DSA the algorithm. Security of the DSA largely depends on the difficulty of computing discrete logarithms. Motivation for this paper lies in the fact that DSA includes authentication function with additional capabilities. Apart from authenticating the message contents is to be verified by third party to resolve any dispute, DSA also verify author and date and time of signature. An in-depth study on the same is worth.
The key objective of this paper is to understand the algorithm and to discuss on why DSA insist on a new signature when signature generation process result in a value of s=0. This paper would discuss the concept of modular arithmetic as a base to the new readers of the topic and then follow to the description of Digital Signature Algorithm and then to the problem.

II MODULAR ARITHMETIC

It is a system for integers in which we can define a boundary for the values returned by it and wrap around the numbers upon reaching certain value called the modulus, to those values which are within it. [1][3][4][5]
For example x mod n divides x by n and returns the value of remainder. Hence the value will always be within n.

A. Digital Signature Algorithm [2]

DSA Key Generation

1. Choose a prime q that is 160- bit long. Choose an L-bit prime p, such that p=qk+1 for some integer k, 512<=L<=1024 and L is divisible by 64.
2. Generate g:choose h, where 1<h<p-1 such that g= h^k mod p > 1 and k=(p-1)/q
3. Secretly choose x by some random method , where 0<x<q and calculate y = g^x (mod p)
4. The public key is (p,q,g,y). The private key x must be kept secret. However (p,q,g,y) can be shared between different users of the system if desired . [1][2]

B. DSA Signature Creation

1. Select a random secret integer k where 0<k<q.
2. Calculate r=(g^k(mod p))(mod q)
3. Calculate s = (k^{-1}.SHA(M)+ x.r)(mod q) where SHA(M) is the SHA-1 hash function applied to the message M
4. The values of *r* and *s* shall be checked to determine if *r* = 0 or *s* = 0. If either *r* = 0 or *s* = 0, a new value of *k* shall be generated, and the signature shall be recalculated. It is extremely unlikely that *r* = 0 or *s* = 0 if signatures are generated properly.

Sends signature (r, s) with message M. [1] [2]

C. DSASignature verification :

1. Reject the signature if either $0<r<q$ or $0<s<q$ is not satisfied
2. Calculate $w= (s)^{-1} \bmod q$
3. Calculate u1= (SHA(M)*w)modq
4. Calculate u2= (r*w)modq
5. Calculate $v=((g^{u1}*y^{u2})\bmod p)\bmod q$

Signature is valid if v=r. [1][2]

III PROBLEM STATEMENT

After the DSA Key Generation process the signature is created. The final step of Signature Creation and the first step of the Signature verification insist that s should not be equal to zero. This paper focuses on this fact. [6][7][8][9][10]

IV DETAILED ANALYSIS

Let us see what happens if s=0.
From the step number 3 of DSA signature Creation we have

$$s = (k^{-1}.SHA(M) + x.r) \pmod q$$

This implies that either the term
$(k^{-1}.SHA(M)+ x.r) =0$
Or
$(k^{-1}.SHA(M)+ x.r)$is completely divisible by q .

Because after doing the mod q operation a zero is returned. We can in fact conclude that s is zero when $(k^{-1}.SHA(M)+ x.r)$divided by q.

Let us take the second possibility as the real cause for instance. If so it turns to be that x.r and $(k^{-1}.SHA(M)$ can be /could be predicted using trial and error method .And hence security may be compromised.

Now let us find what happens when r=0

From the step no2 of DSA Signature Creation by keeping r=0 we get
$r= (g^k \pmod p) \pmod q =0$

This implies that either $g^k=0$ or g^k is a multiple of p or $g^k \pmod p$ is a multiple of q .the value of g being taken greater than 1 can never be equal to zero . Hence the possibility is that it is a multiple of p or $g^k \pmod p$ is a multiple of q.

There is a threat here that if r equals to be zero s can be compromised too. Since s is a function depending on r from equation shown in step number 3 of DSA Signature Creation $s = (k^{-1}.SHA(M)+ x.r)\pmod q$, here if r=0 , M can be easily found(as it depends solely on Secure Hashing Algorithm alone) with k kept for guessing as q is already public.
If s and r equals zero then from the above given cases we can conclude that both r and M may be compromised. Both r and s being the signature authenticating the sender –if compromised might even lead to third party intervention.

V CONCLUSION

The integrity of message M and value of r may be compromised if on generation of signature of DSA we find s=0.Both r, s values are for authenticating the sender, which if compromised questions the integrity of the message M. Hence the recalculation of k need to be done and a new signature is to be calculated once we find such instance.

REFERENCES

[1] William Stallings, Cryptography And Network Security Principles And Practices
[2] Digital Signature Standard, Federal Information Processing Standards Publication 186, **FIPS PUB 186**
[3] Information Security Theory And Practice, Dhiren R. Patel, 2008 Edition
[4] Cryptography And Network Security, Atul Kahate, Second Edition
[5] Cryptography and Information security, V.K. Pachghare, 2009 Edition
[6] Security in Computing Charles P.Pfleeger , Shari Lawrence Pfleeger, Third Edition
[7]Principles And Practices Of Information Security, 2009
[8]Mark Stamp's Information Security, Principles and Practice, Deven N.Shah

[9]Computer Security Art and Science, Matt Bishop, 2003
[10]Hunting Security Bugs, Tom Gallagher, Bryan Jeffries, Lawrence Landaker, 2006
[11] Information Security Policies, Processes and Practice, 2008

