Implementing Swati Verma's Digital Signature Schemes based on Integer Factorization and Discrete Logarithms

A.B.Nimbalkar
A.M.College, Hadapsar, Pune 411028,
Maharashtra India

C.G.Desai
H.O.D. MIT Aurangabad,
Maharashtra, India

ABSTRACT

A digital signature is a cryptographic method for verifying the identity of an individual. It can be a process, computer system, or any other entity, in much the same way as a handwritten signature verifies the identity of a person. Digital signatures use the properties of public-key cryptography to produce pieces of information that verify the origin of the data. Several digital schemes have been proposed as on date based on factorization, discrete logarithm and elliptical curve. However, the Swati Verma and Birendra Kumar Sharma [8] digital signature scheme which combines factorization and discrete logarithm together making it difficult for solving two hard problems from the hackers point of view. This paper presents the implementation of same, with the help of different tools and further analyzes them from different perceptions.

Key words : Cryptography, Integer Factoring, Discrete Logarithm, Digital Signature.

1. INTRODUCTION

The security of most the digital signature algorithms are based on the difficulty of solving some hard theoretical problems. Digital signature algorithms are based on the concept of public key cryptography given by Diffie and Hellman [1]. Since then many public key cryptosystems are introduced, which are based on either prime factorization (FAC) or Discrete Logarithm (DL) problems [2].Although the schemes based on one of the above cryptosystem appears secure today, they may be unsecure in future. The security of the digital signatures can be enhanced by using factorization (FAC) or Discrete Logarithm (DL) problems, which are most commonly hard problems those can be used but not NP-complete. L. Harn [4] in 1995 showed that one can break the He-Kiesler[5] algorithm if one has the ability to solve the prime factorization.

Lee and Hwang [6] showed that if one has the ability to solve the discrete logarithms, one can break the He-Kiesler algorithm. Shimin Wei [7] showed that any attacker can forge the signature of He-Kiesler algorithm without solving any hard problem in 2002. Now, we implement the Swati Verma [8] signature scheme based on factorization and discrete logarithm problem both with different parameters and using a collision-free one-way hash function in this paper.

2. SWATI VERMA AND BIRENDRA KUMAR SHARMA SCHEME BASED ON INTEGER FACTORIZATION AND DISCRETE LOGARITHM BASED ALGORITHM.

2.1 INITIALZATION

Let's select the following parameters:

p: a large prime p = 4p1 q1+ 1, where p1 = $2p_2$ + 1, q1 = $2q_2$ + 1, and p1, q1, p_2, q_2 are all primes and let n = p1.q1.

g: an primitive element of Galois field GF(q).

h (.) : a collision-free one-way hash function.

Further, the user chooses a private key X Є Z_n such that gcd(X, n) = 1 and computes a corresponding public key which is certified by the certificate authority.

$$y = g^{x^2} \bmod p \quad \text{--(1)}$$

2.2 DIGITAL SIGNATURE GENERATION

To sign a message M, the signee carries out the following steps.

1. Randomly select an integer T Є Z_n such that gcd (T, n) = 1,
2. Computes $r_1 = g^{r^2} \bmod p$ --(2)

   And makes $r_2 = g^{r^{-2}} \bmod p$ --(3)
3. Find s such that

   $$h(r_1, r_2, m)T^{-1} = x_{r_1} + Ts^2 \bmod n \quad \text{--(4)}$$

   Where h is a collision-free one-way hash function defined by the system
4. (r1, r2, s) is a signature of message M. The signee then sends (r1, r2, s) to the verifier.

2.3 DIGITAL SIGNATURE VERIFICATION

On receiving the digital signature (r1 r2 s) the verifier can confirm the validity of the digital signature by the following equation,

$$r^{s^4} \cdot r_2^{h(r_1 \cdot r_2 \cdot m)^2} = y^{r^2} \cdot g^{2h(r_1 \cdot r_2 \cdot m)s^2}$$

-- (5)

If the equation holds, then (r1, r2, s) is a valid signature of message M.

2.4 IMPLEMENTATION

We implement the S.Verma and B. Sharma[8] scheme in Mathematica 9.0.

The code for the same is :

```
(* Program of Swati Verma's new Scheme *) .
p=RandomPrime[{50^40,50^42}];
Print["P is :",p];
a=Take[FactorInteger[p-1],-2];
p1=a[[1,1]];
q1=a[[2,1]];
n=p1 * q1;
Print["n is :",n];
g=PrimitiveRoot[p];
Print["g is :",g];
(*  Calculate x  *)
gd=0;
 x=RandomInteger[{2,n-1}];
While[gd != 1 && x>0 && x <n,
       x=x+1;
      gd=GCD[x,p-1];
]
Print["x is :",x];
y=PowerMod[g,Power[x,2],p];
Print["Y is :",y];

(* Do the loop till size of s1 is greater than zero *)
len=0;
While[len== 0,
(*  Calculate  t  *)
gd=0;
(*Randomly generate t"];*)
 t=RandomInteger[{3,n-1}];
While[gd != 1 && t>0 && t <n,
       t=t+1;
      gd=GCD[t,p-1];
]
Print["t",t];

pow1=Power[t,2];
pow=Mod[pow1,p-1];
powinv=PowerMod[pow,-1,p-1];

r1=PowerMod[g,pow,p];

r2=PowerMod[g,powinv,p];
 (* Find s *)
a1=PowerMod[t,-1,p-1];
b1=PowerMod[t,-2,p-1];
(* Take Messsage  *)
m=12345;
(*Message digest using SHA 256 algo *)
mHash=Hash[{r1,r2,m},"SHA256"];
c11=mHash*b1 - x*r1*a1;
c1=Mod[c11,p-1];
s1=PowerModList[c1,1/2,p-1];
len=Length[s1]
(* Print["if length is 0 regenerate the t, length is:
",Length[s1]]; *)
](* End of Repeated loop *)
Print["t is:",t];
Print["r1 is:",r1];
Print["r2 is:",r2];
 Print["message Digest :",mHash];
Print["c1 is:",c1];
s=First[s1];
Print["s is:",s];
(*If[s,Print["OK"]];
If[!IntegerQ[n]||n\[LessEqual]0,Break[]];*)
(* Varification of signature *)
(* To reduce the Overflow *)
```

```
k1=PowerMod[r1,s,p];

k2=PowerMod[k1,s,p];

k3=PowerMod[k2,s,p];

k4=PowerMod[k3,s,p];

(* k4  is r1^(s^4)   *)

l1=PowerMod[r2,mHash,p];

l2=PowerMod[l1,mHash,p];

z=k4*l2;

lhs= Mod[z,p];

Print["lhs is :",lhs];

g1=PowerMod[y,r1,p];

g2=PowerMod[g1,r1,p];

h1=PowerMod[g,2,p];

h2=PowerMod[h1,mHash,p];

h3=PowerMod[h2,s,p];

h4=PowerMod[h3,s,p];

u=g2*h4;

rhs=Mod[u, p];

Print["rhs is :",rhs];
```

After Execution of program. The Output is:

P is :
700860204563397141266162425059542467556443075505401155901381049004954 67

n is : 10462140406278983219380129466578735124976097 49

g is :2

x is : 519432761807905179856445795609649021122516111

Y is :
555749995386359465088580927268698828713643006538384875116761216479927 92

t is: 4709961571026807876289191848171810444033148 01

r1 is:
178213901577969366314991451837006962367491650374905110662382470333702 0

r2 is:
541807724452416808755055577168617826060501979571949282588812194860116 56

m is : 12345;

Message Digest:
114238795066522747971603047124097974728520512704022213991907064275570404935222

s is:
493485648355024060233815072826820216120627147658531038675611945895746 0

VERIFICATION

LHS IS :
224718380431333700738501912182266476906133948441074468429872966246005 49

RHS IS :
224718380431333700738501912182266476906133948441074468429872966246005 49

3. Conclusion:

For developing this algorithm based on integral factorization and discrete logarithms , the large prime numbers and large digit numbers are used. There are some difficulties for implementation of these algorithms in MuPAD and MatLab. As per the method adopted for implementing these algorithms, the MuPAD and Matlab7.1 do not support large digit numbers and some mathematical functions. So there is limitation for secure key generation.

Due to this limitation, the Mathematica 9.0 is used. It supports for large integer number and maximum mathematical functions.

Looking at the advantage supported by Mathematica9.0, the Swati Verma, Birendra Kumar Sharma‘s algorithm is implemented in Mathematcia9.0. However, it was observed that programming controls in Mathematica9.0 are easier as compared to MuPAD and MatLab.

The S.Verma and B. Sharma [8] algorithm makes signature hard to break due the Hash Function, for 5 digit message, it will produce near about 78 digits Message Digest. The value for ‘t’ is randomly generated based on condition that the g.c.d. of t and n should 1. In certain cases it takes more time to generate the value of ‘t’.

4. REFERENCES

[1] Diffie W. and Hellman M.E, “New directions in cryptography”, *IEEE Transactions on Information Theory*, 22, 644- 654, (1976).

[2] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. Information Theory, IEEE Transactions on,31(4):469- 472, 2002.

[3] Harn L., “Public-key cryptosystem design based on factoring and discrete logarithms”, *IEE Proceedings: Computers and Digital Techniques,* 141, 193-195, (1994).

[4] L. Harn. Comment: Enhancing the security of El Gamal’ssignature scheme. IEE Proceedings-Computers and Dig-ital Techniques, 142:376, 1995.

[5] He J. and Kiesler T., “Enhancing the security of ElGamal's signature schemes”, *IEE Proc. Comput. Digital Technol.* 141, 249-252, (1994).

[6] N.Y. Lee and T. Hwang. The security of He and Kiesler’ssignature schemes. In Computers and Digital Techniques,IEE Proceedings-, volume 142, pages 370-372. IET, 2002.

[7] S. Wei. A New Digital Signature Scheme Based on Factoring and Discrete Logarithms. Progress on Cryptography, pages 107-111, 2004

[8] 'A New Signature Scheme Based on Factoring and Discrete Logarithm Problems' Swati Verma*, Birendra Kumar Sharma, International Journal of Information & Network Security (IJINS) Vol.1, No.3, August 2012, pp. 158~162.