Goldbach's Conjecture on ECDSA Protocols

N Vijayarangan, S Kasilingam, Nitin Agarwal

Abstract - In this paper, an algorithm on Goldbach's conjecture is newly defined for computing a large even number as a sum of two primes or a sum of prime and composite. Using the conjecture, an ECDSA (Elliptic Curve Digital Signature Algorithm) protocol is newly proposed for authentication. The protocol describes the process of key generation, signature generation and signature verification as well as security issues.

Index Terms - Elliptic curves, Digital signature, Multi - precision integer, Goldbach's conjecture, ANSI X9.62

1. INTRODUCTION

Goldbach's original conjecture (sometimes called the "ternary" Goldbach conjecture), written in a June 7, 1742 letter to Euler, states that every integer is the sum of three primes [1]. As re-expressed by Euler, an equivalent of this conjecture (called the "strong" or "binary" Goldbach conjecture) asserts that all positive even can be expressed as the sum of two primes. Later [2,3], we could find any large even number into a sum of numbers p and q where p is prime and q may be either composite or prime. The prime p or q plays an important role in the proposed ECDSA protocols.

2. ELLIPTIC CURVES

In our application, we have taken reduced form of elliptic curves over prime. Let $p > 3$ be prime. The elliptic curve $y^2 = x^3 + ax + b$ over z_p is the set of solutions $(x,y) \varepsilon\ z_p \times z_p$ to the congruence $y^2 \equiv x^3 + ax + b \pmod p$, where $a,b\ \varepsilon\ z_p$ are constants such that $4a^3 + 27b^2 \neq 0 \pmod p$, together with a special point O called the point at infinity.

An elliptic curve E can be made into an abelian group by defining a suitable operation on its points. The operation is written additively, and is defined as follows (where all arithmetic operations are performed in z_p):

The authors are with the PKI Group, Core R&D, ITI Limited, Bangalore, India.
E-mail: {vijayrangan_2005, kasilingam_s, nitagl} @yahoo.com

Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are points on E. If $x_2 = x_1$ and $y_2 = -y_1$, then $P + Q = O$; otherwise $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$, and

$$\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad \text{if } P \neq Q$$

$$\lambda = \frac{(3x_1^2 + a)}{2y_1} \quad \text{if } P = Q.$$

Finally, define $P + Q = O = Q + P$ for all $P\ \varepsilon\ E$. With this definition of addition, it can be shown that E is an abelian group with identity element O.

Note that inverses are very easy to compute. The inverse of (x,y) (which we write as –(x,y) since the group operation is additive) is (x,-y) for all $(x,y)\ \varepsilon\ E$.

The following ECDSA protocol is based on quadratic residue and Goldbach's conjecture. It is different from ANSI standard [9,10] and computational complexities are involved in this method.

3. PARTITION ALGORITHM I

The following algorithm gives the sum of two primes for 32-bit even number n. This method lists out various combination of sum of two primes (partitions) for the same n. Let partition $=\{p_1, p_2\}$ and count = {number of partitions}.

1. Input n, count = 0.
2. Find a suitable k_s such that $6k_s + \varepsilon = n$ or $6k_s - \varepsilon = n$, where $\varepsilon > 0$.
3. Construct a sequence $S = \{1,2,3,6k\pm1\}$ where $k = 1$ to k_s.
4. Apply Miller-Rabin method to S to remove composite numbers.
   Then construct a sequence of primes $P = \{p_1,p_2,\ldots,p_m\}$.
5. For j = 1 to m
   5.1 Fix p_j

5.2 For i = 1 to m, $i \neq j$
5.2.1 If $p_j + p_i = n$, then count = count + 1, partition = { $p_{t,} p_i$ }. Break.
5.3 Next j

The following C program is given below for partition algorithm I. Only important functions are mentioned, not complete code.

```
/* The following function miller()performs
the Miller-Rabin Test to filter out
composite numbers
 Input to the function is the given number
n.
*/
int miller(unsigned long n)
{

   unsigned long r;
   unsigned long a = 2;
   unsigned int s=0;

   if(n==2)
   return 1;
   else if(n==1)
      return 0;

   if(!(n&1))
   {
      if(n==0)
      {
         printf("\n 0 is not prime\n");
         return 0;
      }
      printf("\n number n is even it cannot
be prime.\n");
      return 0;
   }

   r=n-1;

   while(!(r&1))
   {
      r>>=1;
      s+=1;
   }

   if(IsPrime(r,n,s)==1)
   return 1;

   //printf("\n program is in progress");
   return 0;

}

/* The following IsPrime function supports
Miller-Rabin test */

   Fo a base a, m is computed from n-1 =
(2^s)*m. Then test the condition (a^m)mod n
not equal to 1. If so, then declare that n
is composite.*/

int IsPrime(unsigned long r, unsigned long
n, unsigned int s)
{
   unsigned long a = 2;
   unsigned long k=1;
   unsigned long i;
   unsigned int j;

   for(i=0; i<r;i++)
   {
      k=(k*a)%n;
   }
   k=k%n;

   if(k==1 || k==(n-1))
      return 1;
   else
   {
      j = 1;

      while((j<s) && (k!=n-1))
      {
         k= ((k%n)*(k%n))%n;
         if(k==1)
            return 0;
         j+=1;
      }
      if(k!=(n-1))
         return 0;
   }
   return 1;
}

/* primes()splits n into p1 + p2 from which
first summand p1 is taken.
(Goldbach's conjecture), where p1 is the
first prime and
 p2 the second */

unsigned long primes(unsigned long n)
{
unsigned long k;
unsigned long k1;
unsigned long t;
k=n;
t=(k-1)/6;

while(!miller(6*t+1))
{
   t-=1;
   k=6*t+1;
   k1=6*t-1;
   if(miller(k1))
      return (k1);
}
return(6*t+1);
}

/* The following function verifies p2 is
prime or composite. This is an optional
case to test p2 */

unsigned long primeadd(unsigned long n,
unsigned long prime1)
{
   unsigned long k;

   k=n-prime1;

   if(miller(k))
      return(k);
   else
      return 0;
}
```

In 32-bit, any even number can be expressed as a sum of two primes. Whereas in multi-precision integers (MPI), we are not sure. So, we can represent an MPI even number into a sum of two primes or a sum of prime and composite.

The following algorithm is for multi-precision integer to compute Goldbach's conjecture.

4. PARTITION ALGORITHM II

1. Input n, count = 0.
2. Find a suitable k_s such that $6\ k_s + \varepsilon = n$ or $6\ k_s - \varepsilon = n$, where $\varepsilon > 0$.
3. Construct a sequence $S = \{1,2,3,6k\pm1\}$ where $k = 1$ to k_s.
4. Make S into disjoint sets $S_1 \cup S_2 \cup \ldots \cup S_m$.
4. For each S_j, perform step 5 (as in Partition Algorithm I) in parallel processing
6. Find $\{n = p_k + p_r$, count$\}$.

Algorithm II computes any even number into a sum of two primes or a sum of prime and composite. In each S_j consists of collection of primes and odd composites and passes into step 5 of Algorithm I. These computations should be done in parallel processing setup.

Using algorithm I or II, we have proposed the following ECDSA protocols.

5. ECDSA PROTOCOL BASED ON GOLDBACH's BINARY CONJECTURE METHOD

Key Generation

E is an elliptic curve defined over z_p, and P is a point of prime order n on the curve E; these are system-wide parameters.

Each sender (A) has to do the following procedures:

1. Select a random integer d in the interval [1, n-1].
2. Compute Q = d P.
3. A's public key is Q; A's private key is d.

The following protocols are proposed for ECDSA in the process of signing and verification:

Signature Generation

1. Select a random integer k_1 in the interval [1, n-1]
2. Compute $k_1\ P = (x_1, y_1)$ and $r_1 = x_1 \pmod n$ (where x_1 is regarded as an integer between 0 and p-1).
3. Compute $k_1^{-1} \pmod n$.
4. Compute $s_1 = k_1^{-1}\{\ h(m) + d\ r_1\ \} \pmod n$. If $s_1 = 0$, then go back to step 1. (where h is the secure hash algorithm).
5. The signature for the message m is the pair of integers (r_1, s_1).
6. Choose k_2 as prime from Goldbach's conjecture i.e., k_2 is chosen p_1 or p_2 from the equation int $(k_1/2) = p_1 + p_2$ where p_1 (prime), p_2 (prime or composite) and int( ) returns integral part. It is our choice to choose prime p_1 or p_2. Assume that $p_1 > p_2$.
7. Select $k_2 = p_2$ and compute $k_2\ P = (x_2, y_2)$ and $r_2 = x_2 \pmod{p_1}$ (where x_2 is regarded as an integer between 0 and p_1-1).
8. Compute $k_2^{-1} \pmod{p_1}$.
9. Compute $s_2 = k_2^{-1}\{\ h(m) + d\ r_2\ \} \pmod{p_1}$. If $s_2 = 0$, then go back to step 1.
10. The signature for the message m is the pair of integers.
11. Compute $(r,s) = ((r_1 + r_2) \bmod n, (s_1 + s_2) \bmod n)$ and send this pair to the receiver (B).

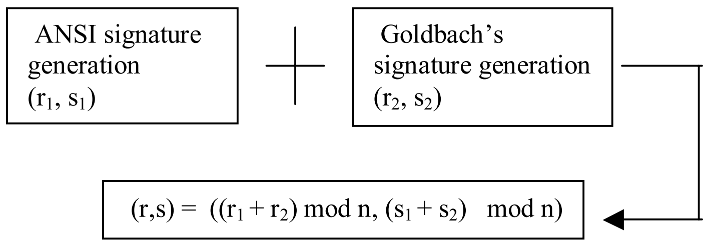

Fig.1: ANSI-Goldbach Signature generation

Signature Verification

To verify A's signature (r,s) on m, B should do the following:

1. Take the public key Q.
2. Compute $w_1 = s_1^{-1} \pmod n$ and h(m).
3. Compute $u_1 = (h(m)\ w_1) \bmod n$ and $u_2 = r_1 w_1 \pmod n$.
4. Compute $(u_1\ P + u_2\ Q) = (x_\alpha, y_\alpha)$ and $v_1 = x_\alpha \pmod n$.
5. Get the prime p_1 from the sender (A).
6. Compute $w_2 = s_2^{-1} \pmod{p_1}$.
7. Compute $u_3 = (h(m)\ w_2) \bmod p_1$ and $u_4 = r_2\ w_2 \pmod{p_1}$.
8. Compute $(u_3\ P + u_4\ Q) = (x_\beta, y_\beta)$ and $v_2 = x_\beta \pmod{p_1}$.
9. Compute $v = (v_1 + v_2) \bmod n$.
10. Accept the signature if and only if $v = r$.

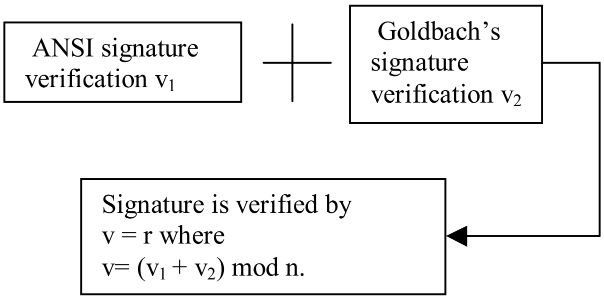

Fig.2: ANSI-Goldbach signature verification

In the above protocol, we have computed two pairs (r_1, s_1) and (r_2, s_2) where (r_1, s_1) follows from ANSI standard ECDSA protocol and (r_2, s_2) follows from Goldbach's binary conjecture method. Then the signature is verified by taking $(v_1 + v_2) \bmod n \equiv (r_1 + r_2) \bmod n$. The above authentication scheme also performs well if we set Goldbach signature generation and verification alone (called Goldbach ECDSA). Hence we have two new protocols: ANSI-Goldbach ECDSA and Goldbach ECDSA. Security on these protocols is discussed in due course.

Example: Let us take an elliptic curve E: $y^2 = x^3 + ax + b \pmod p$, where a = 1, b=1, p = 11. Take an elliptic point P = (2,7) of order n = 13 and set a public key Q = 10P = (8,8). First, we choose $k_1 = 11 \varepsilon [1, n]$. Then choose a prime number $k_2 = 2$ from the function int $(k_1/2) = p_1 + p_2$ (where $p_1 = 3$ and $p_2 = 2$). Proceed as in the above ECDSA procedure, we can easily verify the signature for any message.

Note: It is also tested that ANSI-Goldbach ECDSA and Goldbach ECDSA perform well when chosen number for p_2 is composite.

6. SECURITY

In ANSI-Goldbach ECDSA protocol, the security lies on finding out k_1 and k_2. It is practically difficult to find a prime or composite number taken for k_2, since many partitions on Goldbach's binary conjecture are available for each k_2.
The basis for the security of ANSI ECDSA is the apparent intractability of the elliptic curve discrete logarithm problem (ECDLP): given an elliptic curve E defined over z_p, a point P ε $E(z_p)$ of order n, and a point R ε $E(z_p)$, determine the integer k ε [0, n-1], such that R = k P. The Pollard-rho algorithm reduces the determination of k to modulo each of the prime factors of n. The computing power required to compute ECDLP with the Pollard–rho method is $(\Pi\ n/2)^{0.5}$ for every n-bit. Whereas, in our protocol, the computing ECDLP takes $(\Pi\ n/2)^{0.5} + \delta$ for every n-bit (where δ depends on the size of k_2). So, the computational effort in finding discrete logarithm problem in the ANSI-Goldbach ECDSA protocol is more.

Time analysis is taken on signing and verification process of ANSI-Goldbach and Goldbach ECDSA. It is computed in Pentium III 700 MHz. See Table 1 & 2.

Table 1 : ANSI-Goldbach ECDSA

Bit length	Signature generation mSec	Signature verification mSec
192	100	60
512	700	550
1024	4000	2800

Table 2 : Goldbach ECDSA

Bit length	Signature generation mSec	Signature verification mSec
192	70	37
512	580	400
1024	3000	1800

7. CONCLUSION

We have shown that ANSI-Goldbach authentication scheme is more secure than ANSI ECDSA. Practically, it is being proved that ANSI ECDSA and Goldbach ECDSA are almost equivalent (why?). The reason is that ANSI ECDSA is based on the (large) random value k ε [1, n-1] whereas in Goldbach ECDSA, chosen k is prime or composite which is obtained from Goldbach's conjecture. Hence finding out k becomes ECDLP in both cases.

8. REFERENCES

[1] Caldwell, C. K. "Prime Links++." http://primes.utm.edu/links/theory/conjectures/Goldbach/

[2] Chen, J.R, "On the representation of a large even integer as the sum of a prime and the product of at most two primes". [Chinese] J. Kexuse Tongbao 17 (1966), 385-386.

[3] Chen, J.R: "On the representation of a large even integer as the sum a prime and the product of at most two primes", Sci. Sinica 16(1973) 157-176.

[4] Douglas R. Stinson, "Cryptography : Theory and Practice", *CRC press*, 1995.

[5] Eric Bach and Jeffrey Shallit, "Algorithmic Number Theory", *The MIT Press*, 1996.

[6] L.E. Dickson, "Goldbach's Theorem in: History of the theory of numbers" Washington (1919), pp.421- 424.

[7] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, Vol.48 (1989) pp. 139-150.

[8] N. Koblitz, "Algebraic Aspects of Cryptography", *Springer-Verlag* (1998).

[9] N. Koblitz, "Towards a quarter-century of public key cryptography", *Kluwer Academic Publishers* (2000).

[10] National Institute for Standards and Technology, "Digital signature standard", *FIPS Publication* 186 (1993).

[11] J. Richstein, in: Bosma, W. (Ed.), "Computing the Number of Goldbach Partitions up to $5x10^8$", *Proceedings of the 4th Symposium on Algorithmic Number Theory*, Leiden, The Netherlands, July 2-7, 2000 (Springer LNCS 1838)

[12] J. Richstein, "Verifying Goldbach's conjecture up to $4x10^{14}$", *Math. Comp.*, to appear (Institute of Informatics, Germany).

[13] Y. Saouter, "Checking the odd Goldbach conjecture up to" , *Math. Comp.* 67 (1998) 863-866.