IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-3 E-ISSN: 2347-2693

Embedding more security in digital signature system by using combination of public key cryptography and secret sharing scheme

Surbhi Sharma

Department of Computer Science,
JSS Academy of Technical Education, Noida, India

www.ijcseonline.org

Received: Mar/02/2016 Revised: Mar/10/2016 Accepted: Mar/24/2016 Published: Mar/31/2016

***Abstract*—** The digital signature system is more powerful than the authentication systems, because the digital signatures support both authentication and non-repudiation. Non-repudiation is necessary for the applications like electronic commerce, digital cash, banking applications, electronic voting etc. Here an efficient scheme based on RSA and arbitrary – precision numbers is discussed which is very secure. Also a secret sharing scheme is discussed which is based on visual cryptography. If the private key of users are compromised then intruder can use it illegitimately for signing purpose. So here the ways has been proposed to increase security of digital signatures by using combination of public key cryptography and secret sharing technique.

***Keywords*—**Digital Signatures; Arbitrary precision numbers; Secret sharing scheme using visual cryptography

I. INTRODUCTION

The power of public key cryptography is undeniable. The choices are public key encryption schemes, digital signature schemes, identification schemes and many more. These schemes are with different functional properties and security. In public key cryptography, the keys can be used for encryption and digital signatures purposes. Public key based authentication and digital signature protocols are standardized by International Telecommunications Union (ITU) standard recommendation X.509, The Directory-Authentication Framework [1]. Here we are dealing with digital signature scheme. The digital signature system are more powerful than the authentication systems, because the digital signatures support both authentication and non-repudiation [2]. Non-repudiation is necessary for the applications like electronic commerce, digital cash, banking applications, electronic voting etc.

An efficient digital signature system should be based on public key cryptography because otherwise all the messages should pass through the trusted server and this is an enormous overhead. The X.509 recommendation has proposed good public key based solutions to digital signature problem.

Here an efficient scheme based on RSA and arbitrary-precision numbers is discussed which is very secure.

Also in digital signature protocols, there is an assumption that Certification Authorities are trusted such that they never fail and nobody can compromise them.

Corresponding Author: *Surbhi Sharma*
Department of Computer Science, JSS Academy of Technical Education,, India

There are many problems with Trusted Third Parties (or Certifier Authority). Firstly there is always a chance of failure of server of certifier authority or failure of communication links. This makes issue of new certificates impossible. Secondly there is always a chance for intruder to block the operation by compromising private keys of either certifier or the ones certifier issues for users. An intruder can create valid but spoofing certificates by getting private key of certifier and can impersonate any user which trusts the compromised certifier. If the private key of users are compromised then intruder can use it illegitimately for signing purpose. So here the ways has been proposed to increase security of digital signatures by using combination of public key cryptography and secret sharing technique.

II. KEY GENERATION FOR PROPOSED SYSTEM

There are various steps involved in key generation like choosing cryptographic primitive, choosing size of key and preparing an efficient and secured scheme for generating key, signing and verification of signatures.

A. Choosing Size of the Key

Public key schemes have one design which can be used with keys of any length. It is up to the user rather than designer to determine an appropriate key length for use with scheme. The most common mechanism for determining an appropriate key length is to consult a standard [3]. There are several standards like NIST (2007) and ECRYPT-II (2010).

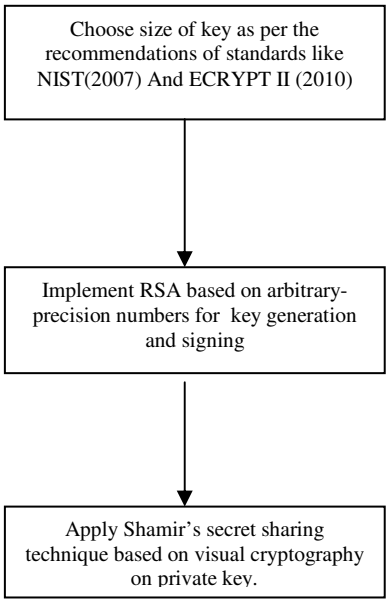

Figure 1- Flow diagram for making digital signature system more secure

These standards give key lengths in terms of the amount of time that the key will "work"- i.e. the amount of time that that the data secured by a key of this length can be considered secure. Key sizes are recommended by comparing them to symmetric keys with equivalent security and / or by giving estimate for length of time that the key will keep data secure.

A *cryptographic scheme* is a set of practical algorithms that can be used to solve some security problem and a *cryptographic primitive* is a basic piece of mathematics on which cryptography can be based, but which cannot be used as cryptographic scheme on its own.

To determine secure parameters for cryptographic scheme there is a need to determine secure parameters for underlying primitive. A few basic primitives are RSA based, Discrete logarithm based and Elliptic curve based [4].

For the proposed work RSA based (or factoring based) primitive is considered. For RSA, key size recommended by two standards is given below in the Table 1 as given in [3].

Table 1- NIST (2007) and ECRYPT (2010) recommendations for key size for RSA primitive

Equivalent symmetric key size	ECRYPT-II (2010) Recommendations		NIST (2007) Recommendations	
	RSA based modulus size	Validity	RSA based modulus size	Validity
80	1248	Up to 2014	1024	Up to 2010
112	2423	Up to 2020	2048	Up to 2030
128	3248	Up to 2040	3072	Beyond 2030
192	7936	-	7680	-
256	15424	Foreseeable future	15360	-

B. RSA Scheme based on Arbitrary Precision Numbers

RSA is proposed cryptosystem in by Rivest, Shamir and Adleman [5]. We can use this system as a signature scheme.

RSA key generation. The signer chooses two large secret primes p and q, and calculates public modulus n as n = pq, selects number e, such that $0 < e < \varphi(n)$ and e is relatively prime to $\varphi(n)$. Function $\varphi(.)$ is Euler's totient function.
There exists an inverse d of e modulo $\varphi(n)$, i.e. $d = e^{-1} \bmod \varphi(n)$.The party's public key is e and its private key d.
RSA signing. $s = m^d \bmod n$
Where s is the signature on m, m is the massage to be signed.
RSA verification. To verify that s is really the signer's signature on m, we verify if
$m = s^e \bmod n$ = yes or no
If the result is yes then s is the signer's signature on m.
RSA Problem : An attacker can factor n into p and q, and computes $(p - 1)(q - 1)$ which allows the determination of d from e.
With the ability to recover prime factors, an attacker can compute the secret exponent d from a public key (n, e). To deal with this problem n should be taken very large. Factoring large numbers seems to be much more difficult than determining whether it is prime or composite. In practice, RSA keys are typically 1024 to 4096 bits long. Some experts believe that 1024-bit keys may become breakable in the near future ; few see any way that 4096-bit keys could be broken in the foreseeable future. Therefore, it is generally presumed that RSA is secure if n is sufficiently large.
Difficulty with implementing a scheme with large prime numbers is fixed-precision arithmetic found in most Arithmetic Logical Unit (ALU) hardware, which typically offers between 8 and 64 bits of precision. So there is a need of numbers with larger precision.
Arbitrary-precision numbers (also known as bigintegers) indicate the calculations are performed on numbers whose digits of precision are limited only by available memory of host computer. Several modern programming languages have support for these numbers. Rather than store values as a fixed number of binary bits related to the size of the processor register, these implementations typically use variable-length array of

digits. The implementation of asymmetrical cryptographic schemes often requires the use of numbers that are many times larger than the integer data types that are supported natively by the compiler. In general, a positive number can be written as a polynomial of order k, where k is non-negative and $a_k \neq 0$.

$n = a_k b^k + a_{k-1} b^{k-1} + ... + a_1 b^1 + a_0$

b represents the base with 0 <= a <= b-1.

For base 10 representation, b = 10 and the digits allowed at each position is 0 <= a <= 9. When we write the number 12345 in base 10, it actually means

$12345_{10} = 1x10^4 + 2x10^3 + 3x10^2 + 4x10^1 + 5x10^0$

Similarly, if the same number is specified in base 16, then

$12345_{16} = 1x16^4 + 2x16^3 + 3x16^2 + 4x16^1 + 5x16^0$

Hence, it is important to know the base that the number is specified in, since the same representation could represent different values when interpreted in different bases.
In our BigInteger implementation, we store the number as an array of consecutive bytes. This is equivalent to the representation of the number in base 256 with each digit having values 0 to FF. A numerical example that shows the addition of two numbers in this base is given below

```
  1    1
 FE   A6 FF
 +    FF FF
 ----------
 FF   A6 FE
 ----------
```

In this example, FF + FF = 1FE. This exceeds the maximum value of each digit in base 256. Hence, we retain FE in the current position and carry 1 to the next higher position. Following steps are used:

```
1. carry = 0;
2. for(int i = 0; i < len; i++)
3. sum = bi1.data[i] + bi2.data[i] + carry;
4. carry = sum >> 8;
5. result.data[i] = (byte)(sum & 0xFF);
   // bi1 and bi2 are the two large numbers to add.
   data is the byte array holding the large number.
```

Primality testing is required for determining whether an input number is prime or not. The difference between primality testing and integer factorization is that primality test does not necessarily give prime factors. It only states whether input number is prime or not. Most popular primality test are probabilistic tests [6]. These tests use, apart from tested number n, some other numbers a which are chosen at random from some sample space . Following are the steps involved in Miller-Rabin probabilistic test:

Let RBG be an approved random bit generator.

Input: 1. w The odd integer to be tested for primality. This will be either p or q, or one of the auxiliary primes p_1, p_2, q_1 or q_2.

2. iterations The number of iterations of the test to be performed;

Output: status The status returned from the validation procedure, where status is either PROBABLY PRIME or COMPOSITE

Process:

1. Let a be the largest integer such that 2^a divides w−1.

2. m = $(w-1) / 2^a$.

3. wlen = len (w).

4. For i = 1 to iterations do

4.1 Obtain a string b of wlen bits from an RBG.

Comment: Ensure that 1 < b < w−1.

4.2 If (($b \leq 1$) or ($b \geq w-1$)), then go to step 4.1.

4.3 $z = b^m \bmod w$.

4.4 If (($z = 1$) or ($z = w - 1$)), then go to step 4.7.

4.5 For j = 1 to $a - 1$ do.

4.5.1 $z = z^2 \bmod w$.

4.5.2 If ($z = w-1$), then go to step 4.7.

4.5.3 If ($z = 1$), then go to step 4.6.

4.6 Return COMPOSITE.

4.7 Continue

. Comment: Increment i for the do-loop in step 4.

5. Return PROBABLY PRIME

III. USING SECRET SHARING SCHEME FOR KEY SAFETY

Secret sharing scheme is defined as a technique in which a dealer distributes some shares of secret among a set of participants such that some authorized subsets of them can recover the secret, while others cannot [7,8]. The first such scheme was developed by Shamir in 1979 .This scheme is

based on polynomial interpolation. Following is the algorithm as discussed in [8].

1. Choose a prime p, p > max(D,n),where D belongs to Z_p is the secret.
2. Choose k-1 random numbers $a_1,a_2\ldots$, a_{k-1},uniformly and independently from the field Z_p.
3. Using a_i,$1 \leq i \leq (k-1)$ and secret D, generate polynomial f(x) of degree k-1 , $f(x) = D + a_1x + a_2x^2 + \ldots + a_{k-1}x^{k-1} (\text{mod } p)$.
4. Sample f(x) at n points $D_i = f(i)$,$1 \leq i \leq n$ such that the shares are given by(i,D_i).

Reconstruction of the secret is performed by interpolating any k points(shares) and evaluating D=f(0).

A. Problem with Text based Secret Sharing Scheme

Firstly considerably huge amount of time is required for distribution and reconstruction of secret [9]. Also there is a need of distributor that can generate and distribute secret. Moreover in secret sharing systems, the shareholders should reveal their shares to commonly known combiner in order to restore secret. The combiner reconstructs the secret and naturally learns it. These situations can pose a threat to security. The above problems can be reduced to some extent by using secret sharing scheme based on visual cryptography. In this scheme no extra computation is required for reconstruction of secret. And there is no need of trusted combiner.

B. Using Secret Sharing Scheme Based on Visual Cryptography

The scheme works on the principle that the message consists of a collection of black and white pixels and each pixel is handled separately [10].

Each original pixel appears in n modified versions (shares), one for each transparency. Each share is a collection of m black and white subpixels, which are printed in close proximity to each other so that human visual system averages their black/ white contributions. This is also a threshold scheme, just like its counterparts which are text based, where out of n shares k are required for reconstruction. But we have implemented k out of k scheme where all shares are required for reconstruction as it also enhances security. The resulting structure can be described by n×m Boolean matrix $S = [s_{ij}]$ where $s_{ij} = 1$ iff the j^{th} subpixel in i^{th} transparency is black. When transparencies $i_1,i_2\ldots i_r$ are stacked together in a way that properly aligns the subpixel,we see a combined share whose black subpixels are represented by Boolean "or" of rows $i_1,i_2..i_r$ in S. Following are important parameters:

- m – The number of pixels in a share. This represents loss in resolution from the original picture
- r – The size of collection of two matrices
- α – The relative difference in weight between combined shares that come from white pixel and black pixel in original picture.

Construction of k out of k scheme

Following construction is used in proposed work:

1. Make use of two lists of vectors $J1^0,J2^0\ldots Jk^0$ and $J1^1,J2^1..JK^1$ and name them A and B.
2. In A and B vectors of length k over GF(2)with the property that every k-1 of them are linearly independent over GF(2), but the set of k vectors is not independent.
3. Each list defines a $k\times 2^k$ matrix S_t for value of t is either 0 or 1.
4. The parameters are defined as $m = 2^k$, $\alpha = 1/2^k$ and $r = 2^k!$

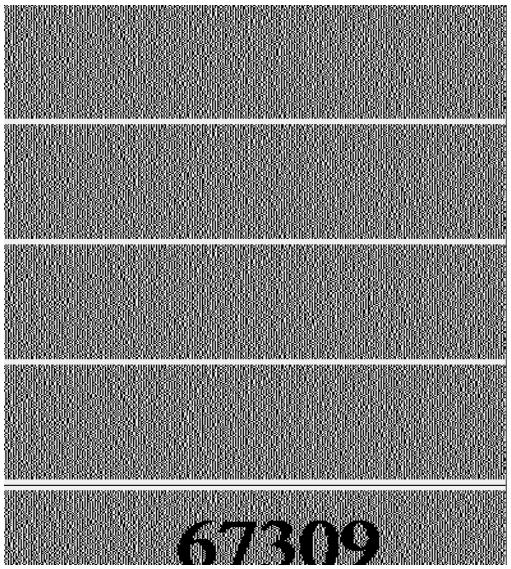

Figure 2- Images used for secret sharing

IV. CONCLUSION

The proposed digital signature system is very secure as public key cryptography is based on arbitrary- precision numbers that helps to provide more precision to numbers while doing calculations. Also the private key is made secure by using secret sharing technique based on visual cryptography. Only when the set of images is stacked together secret can be reconstructed. To add more security all the images can be stored in different places like some are in mobile and some are in Computer .

REFERENCES

[1] ITU-T Recommendation X.509, ISO/IEC 9594-8, Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, **1993.**

[2] S.R Subramanyam and Byung K.Yi, "Digital Signatures", IEEE, April **2006.**

[3] Alexander W.Dent , "Choosing key sizes for cryptography", Elsevier Ltd. , University Of London, Royal Holloway, UK, **2010**, pp.**21-27**.

[4] William Stallings "Network Security Essentials (Applications and Standards)", Pearson Education,**2004.**

[5] *R.L.* Rivest ,A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM ,**1978**, pp. **120–126.**

[6] Digital security standard, federal information processing standard publication, available at *http://dx.doi.org/10.6028/nist.fips.186-4,issue July **2013***.

[7] Kai Wang et al.,"A multiple secret sharing scheme based on matrix projection",Proceedings of 33rd annual IEEE international computer software and applications conference,U.S.A,**2009**,pp.**400-405**.

[8] A.Shamir,"How to share a secret",Communication of ACM,vol.22,**1979**,pp-**612-613.**

[9] Li Bai and Xukai Zou,"A proactive secret sharing scheme in matrix projection method",International Journal of Security and Networks,vol.4,**2009**,pp.**201-209**.

[10] Moni Naor and Adi Shamir ,Visual Criptography, Eurocrypt 94.