Volume 2, Issue 11, November 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Digital Signature Algorithm based on x^{th} Root Problem

Kamal kumar Agrawal* *M.Tech Student SBCET Jaipur, India*

Ruchi Patira *Assistant ProfessorSBCET Jaipur, India*

Kapil Madhur *M.Tech Student SBCET Jaipur, India*

Abstract- Generally, digital signature algorithms are based on NP-Complete problems like prime factorization problem, discrete logarithm problem, elliptic curve problem etc. The difficulty of RSA digital signature algorithm is based on prime factorization problem. The well known e^{th} ***root problem is proved equivalent to prime factorization problem. This paper introduces a new hard number theoretic problem named as*** x^{th} ***root problem. A new variant of Digital Signature Algorithm is also proposed based on*** x^{th} ***root problem.***

Keywords- Digital Signature; Factorization; RSA; x^{th} ***root; Cryptanalysis***

I. INTRODUCTION

A digital signature algorithm is a mathematical scheme which provides authenticity of a digital message and assure the recipient that the message was created by an authorized A digital signature algorithm is a mathematical scheme which provides authenticity of a sender and was not modified in transit. RSA Digital Signature Algorithm (RSADSA)[6] proposed by Rivest, Shamir and Adleman, is a popular and well known digital signature algorithm. The RSA signature algorithm is unforgeable in the random oracle model assuming the integer factorization problem is unsolvable. This scheme is based to the RSA public key cryptosystem [6]. The Rabin Digital Signature Algorithm [5] is also an asymmetric cryptographic technique, whose security, like that of RSA, is related to the difficulty of prime factorization. The ElGamal Digital Signature Algorithm [2] is based on difficulty of solving Discrete Logarithm problem. Hence there are many Digital Signature Algorithms that are based on one of the computationally infeasible problems. In this paper, a new variant of digital signature algorithm is proposed which is based on difficulties of solving x^{th} root problem in modular arithmetic. Rest of the paper is organized as follows. Section 2 describes brief overview of e^{th} root and x^{th} root problem. Section 3 contains the proposed algorithm. Finally, in Section 5, paper is concluded.

II. THE E^{TH} ROOT AND X^{TH} ROOT PROBLEM

RSA cryptosystem is based on computing e^{th} roots over the integer modulo with composite integer.

Definition 1. Let G be a Group and $e < |G|$ be an integer. An element $b \in G$ is called an e^{th} root of an element $a \in G$ if we have

$$b^e = a.$$

If $\gcd(e, |G|) = 1$ holds, then an e^{th} root always exists and is unique. Computing e^{th} roots is feasible if $|G|$ is known: it can be solved by computing e^{-1} in $Z^*_{|G|}$ and thus obtaining $b = a^{e^{-1}}$.

Definition 2. (The e^{th} root problem (ERP)): Given a group G of unknown order, a positive integer $e < |G|$ and an element $a \in G$, and an element $b \in |G|$ such that $b^e = a$.

If $G = Z_n^*$, with n being the product of two primes p and q, and the condition that $b \in G$ is replaced by $b \in Z_n$, we get the RSA problem. In this case the order of the group can be found by factoring n. Thus, if FACTORING is easy then so is the ERP.

Definition 3. Let n be the product of two primes p and q. Then an element $a \in Z_n$ is a quadratic residue modulo n

(or a square) if there exists an integer
w such that

$$w^2 \equiv a \bmod n.$$

If there exist no such $w \in Z_n$, a is called a quadratic non-residue. The set of all quadratic residues and the set of all non-residues in Z_n^* are denoted by QR_n and QNR_n, respectively.

If the factorization of the modulus is known, then there exists an e cient algo-rithm to decide whether an element is in QR_n. For a modulus with unknown factorization this is believed to be as hard as factoring, but this has not been proved.

Theorem 1. The x^{th} root Theorem: If $n = p \times q$, where p and q are large primes then, the equation $b^x \times c = 1 \bmod n$ has g number of solutions where $g = \gcd(x, \phi(n)), c \in Z_n$ and x is an odd number in Z_n, for $b \in Z_n$

Proof. Let $n = p \times q$, where p and q are prime numbers and $\phi(n)$ is the Euler's totient function. Consider the equation –

$$b^x = 1 \div c \bmod n. \qquad (1)$$

Where b and $x \in Z_n$. Now from Diophantine equation for x and $\phi(n); \exists u$ and v such that $xu - \phi(n)v = f$, where $f \in Z_n$. Now let $g = \gcd(x, \phi(n))$ then the computation $b \equiv (1 \div c)^u \pmod{n}$ gives the required value of b, since

$$b = (1 \div c)^u \bmod n$$

$$= (1 \div c)^{1+v\phi(n)/x} \bmod n$$

$$= (1 \div c)^{1 \div x} \bmod n.$$

Now if $g \mid f$ (g divides f) then it is easy to calculate b. Therefore the equation (1) is solvable for b if and only if $g \mid f$ i.e. if g - f then (1) has no solution.

Now if $g = 1$ then $g \mid f, \forall f$ therefore equation (1) is solvable for any f. But if $g > 1$, from equation (1),

$$b^x = 1 \div c \bmod n.$$

By Chinese Remainder Theorem, solving equation (2) is equivalent to solving the following two congruences.

$$b^x = 1 \div c \bmod p \qquad (3)$$

$$b^x = 1 \div c \bmod q \qquad (4)$$

Let $y = \left(x \div y^{-1}\right) \bmod (p-1)$ then $b^g = b^{(xy)} = (1 \div c)^y \pmod{p}$ (from equation (3)).
Let $w = (1 \div c)^y \pmod{p}$ then $b^g = w \bmod p$:
Thus problem of finding b, now reduces to the g^{th} root of $w \pmod{p}$. Now
Euler's theorem suggests that the equation $w^{\frac{(p-1)(q-1)}{g}} = 1 \bmod p$ has g number of solutions.

Corollary 1. If g > 2 is a large odd number then solving b^x c = 1 mod n, where n = p q, b; c 2 Z_n, x is odd number in Z_n, is a hard number theoretic problem, for b 2 Z_n.

Proof. From x^{th} root theorem, it is clear that $b^x c = 1 \bmod n$ has g number of solutions, where g = gcd(x; (n)). Now if g is even number, then using the Tonelli-Shanks algorithm[7] one can nd g^{th} root by successive square roots (for details refer x^{th} root theorem). But if g is an odd number then, from best of author's knowledge there is no general algorithm to nd g^{th} root except exhaus-tive search. Therefore if g is large odd then nding value of b for the equation $b^x c = 1 \bmod n$ becomes a hard number theoretic problem.

This hard number theoretic problem may be applied in the area of cryptography. This paper presents a Digital Signature Algorithm based on x^{th} root problem.

III. THE PROPOSED SIGNATURE ALGORITHM

This section proposes a new variant of digital signature algorithm based on difficulty of solving x^{th} root problem. The algorithm is as follows:

A. Key Generation

Followings are the key generation steps:

- Choose two large prime numbers p and q and calculate $n = p \times q.$
- Calculate $\phi(n) = (p-1) \times (q-1)$
- Choose large random numbers b; r 2 Z_n.
- Choose an odd large number x 2 Z_n such that x 2= $Z_{(n)}$ and g=gcd(x; (n)) is modestly large odd number (at least larger than 2).
- Calculate c such that

$$b^x \times c = 1 \bmod n$$

- Calculate y such that

$$y = r^x \bmod n$$

- Public key is (x; c; y) and private key is (b; r).

B. Signature Generation

- Calculate signature $s = r \times b^{H(m)}$ mod n, Here H (:) is a one way hash function. s is the signature of message m.
- Sender sends (s; m) to receiver.

C. Signature Verification

Calculates H(m) using the received message m at receiver's end. **{** if

$$s^x \times c^{H(m)} \equiv y(\bmod n)$$

then the signature is valid else reject the signature.

D. Proof of Correctness

This section contains correctness proof for signature veri cation process of the proposed digital signature algorithm..

$$\begin{aligned} \text{L.H.S.} &= \left(s^x \times c^{H(m)} \bmod n\right), \\ &= \left(r \times b^{H(m)}\right)^x \times c^{H(m)} \bmod n, \\ &= \left(r^x \times \left(b^x \times c\right)^{H(m)}\right) \bmod n, \end{aligned}$$

$= y$

= R.H.S.

IV. ANALYSIS OF THE PROPOSED ALGORITHM

A. Security Analysis

There are some possible areas where an adversary (Adv) can try to attack on this newly developed signature algorithm. Following are the possible attacks (not exhaustive):

- **Key-Only Attack:** Adv wishes to obtain private key (b; r) using all informa-tion that is available from the system. For this, Adv needs to solve the x^{th} root problem. For nding b, Adv has to solve b = c $^{1=x}$ mod n and as described in x^{th} root theorem, for large gcd(x; (n)), it is a hard number theoretic problem. Further for nding r, Adv has to solve r = y $^{1=x}$ mod n and it is also a x^{th} root problem.

- **Chosen-Message Attack:** In this attack, Adv requires a sign on some messages of his choice by the authorized signatory. With the help of chosen-messages and corresponding signatures, Adv generates another message and can forge sender's signature on it. The RSADSA algorithm is forgeable for this attack. For attack on RSADSA, suppose, Adv asks signer to sign two legitimate messages m_1 and m_2 for him. Let us assume s_1 and s_2 are signatures of m_1 and m_2 respectively. Adv later creates a new message $m = m_1 m_2$ with signature $s = s_1 s_2$. Adv can then claim that signer has signed m. The chosen-message attack for the proposed algorithm is a matter of further research as there is no obvious method which shows that the proposed algorithm is vulnerable to this attack.

- **Blinding:** In this attack, in case of RSADSA suppose Adv wants sender's sig-nature on his message m. For this Adv try the following: he picks a random r 2 Z_n and calculates $m^0 = r^e$ m mod n: He then asks sender to sign the message m^0. Sender may provide his signature s^0 on the message m^0. But we know that $s^0 = (m^0)^d$ mod n. Adv now computes $s = s^0$=r mod n and obtains sender's signature s on the original m. This technique, called blinding, enables Adv to obtain a valid signature on a message of his choice by asking Sender to sign a random blinded message. Sender has no information as to what message he is actually signing. So, RSA is vulnerable to this attack. Again an intensive research is required to check whether the proposed algorithm is vulnerable to Blinding or not. Currently, best of authors e orts it seems not vulnerable for Blinding.

B. Performance Analysis

Using the criterion presented in [1], the complexity of each method is estimated as a function of number of bit operations required. The basic exponential operation here is a^bmod n and time complexity of this operation is O(logb M(n)), where M(n) is the complexity of multiplying two n bit integers. In the proposed algorithm signature generation requires one modular exponentiation and signature verification requires two modular exponentiation, which leads to the complexity of the algorithm to be $O(\log^3 n)$ for signature generation and 2 $O(\log^3 n)$ for signature verification as here b = O(n) and time complexity of multiplying two n bit integers is $O(\log^2 n)$. If the complexity of proposed DSA is compared with other DSA algorithms of same category (i.e. DSA algorithms that are based on multiple hard problems) then we see that the Dimitrios Poulakis signature algo-rithm [4] requires 6 modular exponentiation in signature generation and 2 mod-ular exponentiation in signature veri cation. Ismail E. S signature algorithm [3] requires 5 modular exponentiation in signature generation and 5 modular expo-nentiation in signature veri cation. Shimin Wei signature algorithm [8] requires 5 modular exponentiation in signature generation and 8 modular exponentia-tion in signature veri cation. So it is clear that the complexity of the proposed algorithm is competitive equivalent to most of the digital signature algorithms.

V. CONCLUSION

In this paper, a new hard number theoretic problem called x^{th} root problem" is introduced. The application of the proposed problem is shown for designing a new Digital Signature Algorithm. The performance of the proposed algorithm is found to be competitive to the most of the digital signature algorithms which are based on multiple hard problems.

REFERENCES

1. D. Boneh and H. Shacham. Fast variants of RSA. CryptoBytes (RSA Laboratories), 5:1{9, 2002.

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. ES Ismail, NMF Tahat, and RR Ahmad. A New Digital Signature Scheme Based on Factoring and Discrete Logarithms. Journal of Mathematics and Statistics, 4(4):222{225, 2008.

4. D. Poulakis. A variant of Digital Signature Algorithm. Designs, Codes and Cryp-tography, 51(1):99{104, 2009.

5. M.O. Rabin. Digitalized signatures and public-key functions as intractable as fac-torization. 1979.

6. R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120{126, 1978.

7. D. Shanks. Five number-theoretic algorithms. Congressus Numerantium, 7:51{69, 1973.

8. S. Wei. A New Digital Signature Scheme Based on Factoring and Discrete Loga-rithms. Progress on Cryptography, pages 107{111, 2004.

