A Universally Verifiable Blind Signcryption Scheme with Message Recovery

Manikant Prasad

Computer Science and Engineering

NIT Rourkela

A thesis submitted for the degree of

B.Tech

2015 March

Abstract

The proposed scheme provides universal verifiability to the blind-signcryptext with full message recovery at the end of the intended receiver. Based on a three entity model, the signcryptor, the requester and the receiver, the scheme also provides traceability and non-repudiation along with unforgeability of the parameters.

Dedicated to my Parents and whole family..

Acknowledgements

I take the opportunity to express my reverence to my supervisor Prof. Dr. Sujata Mohanty for her guidance, inspiration and innovative technical discussions all during the course of this work. I find words inadequate to thank her for enabling me to complete this work in spite of all obstacles.

I am also thankful to all faculty members and research students and Undergraduates of CSE Department, NIT Rourkela. Special thanks go to Prof. Dr. S. K. Jena and Prof. Dr. Ratnakar Dash for their constructive criticism and motivation during the course of my work.

Contents

List of Figures

LIST OF FIGURES

1

Introduction

1.1 Blind Signature

The concept of Blind Signature (1) came earlier in 1982 to allow a user to produce signature over a message, without revealing the original message to the Signer and maintain the anonymity of participants. Since then it has found a popular use in secure payment systems etc.

1.1.1 Introduction to Blind Signature

Blind Signature is a digital signature scheme where the original message is disguised from the signer of the message as the signing entity is different from the author of the message and the un-blinded message can be universally verified by anyone like any other digital signature schemes.

It is required generally in the three-party model where the signer is a third party and the message being exchanged between the author of the message and the intended receiver is sensitive enough to be hidden from the third party or if the sender and receiver choose not to trust the third party completely with the original content of the message being exchanged.

1.1.2 How it works

1. Suppose Alice wants Bob to sign a message m, but does not want Bob to know the contents of the message.

2. Alice "blinds" the message m, with some random number b (the blinding factor). This results in **blind(m,b)**.

3. Bob signs this message, resulting in **sign(blind(m,b),d)**, where d is Bob's private key.

4. Alice then unblinds the message using b, resulting in **unblind(sign(blind(m,b),d),b)**.

5. The functions are designed so that this reduces to **sign(m,d)**, i.e. Bob's signature on m.

1.1.3 Blind RSA Signature

One of the simplest blind signature schemes is based on RSA signing. Here are the steps:

1. Assume e is the public RSA exponent, d is the secret RSA exponent and N is the RSA modulus.

2. Select random value r, such that r is relatively prime to N (i.e. $gcd(r, N) = 1$).

3. r is raised to the public exponent e modulo N.

4. $r^e mod N$ is used as a blinding factor.

5. Because r is a random value, $r^e mod N$ is random too.

6. The author of the message computes the product of the message and blinding factor, i.e.

$$m' \equiv mr^e \pmod{N}$$

7. And sends the resulting value m' to the signing authority. Because r is a random value and the mapping $r \mapsto r^e \bmod N$ is a permutation it follows that $r^e \bmod N$ is random too. This implies that m' does not leak any information about m. The signing authority then calculates the blinded signature s' as:

$$s' \equiv (m')^d \pmod{N}$$

8. s' is sent back to the author of the message, who can then remove the blinding factor to reveal s, the valid RSA signature of m:

$$s \equiv s' \cdot r^{-1} \pmod{N}$$

9. This works because RSA keys satisfy the equation $r^{ed} \equiv r \pmod{N}$ and thus

$$s \equiv s' \cdot r^{-1} \equiv (m')^d r^{-1} \equiv m^d r^{ed} r^{-1} \equiv m^d r r^{-1} \equiv m^d \pmod{N},$$

hence s is indeed the signature of m.

1.1.4 Uses of Blind Signature

Blind signature schemes see a great deal of use in applications where sender privacy is important. This includes various "digital cash" schemes and voting protocols.

For example, the integrity of some electronic voting system may require that each ballot be certified by an election authority before it can be accepted for counting; this allows the authority to check the credentials of the voter to ensure that they are allowed to vote, and that they are not submitting more than one ballot. Simultaneously, it is important that this authority does not learn the voter's selections. An unlinkable blind signature provides this guarantee, as the authority will not see the contents of any ballot it signs, and will be unable to link the blinded ballots it signs back to the un-blinded ballots it receives for counting.

1.2 Signcryption

Signcryption (2) was introduced in 1997 to bring down the cost of encryption and signature of the message by implementing both in one single step unlike encrypt-then-sign and sign-then-encrypt schemes where it took two separate steps for the same.

1.2.1 Introduction to Signcryption

To avoid forgery and ensure the confidentiality of a letter, it is a common practice for the originator of the letter to sign it and then seal the signed letter in an envelope. The same two-step approach can be adapted to the digital world where the originator

of a digital message can ensure the unforgeability and confidentiality of the message by signing the message using a digital signature algorithm followed by encrypting the digitally signed message using a public key encryption algorithm.

Cryptographic operations for signature and encryption are relatively expensive as they typically involve computations on astronomically large numbers and generate additional communication overhead. With the "digital signature followed by public key encryption" method described above, the computational and communication overhead for achieving unforgeability and confidentiality is the sum of the overhead for digital signature and that for public key encryption.

Signcryption is a public key cryptographic method that achieves unforgeability and confidentiality simultaneously with significantly smaller overhead than that required by "digital signature followed by public key encryption". It does this by signing and encrypting a message in a single step, fulfilling a cryptographer's dream to " kill two birds with one stone ".

1.3 Related Work

There has been some work done on Blind Signcryption in recent past (3, 4) including Blind identity based signcryption (5), but nothing considerable has been achieved yet in the topic and is expected to grow with number of literatures to come as large volume of work is being done for the same.
Apart from that the Blind signature schemes and Signcryption schemes are relatively long existing and huge amount of literature with a good success rate has been written and is available for reference. Combining the two provides greater security than any one used alone. The basic idea behind the concept is to bring down the computational cost as well. As, in blind signature, the actual transaction is of signature only, providing authenticity to the sender. Any encryption of the message, if required, is done separately. While in blind signcryption, a sender can also encrypt the message in the same step along with signature and thus providing higher security at relatively lower computational cost.

2

Objective of the project

2.1 Final aim

Here in this paper we have tried to work with the combination of above stated two strategies to implement a Blind Signcryption scheme which is universally verifiable. The proposed scheme consists of three participants, namely, Signcryptor, Requester and Verifier. This whole scheme is a five phase process- Key Generation, Blinding, Signcryption, Unblinding and Verification. We will discuss about the scheme in detail in further sections. Our ultimate goal is to design an effective scheme which can further be implemented in various applications. We will start with designing a blind signcryption scheme without bilinear pairing as most of the previous work that has been done is based on Identity based encryption and involves bilinear pairing and very less or relatively no amount of related work is done on DLP based scheme.

Once done with that we will move further to find applications whose computation cost can be improved by implementing our scheme and in process making desired changes in the scheme based on requirements of the application.

2. OBJECTIVE OF THE PROJECT

3

Proposed Scheme

3.1 Phases of proposed scheme

The proposed scheme goes through five phases overall during the whole communication. The Signcryptor performs Key Generation after which the requestor interacts with the Signcryptor to perform Blinding. Signcryptor then signcrypts the message and sends it to the requestor, who then un-blinds the message received and then authenticity of received message is verified. Let us now describe the whole scheme in detail at each phase.

3.1.1 Key Generation

Key generation is the first phase of the scheme, where the public/private key pair is generated by the Signcryptor. The operations of this phase are described as follows.

1. The Signcryptor chooses two large primes p_1 & p_2 in random and computes $n = p_1 p_2$. Then he computes $p = 2n + 1$, such that p is a prime.

2. The Signcryptor chooses g as a generator in ${Z_p}^*$ and H as a secure hash algorithm. E and D are publicly known encryption and decryption algorithm.

3. The Signcryptor chooses his private key-pair (x, r) in random and computes y and h as follows,

$$h = g^r mod p \tag{3.1}$$

$$y = g^x mod p \tag{3.2}$$

The public key of Signcryptor is y. Then the Signcryptor sends h to the requester in a secure way and publishes the systems public parameters $param = (g, p, H, y, E, D)$.

Meanwhile, every verifier/intended receiver is assumed to have private/public key pair as $< x_V, y_V >$ and the requester as $< x_R, y_R >$ where $y_i = g^{x_i} modp$.

3.1.2 Blinding

The requester chooses his secret parameters (α, β, ω) in random and computes,

$$K = g^{\omega x_R} modp \tag{3.3}$$

$$C = E_K(m) \tag{3.4}$$

$$r_1 = H(K) \tag{3.5}$$

$$\mu_1 = H(hg^{\alpha}y^{\beta}) modp \tag{3.6}$$

$$\mu_2 = (\mu_1 + \beta) modp \tag{3.7}$$

Then he sends the blinded message (μ_2, r_1, C) to the signcryptor.

3.1.3 Signcryption

After receiving (μ_2, r_1, C), the signcryptor produces the signcrypted text Z as follows and sends it to the requester.

$$Z = (r + \mu_2 x) modp \tag{3.8}$$

3.1.4 Unblinding

1. After receiving Z, the requester computes

$$Z' = (Z + \alpha + \omega x_R) modp \tag{3.9}$$

2. Requester uses the public key of intended verifier/receiver to compute

$$C' = (C + {y_V}^{\omega x_R}) modp \tag{3.10}$$

3. The blind signcrypted text on message m is

$$\sigma = (Z', \mu_1, K, r_1, C') \tag{3.11}$$

3.1.5 Verification and Message Recovery

1. When a receiver obtains σ, he verifies authenticity of the received message as per following two conditions,

$$r? = H(K) \tag{3.12}$$

$$H(g^{Z'}y^{-\mu_1}K^{-1})modp = \mu_1 modp \tag{3.13}$$

2. If the above two expressions are satisfied, the blind signcryption is assumed to be a valid one, otherwise it is rejected.

3. Then the receiver recovers message m as follows

$$m = D_K(C' - K^{x_V} modp) \tag{3.14}$$

3.2 Proof of Correctness

This sections gives a formal proof of the verification equation (3.13) and its correctness.

$$H(g^{Z'}y^{-\mu_1}K^{-1})modp$$

$$= H(g^{Z+\alpha+\omega x_R}y^{-\mu_1}K^{-1})modp$$

$$= H(g^{Z}g^{\alpha}g^{\omega x_R}y^{-\mu_1}K^{-1})modp$$

$$= H(g^{r+\mu_2 x}g^{\alpha}g^{\omega x_R}y^{-\mu_1}K^{-1})modp$$

$$= H(hg^{(\mu_1+\beta)x}g^{\alpha}g^{\omega x_R}y^{-\mu_1}K^{-1})modp$$

$$= H(hg^{\mu_1 x}g^{\beta x}g^{\alpha}Ky^{-\mu_1}K^{-1})modp$$

$$= H(hy^{\beta}g^{\alpha})modp$$

$$= \mu_1 modp$$

3.3 Security Analysis

Here in this section we are discussing about the security of the proposed scheme. Although many are similar to the security analysis of underlying schemes in their respective original papers on Signcryption and Blind Signature, we have pointed out some of the most important security aspects.

3.3.1 Blinding

In blind signcryption, perfect blinding is of most importance so that the Signer should not be able to see the message and anonymity of the participants is maintained. The Signcryptor receives blinded message (μ_2, r_1, C) and produces Signcrypted text over it and hence he has no access to the original message.

3.3.2 Confidentiality

Confidentiality is the sole interest of all the cryptology schemes and should be satisfied by all. This scheme satisfies confidentiality and is only compromised when the private key of the intended verifier/receiver x_V is no longer confidential, which is assumed to be impossible.

1. Required equation for message recovery is,

$$m = D_K(C' - K^{x_V} mod p)$$

2. Now,$K^{x_V} mod p = g^{\omega x_R x_V} mod p$, which cannot be solved even if K and y_V are known separately, as it is Decisional Diffie-Hellman Problem and no polynomial time probabilistic algorithm solves DDHP with non-negligible probability.

3.3.3 Unforgeability

We have three entities here who can try to forge the blind-signcryptext lets name them as S (Malicious Signcryptor), R (Malicious Requester) and V (Malicious Verifier). Let us verify them one by one.

1. If S tries to forge the blind-signcryptext:
   It can produce forged secret parameters $(\alpha', \beta', \omega')$ but to forge the key K it needs to know the secret key x_R of the requester, which is considered impossible and it cannot forge x_R as the public key y_R of the requester has been already made public and knowing the public key one cannot forge a corresponding private key of any user as it requires solving discrete logarithm which is hard.

2. If R tries to forge the blind-signcryptext:
   Any malicious requester can imitate as someone else to try to forge the required

blind-signcryptext by involving with Signcryptor as man-in-the-middle, but for that the adversary R will have to produce a private key corresponding to the public key of the genuine Requester, but it is considered as hard problem and thus, even the adversary R cannot forge parameters like K and others which involves the private key of the genuine Requester.

3. If V tries to forge the blind-signcryptext:
   An intended receiver can forge the blind-signcryptext as it has ways to regenerate all the parameters required and can forge a blind-signcryptext claiming as it being received from the Requester.

3.3.4 Universal Verifiability

The scheme is universal verifiable as all the parameters required for verification are sent to the receiver at once. Anyone having that blind-signcryptext can verify the authenticity of the message using equations (3.12) and (3.13).

3.3.5 Traceability and non-repudiation

A valid requester can always prove his authenticity by publishing his secret key x_R and secret parameter ω and thus using the formula,
$K = g^{\omega x_R} mod p$, If the parameters satisfy this equation then the user is considered legal.

For more security analysis, original (6, 7) schemes can be referred.

4

Implementation results

The proposed scheme was implemented in Java using security package and BigInteger class. The snapshots of implementation results are provided in this chapter.

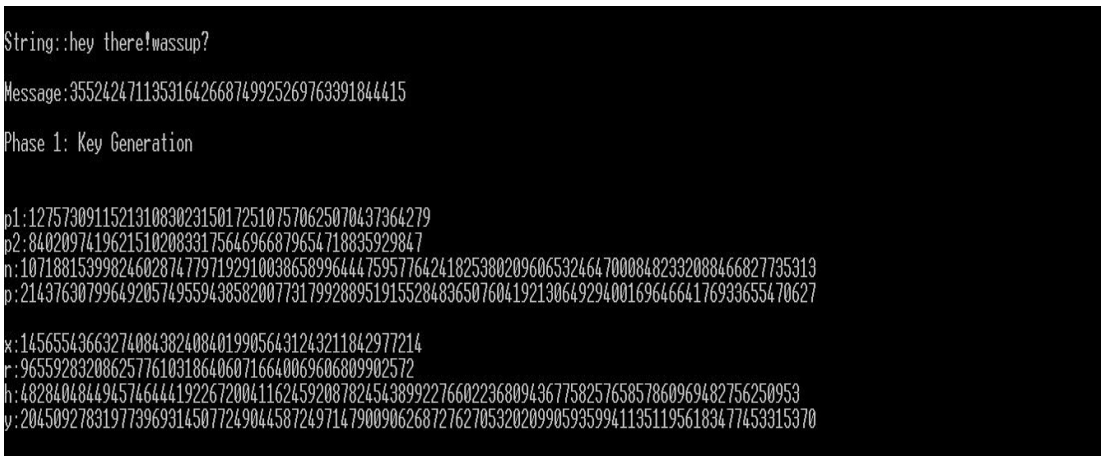

Figure 4.1: Phase 1: Key Generation - The figure shows various parameters generated in the first phase of the scheme

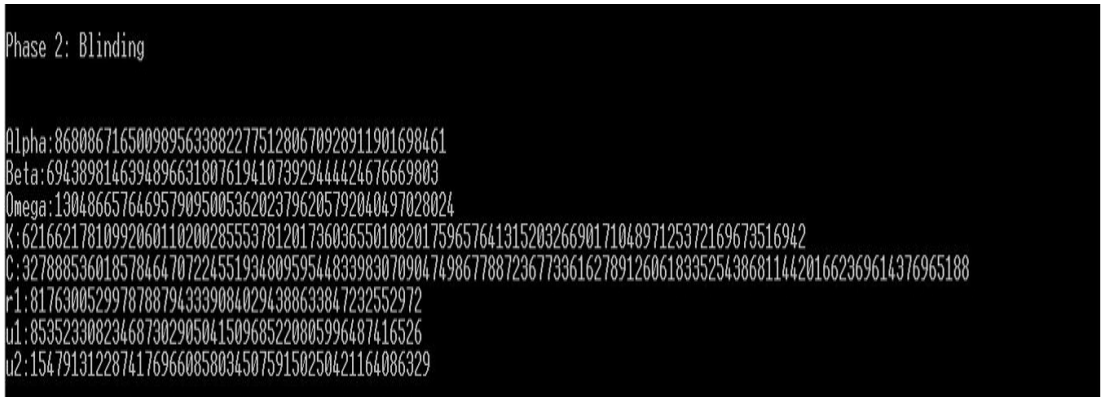

Figure 4.2: Phase 2: Blinding - The figure shows various parameters generated in the blinding phase of the scheme

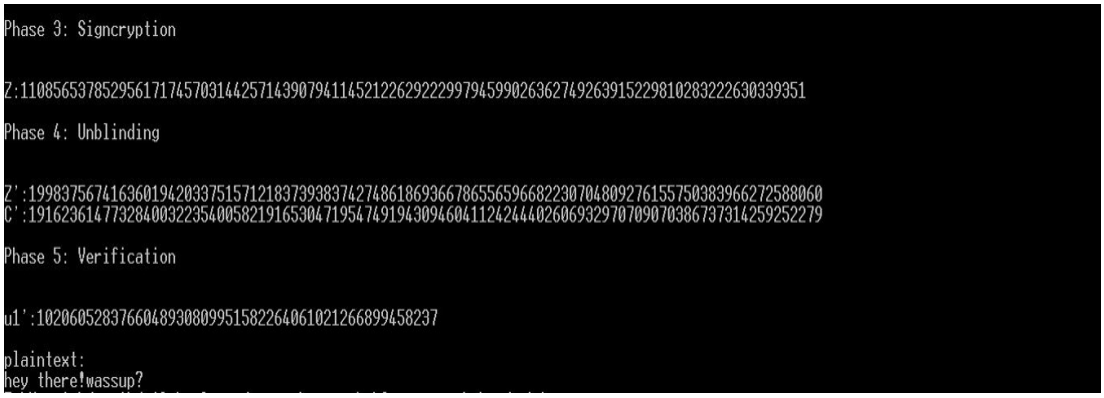

Figure 4.3: Phase 3, 4 & 5: Signcryption, Unblinding and Verification & Message recovery - The figure shows various parameters generated in the last three phases of the scheme

5

E-voting

Election is a fundamental instrument of democracy that provides an official mechanism for people to express their views to the government. Traditionally, the process of voting is quite cumbersome because voter must come in person to vote. This problem results in the low participation rate of voting. Vote-by-mail cancater for certain voters such as those who live in sparsely populated areas and who work far away fromthe voting centers. However, this method is time-consuming and cumbersome for the authority to manage since it requires extra work to send, collect and count the ballots manually. Electronic voting system or EVS can overcome those problems. EVS is expected to make our modern social life more convenient, efficient and inexpensive. By using EVS in national election, a voter can vote from his home or office.

EVS must meet security requirments such as confidentiality, integrity, authentication, and verifiability. This is becauseEVS is more vulnerable than traditional voting due to the nature of digital processing of election data which can be easily manipulated, hence may result in widespread fraud and corruption.

As an instance, we will be applying our scheme to e-voting. A simple e-voting entity model has been described below in the Fig. 5.1 and it also describes how the entities in the proposed scheme can act in the e-voting environment.

Basically there are three entities involved. Their definition and roles have been described below:

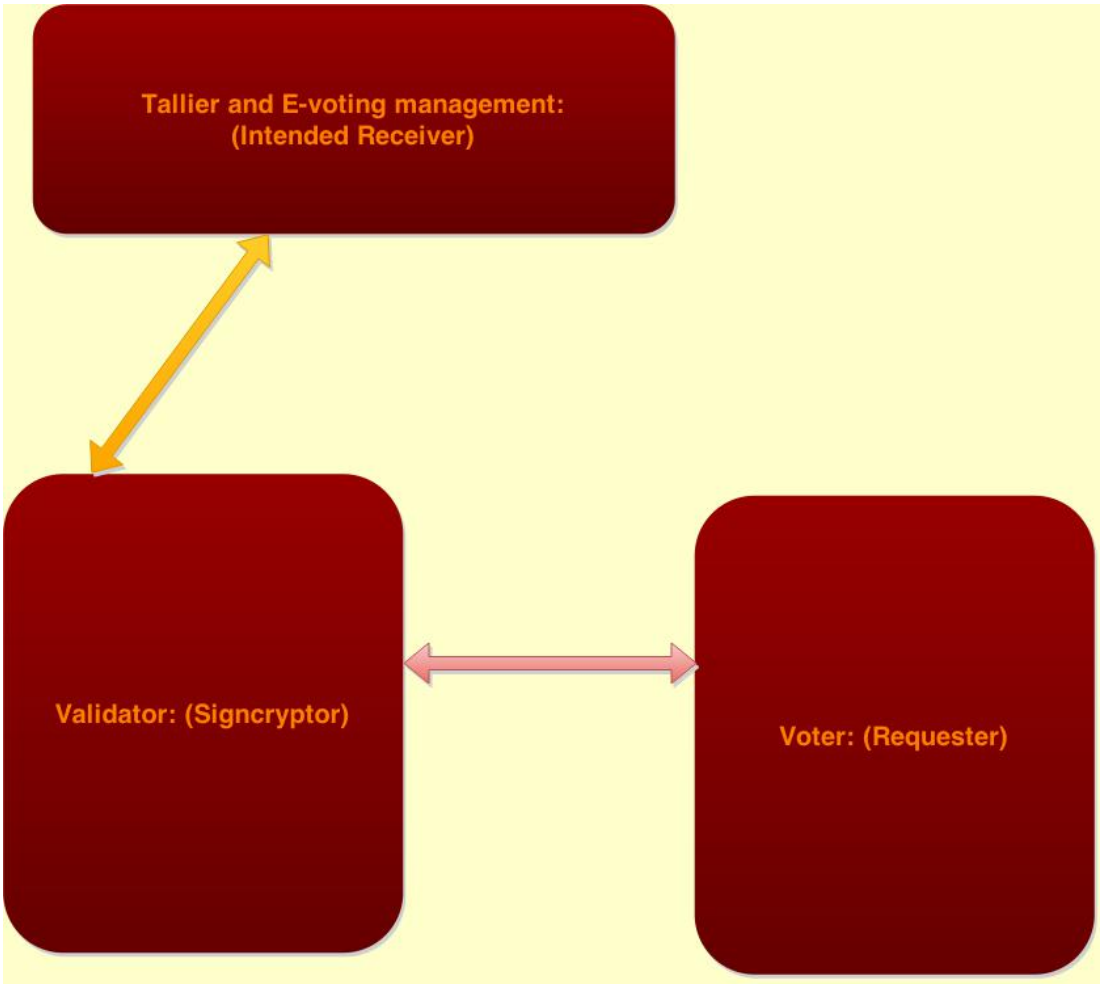

Figure 5.1: E-voting: A simple entity model - The figure shows entities in an e-voting protocol and how entities in the proposed scheme can be used

1. Voter: A voter is a national resident who is eligible to vote according to National Registration Database. A requester in the proposed scheme is equivalent to a voter in the e-voting environment.

2. Candidate: A candidate is the one who is contesting the election. A candidate has to be a legal voter in order to contest the election.

3. Validator: A validator can be a person who manages a particular e-booth and the local voters falling under his group of voters will request for signature from him and in return he will be providing validated ballets to the voters. A signcryptor in the proposed scheme is equivalent to a validator in the e-voting environment.

4. Tallier and E-voting Management: E-voting management can consist of A Tallier, Administrator and Registrar. Administrator of E-Voting is responsible in setting the dates of registration and voting.Besides that, administrator registers voter to become a candidate. Registrar server checks the users particulars with the national registration database to determine the eligibility of a voter and his precinct. A Tallier is the one responsible to validate the incoming ballets and store it in the voting database.

Figure 5.2 provides a sample E-voting System Architecture (8) consisting of described entities and the process of interaction among them.

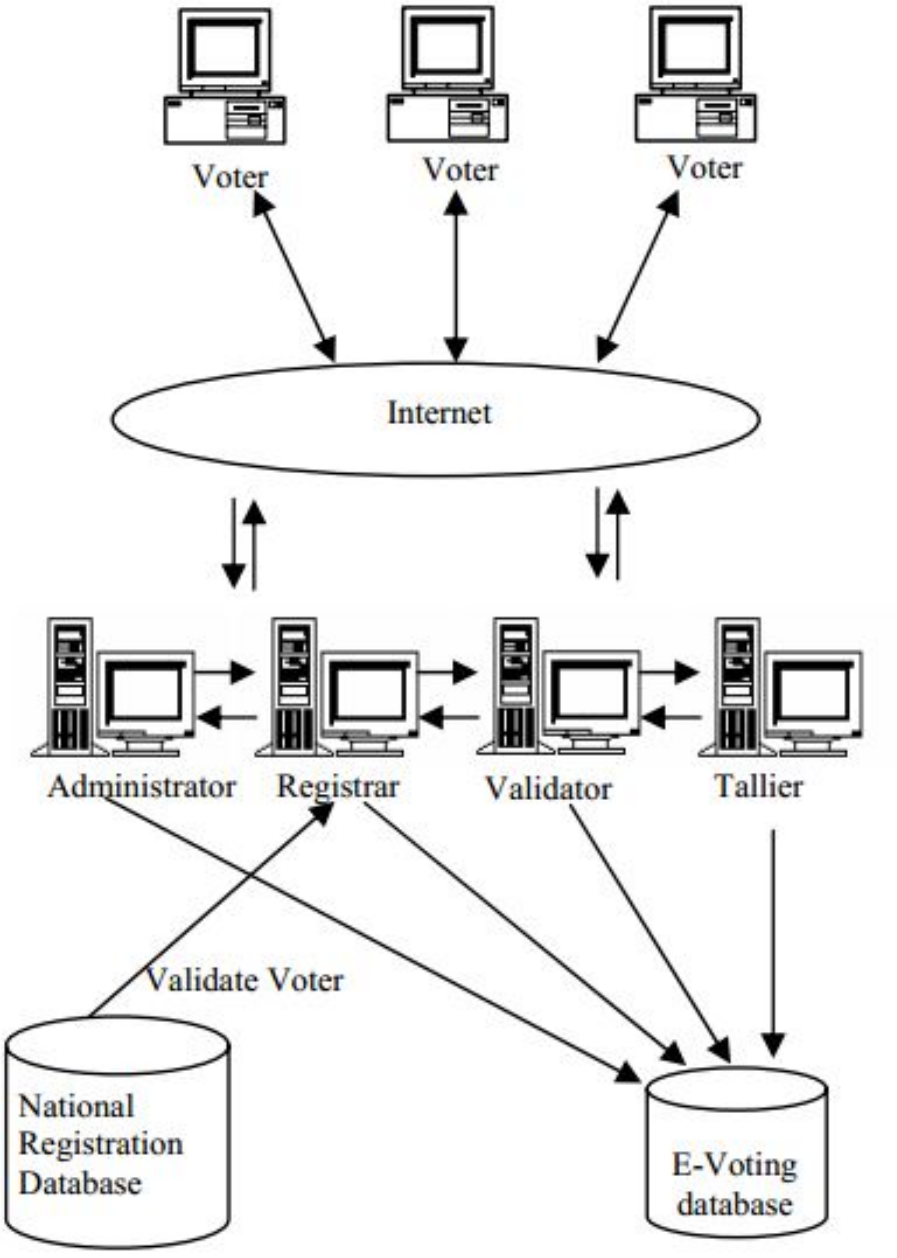

Figure 5.2: E-voting: Sample System Architecture -

6

Conclusion and Future Work

Signcryption and Blind Signatures are both strong individually, with Signcryption providing confidentiality and authentication efficiently and Blind Signatures providing efficient anonymity of participants. Both combined makes the scheme even stronger providing anonymity, authentication, confidentiality and unforgeability. The scheme proposed above is more secure than normal signcryption schemes when it comes to maintain the users anonymity during the transaction.

As the basic scheme now has been designed we will now try to modify it to increase efficiency and security. Also our future work includes designing a multi-user scheme without much increase in space or computation cost.

References

[1] Chaum D. **Blind signatures for untraceable payments**. *Advances in Cryptology Crypto82, LNCS*, 1982. 1

[2] Zheng Y. **Digital Signcryption or how to achieve cost (signature + encryption) ¡¡ cost(signature) +cost(encryption)**. *Advances in cryptology, Crypto97, LNCS, Vol. 1294*, 1997. 3

[3] STADLER A M CAMENISCH J L, PIVETIAU J M. **Blind Signatures based on the discrete logarithm problem**. *Advances in Cryptology-EUR0CRYPT92 Proceedings, Spring Verlag,428-432*, 1992. 4

[4] Sunder Lal Amit K Awasthi. **An Efficient Scheme for Sensitive Message Transmission using Blind Signcryption**. *arXiv:cs.CR/0504095 23 Apr 2005*, 2005. 4

[5] Victor K. Wei Tsz Hon Yuen. **Fast and Proven Secure Blind Identity-Based Signcryption from Pairings**. *Topics in Cryptology CT-RSA 2005, 305-322*, 2005. 4

[6] Zheng Y Baek J., Steinfeld R. **Formal proofs for the Security of Signcryption**. *6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003*, 2002. 11

[7] Stadler M. A. Camenisch J. L., Piveteau JM. **Blind signatures based on the discrete logarithm problem**. *LNCS 950, (1995) 428432*, 1995. 11

[8] Salleh M Ibrahim S, Kamat M. **Secure E-voting with blind signature**. *Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on*, 2003. 16

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other Indian or foreign examination board.

Manikant Prasad
ROURKELA,