Spam Control Mechanism using Identity based Message Admission

Mahesh P.
Canara Engineering College
Benjana Padavu, Bantwal

Basappa B. Kodada
Asst. Prof., Dept. of CSE
Canara Engineering College
Benjana Padavu, Bantwal

Shivakumar K. M
Head, Dept. of CSE
Canara Engineering College
Benjana Padavu, Bantwal

ABSTRACT
Spammers have a tremendous financial incentive to compromise user Email accounts. Many approaches to curb spam have been developed. In the traditional DKIM signature approach no Certificate Authority is used. Survey tells that most of the DNS are exploited to DNS cache poisoning attack. Further smaller keys are exposed to Wiener attack. Also DKIM does not verify the author and does not provide security after signature generation/verification. An attacker can be able to trick the recipient by masquerading as a legitimate sender and insert malicious information and send as spam Mails to other recipients. Therefore this paper addresses the issues by allowing sender to indicate that their emails are signed and encrypted using ID based mediated RSAA technique based on user identity. This system use Certificate Authority and a key Mediator in its architecture. During decryption the Key mediator does partial decryption and the recipient does full decryption of message. If original message is recovered and verified the sender message is accepted else rejected or blocked as spam message.

General Terms
Information security, Anti spam mechanism, Email security

Keywords
Spam control, DKIM, Wiener attack, DNS cache poisoning, and Identity based mediated-RSAA.

1. INTRODUCTION
Email is one of the most accepted ways of exchange of messages between two end users. Due to its worldwide accessibility, relatively fast message transfer, and low sending cost the extent of usage of Email system has grown exponentially. The loopholes in existing e-mail protocols and the increasing amount of e-business and financial transactions contribute to the increase in email based threats. Spam [1][2], phishing [3], and e-mail worms [4] are the forms of the e-mail-based threats nowadays. These reduce the basic quality and purpose of e-mail messages and proves threatful that leads to the financial loss to all entities in communication. Spam is nothing but unsolicited emails sent/ received in bulk [5].

In this paper architecture to control the spam messages by developing a signature based approach to stop messages from entering user inbox effectively is developed. The approach uses an identity based message admission. The signature scheme uses the key pair that is generated by the mediated-RSAA technique based on the UserID of a given user. If the decryption operation goes successful then the message is sent to user mail box else discarded. Hence the main focus is on the User ID needed to encrypt the outgoing mail. The remainder of the paper is arranged as follows. Section 2 provides the background about Spam research and various signature based approaches developed to curb spam and loopholes in traditional DKIM approach with are open to DNS cache poisoning attack and trivial attack on RSA key pair. The details of the proposed architecture of Email system based upon the Identity based mediated RSAA protocol along with the operational description has been discussed in section-3. Architecture of the email system corresponding to the solution in section 3 is designed in section 4 along with the algorithms and sequences of steps. The Detailed cryptanalysis of the proposed scheme with security analysis, time complexity analysis and probability of success has been discussed in section-5. Based upon the experimental results in section 6, conclusions have been drawn and some recommendations for future work have been proposed in section-7.

2. RELATED WORKS
Spam research has received an important attention from the research community in recent years. Anti-spam work groups have been formed, such as the Anti-Spam Research Group (ASRG) of the internet Research Task Force (IRTF) and Anti-Phishing Working Group (APWG) [6]. Variety of spam handling techniques is designed by various research communities. They can be categorized as filter-based approaches, signature-based approaches, protocol-level approaches, Identity based approaches and policy based approach and so on. Here each class of approaches has its own advantages. Using two or more approaches at the same time may work more effectively in better spam control.

SMTP servers do not provide any sort of facility to authenticate entities in communication of email system. Hence Signature-based approach can attempt to deal with the authentication problem by successfully authenticating the sender by the receiver. The basic science behind signature-based approaches is that for a message, the sender signs the message using a key and transmits it. At the other side, the recipients can effectively authenticate the message by signature verification.

There are various signature-based approaches like Domain Keys [7], Sender Policy Framework (SPF) [8] and Message Enhancements for Transmission Authorization (META) Signatures [9], and Certified Server Validation (CSV) [10]. Signature-based approaches in current E-mail system uses techniques that involve a sender to indicate that emails are protected by SPF and DKIM, and blocks a message as spam if neither of those authentication methods passes.

Domain Key Identified Mail specified in [RFC 4871] [11] is an email authentication system for segregating email spoofing. On the sender server side, the domain owner creates a pair of keys, one is a public key and another is a private key. The private key is stored in the email server and is used to

sign outgoing messages. The public key is published in the DNS server and is used by recipients to verify their incoming messages. When a sender tries to send an email message, the pre-stored private key will be used to generate a digital signature for that message. The digital signature will be attached to that message as a header entry. At the recipient server side, the mail server extracts the digital signature and the claimed "From:" Domain header from the received email and retrieves the public key from the DNS server for the Domain. The recipient-side email server uses the public key to verify the digital signature. If the signature is approved, the recipient-side email server will deliver the email to the recipient. Otherwise, the email will be flagged, blocked, or deleted.

Signers must sign message using rsa-sha256 algorithm. Signers may be actors in Email system like MUAs (Mail User Agents), MSAs (Mail Submission Agents) and MTAs (Mail Transfer Agents). The key issue is that a message must be signed before it leaves the administrative domain of the signer. Note that the algorithm used for creating the message digest is SHA-1 or SHA-256 and then the signing algorithm is RSA. The RSA-SHA1 [12] Signing Algorithm computes a message hash using SHA-1 as the hash-algorithm. That hash is then signed by the signer using the RSA algorithm defined in PKCS#1 version 1.5 [RFC3447] [13] as the crypto-algorithm and the signer's private key. DKIM signature scheme is traditionally different from the hierarchical public-key systems where no Certificate Authority infrastructure is required. The verifier requests the public key from a repository in the domain of the claimed signer directly rather than from a third party. The DNS is proposed as the initial mechanism for the public keys. Thus, DKIM currently depends on DNS administration and the security of the DNS system.

DKIM does not attempt to provide solutions to the world's problems with spam, phishing, viruses, worms, etc. DKIM provides one basic tool, in what needs to be a large arsenal, for improving basic trust in the Internet mail service. However, by itself, DKIM is not sufficient to that task.

By itself, a DKIM signature:

1. Does not authenticate or verify the contents of the message header or body, such as the "author from" field, beyond certifying data integrity between the time of signing and the time of verifying.
2. Does not offer any assertions about the behaviors of the signer.
3. Does not prescribe any specific actions for receivers to take upon successful signature verification.
4. Does not provide protection after signature verification.
5. Does not protect against re-sending (replay of) a message that already has a verified signature; therefore, a transit intermediary or a recipient can re-post the message that is, post it as a new message with the original signature remaining verifiable, even though the new recipient(s) might be different from those who were originally specified by the author.

Using DKIM when a message is sent, it that does not identify its author. Rather it permits authentication of the email system operator i.e. the DNS, rather than the content author. If it is possible to obtain effectively anonymous accounts at example.com, knowing that a message definitely came from example.com does threaten the anonymity of the user who authored it. Using DNS spoofing the whole security for the entire user community can be targeted and exploited.

Short RSA keys more easily accessible to off-line attacks. Signers must use RSA keys of at least 1024 bits keys and above. Verifiers must validate signatures with keys ranging between 512 bits to 2048 bits, and they may be able to validate signatures with larger keys. Verifier policies may use the length of the signing key as one metric for determining whether a signature is acceptable.

Also it is not safe to have the private key stored in the host machine's Hard Disk Drive in plaintext form. Also when '*e*' is small the RSA system always leaks half the most significant bits of'd' as given by D. Boneh et. al 1999[15]. Also a technique specified in [16], in which one can search for randomness in order to locate private keys in large volumes of data, such as the hard disk filing system, etc. therefore it should be clear how important is to manage the secrecy of the RSA private key in administrative domain of sender. Using of Hardware based tokens would be better to safeguard the private key. However the cost based issues come into picture in this case. A better solution to come up with the safe and secure use of private key is required.

DNS spoofing or DNS cache poisoning is the most prominent and dangerous attack on DNS server. DNS cache poisoning results in a DNS resolver storing or caching invalid or malicious mappings between symbolic Domain names and IP addresses for that corresponding DNS. Because the process of resolving a name depends on authoritative servers located elsewhere on the Internet, DNS protocol is intrinsically vulnerable to cache poisoning [18]. An attacker may poison the cache by compromising an authoritative DNS server or by forging a response to a recursive DNS query sent by a resolver to an authoritative server. The implementation of this attack is given in [17].

Wiener [19] proposed that information encoded in the public exponent '*e*' might help to factor '*n*'. He showed that for every public exponent $e \in Z^*_{\emptyset(n)}$ that corresponds to a secret exponent d with $d \leq (1/3)\, n^{(1/4)}$ yields the factorization of the modulus in time polynomial in $\log(n)$. Thus shorter RSA keys may lead to key exposure attack and the attacker can easily obtain both public and private exponents and use them to sign or encrypt outgoing mails or decrypt the incoming mails. Justification and algorithms on wiener attack on RSA are specified in [14].

TABLE 1. PAPERS JUSTIFYING THE ATTACK SCENARIOS

Sl. no.	Attack scenario	Research Paper title	Authors	Ref no.
1.	DNS Spoofing or DNS cache poisoning	The Hitchhiker's Guide to DNS Cache Poisoning	Sooel Son, Vitaly Shmatikov	[18]
2.	Wiener attack on RSA	A Generalized Wiener Attack on RSA	Johannes Blomer, Alexander May	[14]

Therefore in the proposed solution using the concept of mediated RSAA given by M. Kutylowski, P. et. al. 2011[20] is recommended. In this algorithm a simple and practical method of splitting RSA private keys between the user and a Key Mediator is used.

3. IDENTITY BASED MEDIATED RSA ADDITIVE (RSAA) ALGORITHM

The projected solution is designed on the basis of standard RSA algorithm which is defined in PKCS#1 version 1.5. Both encryption and decryption process are tangled in operation. The only difference is decryption involves 2 stages of

operation from the basic method. The proposed solution is derived from the simple Identity-Based Mediated RSA [21] where in a ID based authentication of email with unique ID of each user in the domain are used. ID-Based Mediated RSAA (additive) variant is a simple and practical method of splitting RSA private keys between the user and the Key Mediator. For both signature and message decryption operation the architecture involves both parties. ID-Based Mediated RSAA also allows faster revocation of users security privileges.

3.1 System setting

In the initialization phase, a trusted party (CA) sets up the RSA modulus for different users in the same domain. First, CA chooses, at random, two large primes p_0 and q_0 such that P = $2p_0$+1 and Q = $2q_0$+1 are also primes, and finally set n = (P*Q). It can be noted that, since n is a product of two strong primes, a randomly chosen odd number in Z_n has negligible probability of not being relatively prime to $\varnothing(n)$ Considering $\varnothing(n)$= $(2)^2$ p_0 q_0 with only three factors 2, P, Q, the probability of the output from Hash being co-prime to $\varnothing(n)$ is overwhelming on the condition that the output is an odd number, because finding an odder number not co-prime to $\varnothing(n)$ is equivalent to find p_0 or q_0 and consequently factoring n. Therefore $\varnothing(n)$ = LCM [(P-1), (Q-1)].

3.2 Key Generation

Key generation operation begins with creation of ID-mRSAA keys- a public key and pair of private keys.
In ID-based mediated RSAA the public key pair is represented as *(n, PU_{Total})*.

- *n*, the modulus
- PU_{Total}, the public exponent.

Here PU_{Total} is obtained as the output of KeyGen (UserID) represented as a binary string of the same length as the modulus, with the least significant bit set. This ensures that PU_{Total} is odd and with overwhelming probability, relatively prime to $\varnothing(n)$. The KeyGen function is typically a Hash function that is used to generate a hash value of given information i.e. UserID. The output of key generation will be such that

$$PU_{Total} = 0^S \,||\, \text{KeyGen (UserID)} \,||1 \quad (1)$$

Where s= k - |Bit-length used by KeyGen |-1.
(k- Size of modulus used).
However private exponent is split between two parties to satisfy the following relation:

$$PR_{Total} \equiv PR_{KM} + PR_{USR} \left(mod \varnothing(n)\right) \quad (2)$$

Where PR_{KM} = private component of Key Mediator, PR_{USR} = user's private exponent.
The Key mediator's private exponent PR_{KM} is derived from a master service key F_m (Random Integer) and a user identifier UserID. UserID is signed with key F_m, and this signature is a seed for pseudorandom bit generator. The resulting integer is PR_{KM}. Generation of the pseudorandom bit integer uses deterministic random bit generator based on Chinese remainder theorem given by Jean Claude Bajard et. Al 2009 [22]. Therefore PR_{KM} consists of pseudorandom bits of specified length i.e k. The resultant PR_{KM} will be equivalent to $1/(PU_{Total}) \bmod \varnothing(n)$
User's private exponent "PR_{USR}" is derived from Key Mediator's private exponent PR_{KM} and a base RSA key PR_{Total}. Assuming the PR_{Total} is a valid RSA private key corresponding to the public key *(n, PU_{Total})* for which PR_{KM} has been calculated PR_{USR} can be obtained as:

$$PR_{USR} \equiv (PR_{Total} - PR_{KM}) mod \varnothing(n)$$

Where $\varnothing(n) = lcm(p-1)*(q-1)$

$$PR_{Total} = (PU_{Total})^{-1} \bmod \varnothing(n). \quad (3)$$

Thus from equation 1, 2 and 3 the following key pairs are considered:
Public key pair: *(n, PU_{Total}).*
Key mediator Private Key pair: *(n, PR_{KM}).*
User private Key pair: *(n, PR_{USR}).*

3.3 Signature generation/verification

Signature Generation operation for the ID-mediated RSAA is identical to RSA based signature Generation which is defined in PKCS#1 version 1.5. The Signature generation operation MRSAA_SigGenOp computes the signature for a given message at sender side.

- MRSAA_SigGenOp *((n, PU_{Total}), H)*
  Where *(n, PU_{Total})* RSA public key, *H* is hash of a message, integer between 0 and n-1, Output will be signature representative *Sig*, an integer between 0 and n-1.

Signature Verification for the ID-mediated RSAA is also identical to standard RSA. MRSAA_User_SigVerOp operation verifies complete signature of a given message at recipient side.

- MRSAA_User_SigVerOp *((n, PR_{USR}), Sig)*;
  Where PR_{USR} user's private exponent to form the key pair *(n, PR_{USR}). Sig, a* signature representative. Output *H* representative of a complete message Hash an integer between 0 to n-1.

3.4 Encryption/ Decryption

Encryption operation for the ID-mediated RSAA is also identical to standard RSA encryption defined in PKCS#1 version 1.5. Here it uses the public key PU_{Total} generated during Key generation phase. It uses this public key pair *(n, PU_{Total})* to encrypt α where α= (Sig|| message) as input and the cipher text "CT" as output computed as given by

$$CT = \alpha^{PU_{Total}} mod\ n \quad (4)$$

This CT is then sent to the recipient for decryption. Decryption operation it is formulated in 2 stages. A partial decryption is done by Key Mediator and the full decryption on the recipient based on the exponents PR_{USR} and PR_{KM}. The 2 stage can be described below:When the CT arrives at the recipient MTA the control is transferred to the Key Mediator for Partial decryption followed by full decryption on recipient MTA. The operations are specified below.
For partial decryption:

- MRSAA_Key_Mediator_DOp *(K_{KM}, CT)*
  Where K_{KM} Key Mediator's private key in the form of a pair of *(n, PR_{KM})* and CT cipher text representative, an integer between 0 and n-1. Output $D_{Partial}$ will be partially decrypted representative of *α*, an integer between 0 and n-1.

For full decryption:

- MRSAA_User_DOp *(K_{USR}, $D_{Partial}$)*
  Where K_{USR} user's private key in the form of a pair of (*n*, PR_{USR}), "$Msg_{Partial}$" message partially decrypted by KM earlier, an integer between 0 and n-1. Output will be *α*, a full decrypted information required for user verification and data integrity.

Here the decryption operation is only successful only if the generated keys are used by the intended recipient. Else the message cannot be fully decrypted. Therefore only the

recipient who possess the PR_{user} key can only retrieve the "α" where the signature is verified and the sender is authenticated. Based on the above description of the operations involved in the ID based mediated RSAA technique an architecture of the entire Email system that can overcome the vulnerabilities of the existing system as described in the related work has been designed. The architecture diagram and the operational description are given in system design.

The algorithm for Sender, CA, Key Mediator, Recipient is given below

Algorithm ID-mRSAA KeyGeneration (executed by CA)

1. Start
2. Compute public exponent"PU_{Total}" as given by Eq. 3.1 on request by sender.
3. Compute two large primes p_0 and q_0 such that P = $2p_0$+1 and Q = $2q_0$+1 are also primes
4. Compute n=P*Q and Ø (n) = lcm [(P-1),(Q-1)]
5. Check for gcd(PU_{Total}, Ø(n)) = 1, if success go to 6 else go to 3
6. Derive PR_{KM} from a master service key Fm and User-ID of Recipient using CRT_PRBG.
7. Compute $PR_{Total} = (PU_{Total})^{-1} \bmod \varnothing(n)$
8. Compute $PR_{USR} \equiv (PR_{Total} - PR_{KM}) mod \varnothing(n)$
9. Securely send "PR_{USR}" and n to Recipient.
10. Send "n" and "PR_{Total}" to Sender on request.
11. Delete All key files created
12. Stop.

Fig 2 ID-mRSAA Key generation Algorithm (Executed by CA)

Algorithm ID-mRSAA encrypt

Initialize: Sender will be requesting for the modulus value *(n, PU_{Total})* from CA for a given UserID of recipient

1. Start
2. Sender generates the message to be sent as denoted by Msg.
3. Gets the *(n, PU_{Total})* pair for UserID of recipient sent by CA.
4. He then generates the cipher text of given message which is given by $CT = \alpha^{PU_{Total}} \ mod \ n$.
5. Sends the CT to the recipient.
6. Stop.

Fig 3 ID-mRSAA Encryption Algorithm (Executed by Sender)

Algorithm ID-mRSAA decrypt KM

Initialize: KM will be requesting for the *(n, PR_{KM})* pair from CA for a given UserID of recipient and will receive the CT from Recipient MTA

1. Start
2. KM does the partial decryption: $D_{Partial} = CT^{PR_{KM}} \bmod n$ and sends it to the recipient MTA
3. Stop.

Fig 4 ID-mRSAA Partial Decryption Algorithm (Executed by Key Mediators)

Algorithm ID-mRSAA decrypt Usr

Initialize: recipient will be requesting for the *(n, PR_{User})* pair from CA and will receive $D_{Partial}$ from KM to fully decrypt

1. Start
2. Obtains the partial decrypted message from KM in the form of $D_{Partial} = CT^{PR_{KM}} \bmod n$, it then fully decrypts the message as $\alpha = D_{Partial}^{PR_{User}} \bmod n$.
3. If the original α is recovered and process is SUCCESS after verification of signature then accept the message else reject from sending to mailbox.
4. Stop.

Fig 5 ID-mRSAA Full Decryption Algorithm (Executed by Key Mediators)

4. SYSTEM DESIGN

The following diagram gives the proposed architecture of the Email system to implement ID-mediated RSAA algorithm. The architecture consists of the MTA-A of sender and MTA-B of recipient. The Key mediator is the external server that plays the role of both CA (certificate authority) as well as the Partial decryptor.

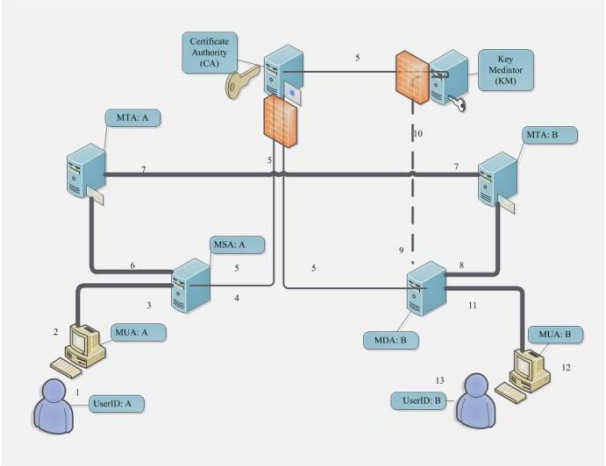

Fig 6: Architecture diagram of Email- system corresponding to ID-mRSAA architecture

The architecture for Email system corresponding to ID-mRSAA is shown in the Figure 6. The sequence of operations is given below:

1. User A composes message and adds the recipient user's UserID: B as Email address.
2. User enters UserID of recipient and uses the IDE plug-in.
3. MUA at A requests extracts UserID to be sent to CA for obtaining PU_{Total} from CA.
4. MSA: A sends the UserID of B to get the public key for Encryption at A.
5. CA sends the public key pair to the MSA A and corresponding private key pairs to Key Mediator (KM) and MDA: B for user B.
6. MSA at A then encrypts the message using this pubic key and sends to MTA-A
7. MTA-A transfers the Cipher Text to MTA-B.
8. MTA-B then receives Cipher Text and sends it to MDA-B for decryption process.
9. MDA-B sends the CT to Key Mediator (KM) for Partial decryption $D_{Partial}$ using private exponent PR_{KM}.
10. $D_{Partial}$ is obtained and sent back to MDA: B for Full decryption using user private exponent PR_{USR}.
11. MDA: B does the full decryption and sends to MUA: B if the operation goes successful.
12. MUA: B then does other validation process for filtering for spam.
13. User B then reads the message in his inbox.

5. CRYPTANALYSIS

In this section an analytical model for analysis of the security of the system for single as well as multi-user setting for the encryption and decryption operation has been formulated. Also the time complexity analysis of the system for a given polynomial time is obtained and the graph screening the variations for different values of parameters as formulated in the sub sections below.

5.1 Analytical model: Security analysis of ID-mRSAA

The security analysis is mainly derived from the results given in [23], where it was shown that a public-key encryption system in a multi-user setting is semantically secure against certain types of attacks if and only if the same system in a single-user setting is semantically secure against the same type of attack.

Analysis of the success of ID-mRSAA system is considered on the basis of analysis of mediated RSA formulated by Xuhua Ding et. Al. 2003 [24]. analyzing the security of ID-based Mediated RSAA is based on the same Lemma's given in [24] since the only difference is the way of key generation for the user. Therefore for the proposed solution $Success_1^{ID\text{-}mRSAA}(t; q_d)$ is used to denote the maximum advantage of all adversary or malicious algorithms in polynomial time t, attacking IB-mRSAA with one user, $Success_n^{ID\text{-}mRSAA}(t; q_d; q_e)$ for the setting with n users, and $Success^{RSA}(t; q_d)$ for RSA as given in [24]. In the above, $q_d(q_e)$ denote the maximum number of decryption /encryption queries allowed for each private/public key. Throughout the analysis, random oracle model [25] is considered for semantic model for the following lemmas:

Lemma 1: Identity Based-mediated RSAA/OAEP system for a single user setting is polynomially secure as standard RSA/OAEP encryption, i.e.

$$Success_1^{ID-mRSAA}(t;q_d) = Success^{RSA}(t';q_d) \quad (5)$$

Where $t' = t + c$,

c is constant value, and q_d is maximum number of decryption queries the adversary can ask.

Basically, if there exists an algorithm breaking the security of ID-mRSAA/OAEP in a single-user mode, attacker can build upon an algorithm breaking standard RSA/OAEP with the same success probability and constant extra overhead. Of course, it is easy to see that breaking RSA/OAEP implies breaking ID-mRSAA. Thus, it is claimed that they are equally secure.

For the multi-user setting, it cannot be claimed that ID-mRSAA with n users is semantically secure by directly applying the security reduction theorem. The reason is that proposed system is not a typical case referred in [26]. Sharing a common RSA modulus among many users results in their respective trapdoors not being independent; consequently, there could be attacks among the users. Furthermore, users in ID-mRSAA may have the incentive not only to attack other users, but also to attempt to break the underlying protocol so that they can bypass the mandatory security control of the KM. However, assuming for the moment, that all users are honest, the following lemma as given below:

Lemma 2: ID-mRSAA/OAEP system with n users is semantically secure if all n users are honest. i.e.

$$Success_n^{ID-mRSAA}(t_n;q_d;q_e) \le q_e n \; Success_1^{ID-mRSAA}(t';q_d)$$

Where $t' = t_n + O\left(log\left(q_e n\right)\right)$

q_e - The number of encryptions allowed to be performed by each user.

Unfortunately, in a real world application, all users cannot be assumed to be honest. For a given set of users in this system a term "system view" is used to refer to the distribution of inputs, outputs, current state information and modules of interactions with decryption oracles, encryption oracles, and the Key Mediator.

The system view for an outside attacker is denoted as:

$A1$::= Pr {N, $(e_0,\ldots,e_n)$, T_O, T_E, T_D, T_{KM} }.

While the system view for a set of users is:

$A2$ ::= Pr {N, $(e_0,\ldots,e_n)$, $d_{u(i)}$, T_O, T_E, T_D, T_{KM}, $T_{du,n}$ }

Where $d_{u(i)}$ is the set of user key-shares; T_O, T_E, T_D, are three scripts recording all queries/answers to the random oracle, encryption oracles and decryption oracles, respectively. T_{KM} is the script recording all requests/replies between all users and the KM, $T_{du,n}$ is the script recording all n users' computation on cipher texts with their own secret key share $d_{u(i)}$. It's been claimed that being an ID-mRSAA user does not afford one extra useful information $d_{u(i)}$ as compared to an outside adversary as claimed in Lemma 3.

Lemma 3: Under CCA-2, the system view of the outside adversary *(A1)* is polynomially indistinguishable from the combined system view *(A2)* of a set of malicious insiders, in the random oracle model.

So it can be said that insider adversaries do not gain advantages over outsiders in terms of obtaining extra information. In ID-mRSAA, each user is allowed to send legitimate decryption queries to its KM. However in this case, an inside adversary can manipulate a challenge cipher text (intended for decryption with d_i) into another cipher text that can be decrypted with its own key d_j and legally decrypt it with the aid of the KM.

Let (e_{a0} ..... e_{av}) be the set of public keys of v malicious users. $E_v = \prod_{a_i} e_{ai}$. They may attempt to use some function *f*, which takes a challenge CT = m^x mod n as input and outputs cipher text CT' = m^{E_v} mod n. following lemma was used to address the conditions for the existence of such f.

Lemma 4: Given two RSA exponents p, q and modulus n, then let *f* be a polynomial time complexity function such that

$$f\left(m^p\right) = m^q mod\ n$$

Such *f* exists if and only if *p/q*. Informally, a hash function H is Division intractable if it is impossible to find distinct $(p_1,\ldots,p_n, q)$ within the domain such that $H(Q) \mid \prod_i (H(P_i))$

Denoting P^{div} (H) as the probability that H fails to hold this property, following proposition regarding the security of ID-mRSAA in a multi-user setting was given as follows.

Proposition 1: ID-mRSAA encryption offers equivalent or more semantic security to RSA against adaptive chosen cipher text attacks in the random oracle model, if the key generation function is division intractable.

In summary, Lemma 3 and Lemma 4 remove the condition of Lemma 2 where all users are assumed to be honest, by requiring the key generation function to be division intractable. Thus, it can reduced that security of ID-mRSAA/OAEP in multiuser setting into single-user, which is as secure as standard RSA/OAEP according to previously defined Lemma 1.

More justification and proofs for security analysis of Identity-Based mediated RSAA are given in appendix in [24]. Based on those proofs it can be concluded that system is polynomially secure as standard RSA encryption.

5.2 Statistical Model: Time Complexity Analysis

The statistical model for the Time Complexity of ID-based mediated RSAA is formulated as given below. To begin with the analysis lemma 1 given in the previous section was consider that gives the equation for the polynomial time taken by the RSA algorithm equivalent to ID-mRSAA algorithm for single user setting according to the lemma.

Consider the various parameters required for Time Complexity analysis as follows. Let c= 520 ms which takes constant time for pre-computation of keys and initialization phase which include the encryption for a single user. Then decryption query q_d runs in discrete time t. According to the lemma the total time is given as $t' = t + c$, so consider the following data specified in the table below for which the t' value is obtained as follows

Table 2. Parameter and values specified for RSA single user setting analysis

q_d (query)	t	Value $t' = t+c$; (c = 520 ms)
1	210 ms	770 ms
2	396 ms	916 ms
3	589 ms	1.10 s
4	799 ms	1.3 s
5	1.15 s	1.67s
6	1.97 s	2.49 s

For the Time Complexity analysis of ID-mRSAA in multi user setting consider the lemma 2 in order to obtain the equation for the total polynomial time taken by the system. According to the lemma in the multi user setting for n user, if t_n is the time utilized for different q_e for a specified n to decrypt, and the pre-initialization stage takes time in the Order of $(log(q_e\ n))$, then total time would be $t' = t_n + c'$; Where c' will be time taken by pre-initialization stage in the Order of $(log(q_e\ n))$.Based on the above equation use it with the parameter values specified in the table below for which t' value are obtained as shown.

Table 3. Table 5.2 Parameter and values specified for ID-mRSAA multi user setting

q_e & n	t_n	$c' = O(log(q_e n))$	Value $t' = t_n + c'$
1	44ms	270ms	505ms
2	90 ms	510ms	600ms
3	135ms	810ms	945
4	180ms	1.08 s	1.26s
5	230ms	1.3 s	1.5s
6	270ms	1.62 s	1.89s

From fig 6 it can be entailed that the total time t' for RSA is smaller compared to that of n-user setting using ID-mRSAA for smaller n. For some n it will yield equal or similar values. As n value increases t' will be a bit low for ID-mRSAA. Consequently the attacker need an algorithm to implement CCA-2 attack on ID-mRSAA such that it is able to obtain CT with similar probability of success and polynomial time t' compared to that of RSA. According to the theory of CCA-2 attack on RSA it is very challenging to realize a similar CT for a given message equivalent to original CT within a given time $t'_{ID\text{-}mRSAA} < t'_{RSA}$ for an attacker if the public exponent become division untraceable for given set of p, q in the system. Thus the larger bit size of public exponent means stronger the CT and more time required to compute CT by the adversary for n user setting. Therefore security of the system with ID-mRSAA under certain attacks will be similar to that of RSA provided public exponent is division untraceable.

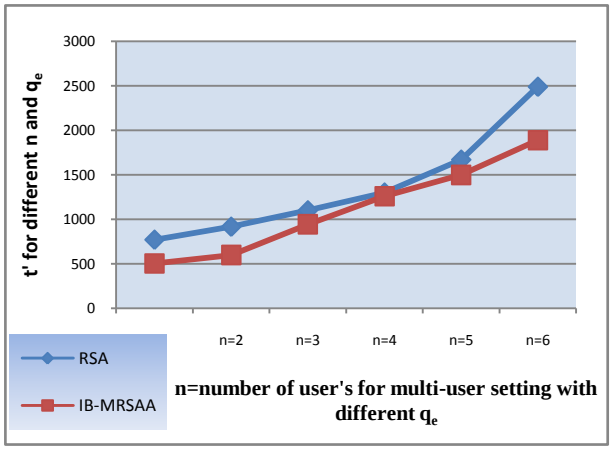

Fig 6. Comparison graph for multi-user setting in ID-mRSAA and RSA setting

5.3 Estimation of Success Rate for an attacker

To derive the equation for success rate for the ID-mRSAA solution consider the type of distribution needed to be considered for the system. In a binomial experiment there are two mutually exclusive outcomes, often referred to as "success" and "failure". If the probability of success is p, the probability of failure is 1 - p. Such an experiment whose outcome is random and can be either of two possibilities, "success" or "failure" is called a Bernoulli trial.The random variable r that counts the number of successes in the n trials, then it is said to have a binomial distribution as: binomial (n, r, p).

When computing a binomial probability, it is necessary to calculate and multiply three separate factors:

1. The number of ways to select exactly r successes
2. The probability of success p raised to the r power,
3. The probability of failure q rose to the (n - r) power.

Thus the formulae are as follows:

The probability of an event, p, occurring exactly r times: $^{n}C_{r} * p^{r} * q^{(n-r)}$ (6)

n= number of trials considered.

r = number of specific events we wish to obtain as success

p = probability that the event will occur as success

q = probability that the event will not occur as success.

Thus for the given solution let use consider assumption that for n trials the attacker in the system tries to obtain the compromised CT' for a given set of parameters as given below. Here the system uses different values of modulus for encryption and decryption. For given set of trials if the probability of success of the system for given number of success events as shown in the table (p,q) then the rate of success for an attacker to compromise the system as follows:

Table 4. Parameter and values specified for ID-mRSAA success rate evaluation

Mod size used (k)	1024	2048
No. of Trials (n)	6	6
No. Success events (r)	2	2
Probability of success (p)	0.33	0.5
Probability of Failure (q=1-p)	0.67	0.5

Rate of Success	0.3291	0.2343
Rate of Failure	0.6709	0.7657

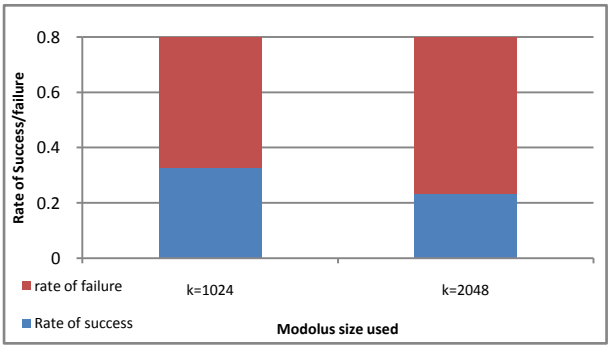

Fig 7. Comparison of success rate for an attacker in ID-mRSAA setting for different k size.

From the table in above it can be summarized that by using the modulus size larger as possible the probability of success to can be large for the given system. This means that the rate of success for an attacker will decrease with increase in mod size.

6. EXPERIMENTAL ANALYSIS

When plain RSA is used for encryption, the public encryption exponent is typically a small integer with only a few 1-bits for example 65, 537 as the default public key value for RSA certificates. Therefore encryptions with small exponents are accelerated with special algorithms for modular exponentiation. However, in ID-mRSAA setting, there is no such luxury of choosing special exponents and a typical public exponent is a relatively large integer with (on the average) half of the bits set to 1.

Some trials were run to assess the cost of ID-mRSAA encryption for public keys derived from recipient email address i.e. unique UserID. The encryption was tested using the java program developed on a 2.5GHz Quad Core workstation. During the trials a fixed message digest – "This is a test message" was used and

1. "default" encryption exponent 65, 537 in RSA and
2. 3 other exponents of length 160, 256, 512 bit for each key in ID-mRSAA.

The results are depicted in table given below:

Table 5. ID-mRSAA Encryption: Performance Comparison of Different Encryption Keys.

Keys	**512 bit**	**1024 bit**	**2048 bit**	**4096 bit**
65,537	1 ms	2 ms	4 ms	12 ms
160 bit key	21ms	51ms	86ms	112ms
256 bit key	36ms	79ms	105ms	133ms
512 bit key	54ms	97ms	122ms	149ms

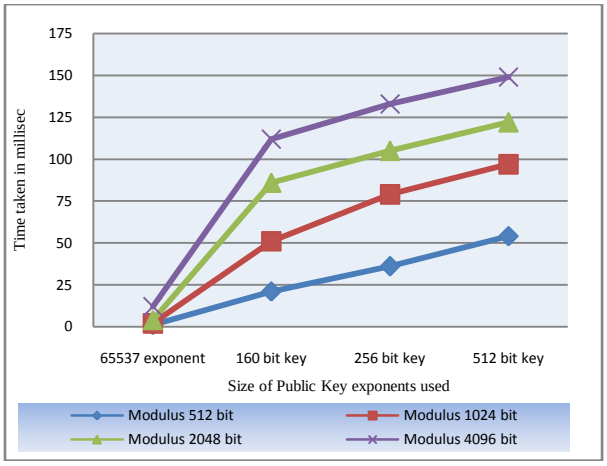

Fig 8. Performance Comparisons of Different Encryption Keys.

From the above graph it can be seen that encryption with a randomized key does introduce additional cost to the user, especially when the modulus size grows. Thus from the above analysis it is clear that the sender in the system must accommodate for the additional cost of encryption while sending (encrypting) the email thereby reducing him to send limited Emails. The time taken for Email operation that included encryption of Email using IDE plug-in and sending the Emails to the recipient was recorded. A fixed sized data of 3MB was used and tested for different "n" user setting to get total time taken to encrypt and send on client side.

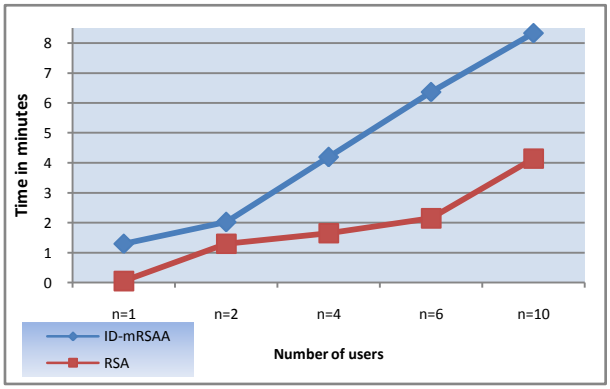

Fig 9. Shows the time at client side for both ID-mRSAA and DKIM (RSA)

Based on the above analysis it can derived that for an ID-mRSAA based Email system some amount of additional cost can be added for sending message to multiple users at a time. By inducing an additional cost for encryption and sending of message a restriction can be imposed on the rate at which the sender sends the bulk message as Spam. Also protection of the integrity of message during transmission from the attacker can be preserved.

7. CONCLUSION AND FUTURE WORK

From the experimentation analysis of the proposed solution for Email system which is based on Identity based mediated RSAA situation of attacker sending unsolicited Emails by obtaining Keys of actual sender and signing the Email using those Key from sender domain and deliberately send Spam mails in the form of Masquerade can be reduced. The possibility of generation of message with signed spoofs of other signer's addresses can be avoided by using this architecture. The message can only be decrypted by intended recipient who has the private key of user PR_{User} . Also by adding an extra cost for encryption limitation to the number of mails to be sent by the sender can be imposed. Thus email security and integrity are provided.

Future work would be dynamic monitoring of q_d/q_e from time-to time in the Key Mediator to check for large values of both encryption/decryption requested operations and take necessary operations based on the situation. Also the performance metric's like Delay and throughput analysis of Email system for different digest size and multiple users and number of mails using this architecture need to be analyzed. Finally the rate of success and Percentage of spam prevention by the system need to be evaluated in future.

8. REFERENCES

[1] L. Cranor and B. LaMacchia, *"Spam!" Communications of the ACM*, vol. 41, no. 8, pp. 74–83, August 1998.

[2] B. Hayes, "*SPAM, SPAM, SPAM, LOVELY SPAM*," American Scientist, vol. 91, no. 3, pp. 200–204, May–June 2003.

[3] S. M. Kerner. *The cost of phishing hits $1.2 billion*. [Online]. Available: http: //www.internetnews.com/ec-news/article.php/3350891. August 2007.

[4] C. C. Zou, W. Gong, and D. Towsley, *"Feedback email worm defense system for enterprise networks,"* University of Massachuset, Technical Report TR-04-CSE-05, April 2004.

[5] The Spamhaus Project. *The definition of spam*. http://www.spamhaus.org/definition.html

[6] Anti-Phishing Working Group. *Phishing archive*. http://www.antiphishing.org/.

[7] Yahoo. *Domainkeys: Proving and protecting email sender identity*. http://antispam.yahoo.com/domainkeys

[8] Meng Wong and Wayne Schlitt.Rfc 4408 - sender policy framework (spf) for authorizing use of domains in e-mail, version 1, April 2006.

[9] William Leibzon. *Message enhancements for transmission authorization*. http://www.metasignatures.org/

[10] Mutual Internet Practices Association. *Certified server validation*. http://mipassoc.org/csv/.

[11] E. Allman ,Sendmail, Inc. ;J. Callas, PGP Corporation; M. Delany M. Libbey Yahoo! Inc. ; J. Fenton M. Thomas Cisco Systems, Inc; Network Working Group*; Request for Comments: 4871* ;May 2007.

[12] D. Eastlake 3rd; Motorola; *RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)*; Request for Comments: 3110; May 2001;

[13] J. Jonsson, B. Kaliski; RSA Laboratories; *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*; Request for Comments: 3447; February 2003.

[14] Johannes Blomer, Alexander May; A Generalized Wiener Attack on RSA.

[15] D. Boneh. Twenty Years of Attacks on the RSA Cryptosystem. Notices of the American Mathematical Society, 46(2):203--213, 1999.

[16] A. Shamir, N. van Someren. *Playing hide and seek with stored keys*; Lectures in computer science, 1998; http://www.simovits.com/archive/keyhide2.pdf.

[17] Sooel Son and Vitaly Shmatikov, The Hitchhiker's Guide to DNS Cache Poisoning,

[18] D. Atkins and R. Austein. Threat Analysis of the Domain Name System (DNS). RFC 3833 (Informational), August 2004.

[19] M. Wiener, "Cryptanalysis of short RSA secret exponents", IEEE Transactions on Information Theory, Vol. 36, pp. 553-558, 1998.

[20] M. Kutylowski, P. Kubiak; M. Tabor, D. Wachnik; *Mediated RSA cryptography specification for additive private key splitting (mRSAA)*; Internet Draft; November 14, 2011.

[21] Xuhua Ding and Gene Tsudik, *Simple identity based cryptography with mediated RSA*, in: The Cryptographers Track RSA Conference, San Francisco, USA, 2003.

[22] Jean Claude Bajard, Heinrich Hördegen ; *Pseudo-Random Generator Based on Chinese Remainder Theorem*;, Advanced Signal Processing Algorithms, Architectures, and Implementations XIX, San-Diego; 2009.

[23] M. Bellare, A. Boldyreva, and S. Micali. *Public-key encryption in a multi-user setting: Security proofs and improvements*. In Preneel [18], pages 259–274.

[24] Xuhua Ding and Gene Tsudik, *Simple identity based cryptography with mediated RSA*, in: The Cryptographers Track RSA Conference, San Francisco, USA, 2003.

[25] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. *RSA-OAEP is secure under the rsa assumption*. In Kilian [15], pages 260–274.

[26] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. *RSA-OAEP is secure under the rsa assumption*. In Kilian [15], pages 260–274.