Shabnam Parveen, Priyanka Gandhi / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.873-876

Enhanced Hybrid Encryption Algorithm for Security of Network

Shabnam Parveen*, Priyanka Gandhi**
*(Department of Computer Science, Kurukshetra University, Kurukshetra)
** (Department of Computer Science, Kurukshetra University, Kurukshetra)

ABSTRACT

One of the biggest problems in cryptography is the distribution of keys. Suppose you live in the United States and want to pass information secretly to your friend in Europe. If you truly want to keep the information secret, you need to agree on some sort of key that you and he can use to encode/decode messages. But you don't want to keep using the same key, or you will make it easier and easier for others to crack your cipher. But it's also a pain to get keys to your friend. If you mail them, they might be stolen. If you send them cryptographically, and someone has broken your code, that person will also have the next key. If you have to go to Europe regularly to hand-deliver the next key, that is also expensive. If you hire some courier to deliver the new key, you have to trust the courier, et cetera. One solution is provided by digital signature as digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. Both security algorithms have their flaws and benefits and they both prove to handy in long term with some compromise. In our research we are proposing a combination of both DSA and RSA as a hybrid link for both protocols. We have enhanced the hardness in security by combining the RSA and DSA encryption algorithms by adding some more security codes.

Keywords **–Authentication, Cipher, Cryptography, Digital Signatures, DSA, Privacy , Private Key, Public Key, Public-key cryptosystems, RSA.**

I INTRODUCTION

The world is becoming more interconnected with the advent of the Internet and new networking technology. There is a large amount of personal, commercial, military, and government information on networking infrastructures worldwide. Network security is becoming of great importance because of intellectual property that can be easily acquired through the internet. The privacy requirements normally encountered in the traditional paper document world are increasingly expected in Internet transactions today. Secure digital communications are necessary for web-based e-commerce, mandated privacy for medical information, etc. In general, secure connections between parties communicating over the Internet is now a requirement.

The national and societal view of the role of encryption will be one of the defining issues for our culture in the twenty-first century. Encryption is cited by Michael Baum, chairman of the Information Security Committee of the American Bar Association, as "an enabling technology that provides companies, their business partners, customers and end users with the ability to get the information and service they need much faster and more securely."1 Ubiquitous digital communications will result in either a secure environment to conduct personal affairs and electronic commerce or a Kafkaesque world lay bare by digital fingerprints indicating our every transaction and thoughts. The information age has brought to light many important issues: protection of privacy, infrastructure protection, law enforcement, national security, and economic competitiveness. In a democracy, it is important to have a public debate on these issues and to ensure that our laws adequately address the issue of cryptography to carry us forward into the twenty-first century.

Encryption is the vital part of information sharing so we will put our efforts into encryption area for RSA algorithm with DSA so that we can make security harder by giving a hybrid algorithm in Asp.net with c#.

II THE BASIC IDEA

The basic idea came to us on the 4th November 1996 . How to protect the Data sending and how the modern attackers break the RSA and DSA. We underlying some algorithms that can be used for Data encryption and signing.

2.1 Using hash functions

It would be more prudent to send a hash of the message. Provided that the hash function is truly one-way (technically, pseudorandom), this will not leak information.

Protocol:

I. Select a random codeword*X*

II. Form its hash$Y=h(X)$

III. Construct a message *M.*

IV. Compute $Z = h(M)$ and publish it anonymously

Now send the message along with the Hash Function. At the Receiver side when receiver decrypt the message it will have message and Z. At receiver side message and its hash generated and matched, if it is correct then accepted by receiver otherwise rejected. This might appear to be only a slightly more technological version of the protocols already used by various liberation groups. It still suffers from the serious problem that, in the face of a capable motivated opponent, the password is only one-time; once it has been revealed, and is known to the newspapers and the police, any journalist or policeman could in theory masquerade as the rebel leader.

2.2 The Guy Fawkes Protocol[5]

Our critical innovation is to introduce a chaining mechanism that lets us bind code words to messages in a way that provides not just authentication but also non repudiation. It also allows the secret codeword to be refreshed, so that the system can be used an arbitrary number of times.

The basic idea is that, at each round of the protocol, firstly commit to a string consisting of (codeword, message, [hash of next codeword]) by publishing a hash of it. This commitment binds the message to the codeword and its successor.

Then reveal the value of this string, proving our knowledge of the codeword and thus authenticating ourselves.

Formally, define the protocol by induction. Suppose that one have published *Zi* followed by the message *Mi* containing $h(Xi)$, where secret codeword is currently *Xi*. One wish to authenticate the message $Mi+1$. We follow the
following protocol:

I. Select a random codeword $Xi+1$

II. Form its hash $h(Xi+1)$

III. Compute $Zi+1 = h(Mi+1; h(Xi+1); Xi)$ and publish it.

IV. Reveal $Mi+1; h(Xi+1)$ and Xi

The first codeword needs to be bootstrapped by some external mechanism; in most applications, this would be a conventional digital signature or an out of band authentication, perhaps using a conventional CA.

In this Guy Fawkes protocol, the objective is to associate a single act of authentication with a stream of future statements rather than a stream of future events. Functionally, the difference is that while the format of all the digital coins is known at the time they are signed, the future statements that wish to authenticate may not be. So it would not be sufficient to simply use a hash chain (as in S/Key) as a set of one-time passwords for authenticating political statements. Anyone who was tapping the line when the statement and password were sent to the newsroom could alter the statement; and statement in the newsroom could also substitute messages at will.

In other words, the broadcast commitment step has the critical effect of providing non repudiation, and gives the Guy Fawkes protocol the same effect as a digital signature. This could have been encrypted and published, with the key made known after the event. So we might ask whether there is anything to signature other than secure association. After all, in the conventional model, a digital signature sets up a secure association between something that has been signed at an arbitrary time, and an authentication instance which may have involved showing a passport to a certification authority.

Explanation:

Firstly, Key Generation Algorithm run in this we Choose two distinct large random prime numbers *p* and *q*. Then Compute $n = p\,q$, where *n* is used as the modulus for both the public and private keys. Then Compute the totient function: $\phi(n) = (p-1)(q-1)$ Choose an integer *e* such that $1 < e < \phi(n)$, and *e* and $\phi(n)$ share no factors other than One, where *e* is released as the public key exponent. Compute *d* to satisfy the congruence relation $d \times e = 1$ modulus $\phi(n)$; *d* is kept as the private key exponent. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret. After distributing the Public and Private keys main Encryption Algorithm starts in that we calculate cipher text as $C=P^e \pmod n$.
After RSA algorithm we create a *message digest* of the information to be sent by using hash function. Uses her *private* key (n, d) to compute the signature, $s1 = m^d \bmod n$. Sends this signature *s* to the recipient B.
Receiver receives the original Message, Message Signature s1, Public-Private key pair. Decrypt the message by its private key. $P=C^d \pmod n$. Uses her *public* key (n, d) to compute the signature, $s2 = m^d \bmod n$. Match s1 and s2 if s1=s2 then accept the message otherwise discard message.

III. Algorithms Used

A. Sender Side algorithms

I. Key Generation Algorithm

1. Choose two distinct large random prime numbers *p* and *q*.
2. Compute $n = p\,q$, where *n* is used as the modulus for both the public and private keys
3. Compute the totient function: $\phi(n) = (p-1)(q-1)$
4. Choose an integer *e* such that $1 < e < \phi(n)$, and *e* and $\phi(n)$ share no factors other than One, where *e* is released as the public key exponent.

5. Compute *d* to satisfy the congruence relation $d \times e = 1$ modulus ϕ *(n)*; *d* is kept as the private key exponent.
6. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.

II. Encryption Algorithm:

7. Calculate cipher text as $C=P^e \pmod n$.

III. Digital Signing Algorithm:

8. Creates a *message digest* of the information to be sent by using hash function.
9. Uses her *private* key (n, d) to compute the signature, $s1 = m^d \mod n$.
10. Sends this signature *s* to the recipient,B.

B. Receiver side algorithms

1. Receiver receives the original Message, Message Signature s1, Public-Private key pair.
2. Decrypt the message by its private key. $P=C^d \pmod n$.
3. Uses her *public* key (n, d) to compute the signature, $s 2= m^d \mod n$.
4. Match s1 and s2 if s1=s2 then accept the message otherwise discard message.

C. Hash Function

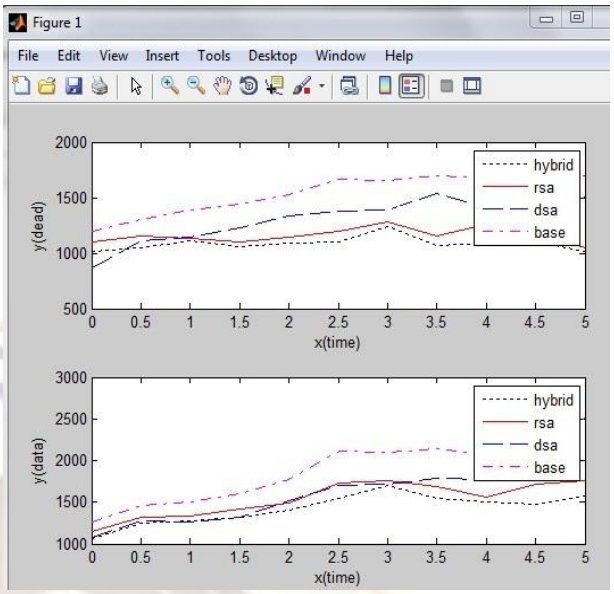
1. Declare character 'str' of unsigned long type.
2. Declare and initialize hash of unsigned integer type
3. Unsigned int hash = 0;
   int q;

```
while (q = str+1)
hash =hash + q;
```

IV. PROBLEM FORMULATION

As Encryption became a vital tool for preventing the threats to data sharing and tool to preserve the data integrity so we will focusing on security enhancing by enhancing the level of encryption in network. For required research we are working on well known encryption algorithms "RSA and DSA". We are proposing the hybrid algorithm for RSA Algorithm with digital signatures with a small case study of applications for this hybrid algorithm.

V. EXPERIMENTAL ANALYSIS



EXPLANATION

First graph is showing Power Efficiency. In this the dead time of base is more and is rising with the time that means the survival period of sensors is less than every other protocol. And the next is DSA its dead time is less than base and is more than RSA and RSA having dead time less than DSA but more than hybrid of DSA and RSA. Hybrid of RSA and DSA is having least dead time that means survival time of sensors is more and is more balanced than DSA, RSA, and base. Second graph is showing Safe Data. In this the base (without protocol) data is least safe. And in RSA data is more safer than base but less than DSA. DSA having data more safer than RSA and less than hybrid of DSA and RSA. Hybrid of DSA and RSA is having more safe data than every other.

VI. CONCLUSION

Present study will reflect the importance of security in network and will provides the better encryption technique for currently implemented encryption techniques. It will explore how to tackle with the threats to data integrity and for safe passage of data from one node to another. This research will provides the great feasibility for authentication process improvement for security in network. Our research will provide a robust algorithm which contains benefits of both RSA and DSA. Our research can easily implemented to different types of networks easily as it is very independent of network lay out. Stronger and complex the encryption, better the results is our motto for our research.

REFERENCES

[1] Federal information processing standards publication," Digital Signature Standard (DSS)", june 2009, Information Technology Laboratory .

[2] Hinek M. Jason.,"On the Security of Some Variants Of RSA", Waterloo, Ontario, Canada : s.n., 2007.

[3] Kitsos, N. Sklavos and O.Koufopavlou," An Efficient Implementation of the Digital Signature Algorithm", VLSI Design Laboratory, Electrical and Computer Engineering Department, University of Patras. Patras, Greece.

[4] R.L. Rivest, A. Shamir, and L. Adleman," A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" in 2006.

[5] oss Anderson1, Francesco Bergadano2, Bruno Crispo2, "A New Family of Authentication Protocols", ACM SIGOPS Operating Systems Review. New York, 2008, pp.9-20.