IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 6, Ver. II (Nov – Dec. 2015), PP 50-52
www.iosrjournals.org

Using NP Problems to Share Keys in Secret-Key Cryptography

Yashasvini Sharma
Department of Computer Science, RGPV, INDIA

Abstract: *Public key cryptography has now become an important means for providing confidentiality by its use of key distribution, in which users can do private communication with the help of encryption keys. It also provides digital signatures which allow users to sign keys to verify their identities. But public key cryptography has its own shortcoming regarding to high cost in keys distribution and excessive computation in encoding and decoding it.*
Whereas private key can omit all above problems but only if we can find a way to share private key confidentially.
This research presents an innovation, which can be our future approach, using technology so-called NP problems, of sending or sharing keys to the receiver without any need of the third party. This will provide an open idea where sender and receiver can share any key for any number of times for encrypting data confidentially that also helpful in overcoming problem of brute force attack.
Keywords: *Public key cryptography, Secret-key cryptography, NP problems, brute force attack, Symmetric key, Plain text*

I. Introduction

1. Secret-key cryptography, also known as symmetric-key cryptography, employs identical private keys for users, while they also hold unique public keys. "Symmetric key" refers to the identical private keys shared by users. But the main problem was" private keys must still be distributed in a confidential mode".

1.1 Strengths

The private keys used in symmetric-key cryptography are robustly resistant to brute force attacks. While only the one-time pad, which combines plaintext with a random key, holds secure in the face of any attacker regardless of time and computing power, symmetric-key algorithms are generally more difficult to crack than their public key counterparts. Additionally, secret-key algorithms require less computing power to be created than equivalent private keys in public-key cryptography.

1.2 Weaknesses

The biggest obstacle in successfully deploying a symmetric-key algorithm is the necessity for a proper exchange of private keys. This transaction must be completed in a secure manner. In the past, this would often have to be done through some type of face- to-face meeting, which proves quite impractical in many circumstances when taking distance and time into account. If one assumes that security is a risk to begin with due to the desire for a secret exchange of data in the first place, the exchange of keys becomes further complicated.

2. Public-key Encryption

The author's public-key method consists of separate encryption and decryption keys, with users only being able to decrypt an encrypted message if they have the appropriate decryption key. Users will exchange public keys; this transaction does not need to be done in a secure manner because the release of public keys does not threaten the security of any private information.

2.1 Strengths

The asymmetric nature of public-key cryptography allows it a sizable advantage over symmetric-key algorithms. The unique private and public keys provided to each user allow them to conduct secure exchanges of information without first needing to devise some way to secretly swap keys.

2.2 Weaknesses

Keys in public-key cryptography, due to their unique nature, are more computationally costly than their counterparts in secret-key cryptography. Asymmetric keys must be many times longer than keys in secret-cryptography in order to boast equivalent security. Keys in asymmetric cryptography are also more vulnerable to brute force attacks than in secret-key cryptography. There exist algorithms for public-key cryptography that allow attackers to crack private keys faster than a brute force method would require. The widely used and

pioneering RSA algorithm has such an algorithm that leaves it susceptible to attacks in less than brute force time. While generating longer keys in other algorithms will usually prevent a brute force attack from succeeding in any meaningful length of time, these computations become more computationally intensive. These longer keys can still vary in effectiveness depending on the computing power available to an attacker. Public-key cryptography also has vulnerabilities to attacks such as the man in the middle attack. In this situation, a malicious third party intercepts a public key on its way to one of the parties involved. The third party can then instead pass along his or her own public key with a message claiming to be from the original sender. An attacker can use this process at every step of an exchange in order to successfully impersonate each member of the conversation without any other parties having knowledge of this deception.

II. Intended Idea-

NP problems-

In computational complexity theory, NP is one of the most fundamental complexity classes. The abbreviation NP refers to "nondeterministic polynomial time."

Intuitively, NP is the set of all decision problems for which the instances where the answer is "yes" have efficiently verifiable proofs of the fact that the answer is indeed "yes".

We can take an example to understand what this means
If we have equation –> 3x+4=37
Can we find x
Yes, we know if one equation is given with one variable we can find that
3x=33
x=11

It is done in two steps and let us says each step is a unit so it is done in 2 unit means we can find solution in some known time, this is called as polynomial solution.
Now take another question
2x^2+ 4y^5=11274
Can we find answer?

If this is a question there must be solution but we have no idea how to get it.
So we left with 2 methods-

1. By guessing values of x and y randomly
2. By taking one by one value of x and y in a sequence until we get answer.

Well we know that by any approach we are not sure when we will get answer, Its unpredictable, We can get it in 2 minutes, 2 hours to 2 decades or just never…
These problems are called as NP or non polynomial means exponential-- that grows very fast.

But again lets say I tell you that x=15 and y=27. Now of coarse you can put it in equation and equate it in polynomial time and can find if this solution is right or wrong.
This is the beauty and power of NP problem.
Means if I myself design a equation say (29^2+4*76^3)/8=220329
And then tell you as (x^2+4*y^3)/z=220329
Now tell me x, y and z

Can you? I don't think so even for such or much large equation brute force also needs large, very large time. But if I tell you ok x=29 and y=76 then you can calculate it in no time.

So we are going to deal with this IDEA.

III. Suggested Method

So in here we are taking easy example to understand-
Let's say sender RAMA want to share a key with receiver SHYAAM which is 123

Now let's build an equation-
{[(123*3)/2+1] +10}/5=39
Now make an equation by hiding 123
Put x=123
(3x+21)/10

Now send 39 to the sender
Sender SHYAAM will make his own equation with this value
{(39+11)/2}*5=125
Now SHYAAM make equation by putting p=11, q=5, r=2
(39q+pq)/r=125 and send to RAMA

Obviously even for this small equation p, q, r can have many value that satisfy equation so trespasser has no idea what can be exactly p, q, r

RAMA place value of 39
{[{(3x+21)/10} + p]/r}*q
3xq+41q+10pq=1500r and send this to SHYAAM

Obviously again by seeing 3xq+41q+10pq=1500r equation one cannot easily find out what new we have placed in (39q+pq)/r=125, moreover it is an easy example where we can predict 125*20=1500 obviously. Actual equation shall be much large, complex and unpredictable. Easy values are used to make things easier. This is hard to find that what have we placed to make such equation and if we make much harder equation this will be next to impossible.
Now SHYAAM will simply place value of p, q, r and will get x=123 easy

In Points

RAMA (sender)	SHYAAM(receiver)
----------→39	
	←---------- (39q+pq)/r=125
----------→3xq+41q+10pq=1500r	
	Calculate x=123

It is simple example using very simple operations, but in reality one can perform heavy operations by using very large numbers to make guessing factor complicated.

IV. Conclusion

Private Key cry[tography is much better than public key cryptography in terms of cost, resistance against brute force attack, selectiveness of password, less complicated operations and robustness because you can change or share a different key for different plain text, but has a biggest problem how to share key? The present work gives a method to deal with the same.

Its biggest benefit is sender and receiver can share different key for different communication and not like public key where you have only one pair of public and private key and if it get stolen, you will face lots of problem which is removed in this case.

References

[1] Encryption: Strengths and Weaknesses of Public-key Cryptography Matt Blumenthal Department of Computing Sciences Villanova University, Villanova, PA 19085 CSC 3990 – Computing Research Topics matthew.blumenthal@villanova.edu
[2] http://www.cgi.com/files/white-papers/cgi_whpr_35_pki_e.pdf
[3] Private and Public Key Cryptography and Ransomware December 2014 Authored by:Ted Fischer Center for Internet Security (CIS) Security Operations Center (SOC) Analyst
[4] ISSN:2249-5789 Prashant Kumar Arya et al , International Journal of Computer Science & Communication Networks,Vol 5(1),17-21 Comparative Study of Asymmetric Key Cryptographic Algorithms
[5] http://searchsecurity.techtarget.com/definition/asymmetric-cryptography
[6] A Comparison of a Public and a Secret Key Cryptosystem, Adam Donlin, SE4H. 29th February, 1995.