Enhanced Public Key Algorithm for Improved Security

Dilpreet Singh
Computer Science Department,
PEC University of Technology
Chandigarh, India

Trilok C. Aseri
Phd,Computer Science Department,
PEC University of Technology
Chandigarh, India

ABSTRACT
The area of security has become more important to personal computer users, organizations, and the military due to the continuous advancements in technology and computing resources. With the advent of Internet and ease of data sharing among users, the integrity of the data has become quite an issue. Moreover, with better computational resources the number of threats also increases. This further pushes for a stronger and better security measures. The entire field of security is vast and in an evolutionary stage. In this paper, we have re-engineered a well known and widely used public-key algorithm of Diffie-Hellman. The paper also provides an insight into other public-key algorithm. In our work, we have proposed some amendments in DH so as to improve the secret key values. Experimental results show that our proposed amendments in DH give significant improvements.

General Terms
Algorithm, Security,

Keywords
Cryptography, Security goals, Security threats.

1. INTRODUCTION
From the past few years the interest in field of computer security has significantly grown and this is pretty much evident in every possible business and industry. Importance of understanding security is instrumental for well functioning markets. The secrecy and integrity of information needs to be there for procuring high benefits and safe transactions. To avoid the unauthorized access to information, both the entities performing transactions need to agree on some secret which the intruder is unaware of [1] [2].
The art and science of transforming information, such that it looks gibberish to unrecognized and unauthorized party is called cryptography. With the continuous evolution of technology and rapid expansion of networking, the amount of data in the near future will also be incredibly large. Thus cryptography will be effectively and broadly applied in all computing areas such as electronic commerce, banking transactions, and trade contracts etc [3].

1.1 Security Goals
The three essential goals [4] of security are defined as: First, Confidentiality, which means that unauthorized people cannot access our secret information. Second, Integrity, which means that the data has not been modified and the data did not come from an imposter. Third, Availability, which means that the resources are ready to use when we need them. Non-availability means denial of service

1.2 Security Threats
A threat is a possibility that an attack can be performed [5]. Security threats to any system are Disclosure, or unauthorized access to information; Deception, or accepting incorrect information; Disruption, or hindering the actual operation; Usurpation, or gaining unauthorized access and control of system.

1.3 Need for Security
The advancements in technology, the Internet, and information sharing has proved to be both a boon and a bane. One of the negative impacts was the large increase in new information threats. Many of these threats pave way for exploiting confidential information being stored by companies and businesses. The Unauthorized access to confidential information is a result of weak security measures. The cost to secure information is increasing with the rising threats and vulnerabilities [6].
Few of the many reasons, as to why we need security is to ensure that information can be accessed by only those who are recognized and authorized, to ensure that the information cannot be modified by intruder and also to make sure that the information and resources are available when they are required.
The remainder of this paper is organized as follows. Section 2 presents some idea on public key cryptography. Section 3 presents problem formulation. In Section 4, experimental results are shown. Section 5, concludes the paper with some future research directions.

2. PUBLIC KEY CRYPTOGRAPHY
The art and science of safeguarding information by converting it into a form which looks gibberish to unintended/unauthorized user is called cryptography. Public key cryptography also known as asymmetric cryptography is a classification of cryptography.
In a public-key cryptosystem both communicating persons require two keys, one key is public and second key is private. If one person, say Bob, wants to communicate with other person named Alice, he recovers the public key of Alice and uses the key to encrypt message, and then sends the encrypted message to Alice. Only Alice can decrypt the received message using her private key. In any case the third person or intruder should not be able to recover the secret key by knowing the public key [7].
Public-key cryptography is a new concept. It allows Alice and Bob to exchange secret keys securely and efficiently over public networks without sharing prior secrets [8].
Public key cryptosystems are classified as:

2.1RSA: Rivest, Shamir, Adleman
The RSA is public-key algorithm which is widely used for security and has enjoyed very wide applicability in cryptography. It was published in 1978 and involves key generation, encryption and decryption steps.

Key Generation [9]:

- Choose two large prime numbers p and q, $p \neq q$.
- Calculate $n = p \times q$.
- Calculate $\varphi(n) = (p-1)(q-1)$.
- Select e, so that $gcd(e, \varphi(n)) = 1, 1 < e < \varphi(n)$.

- Calculate d, so that $d \times e \, mod \varphi(n) = 1$.
- Obtain Public key as {e, n} and Private key as {d, n}.

Although 'n' is made public, the factors 'p' and 'q' are known only to concerned entities say Alice and Bob [10].
Encryption: To encrypt a message use the formula $C = M^e \, mod \, n$, where 'C' is ciphertext, 'M' is plaintext message and 'e' is the public key.
Decryption: To decrypt a message use the formula $M = C^d \, mod \, n$, where 'C' is ciphertext, 'M' is plaintext message and 'd' is the private key.

2.2DSA: Digital Signature Algorithm

The DSA is a public key signature scheme which was proposed by U.S. National Institute of Standards and Technology (NIST) in 1991.It became first digital signature standard (DSS) to be recognized by any government [11].
Key generation in DSA:

- Choose p as a large prime number of bit size 512-1024.
- Choose q which is a prime divisor of $p - 1$ of bit size 160.
- Select a generator g of the unique cyclic group of order q in Z^{xp}.
- Select a random integer x such that $1 \leq a \leq q - 1$.
- Calculate $y = g^x \, mod \, p$.

The public key is p, q, g, y, and the private key is x.

Message signing in DSA:
In order to sign a message 'm', the signer selects a new integer k, $1 < k < q$, and computes $r = (g^k \, mod \, p) mod \, q$, and $s = k^{-1}(h(m) + xr) mod \, q$. The signature for message 'm' is a pair (r, s) [12].

Signature Verification:
The verification of signature is done by using the formula $r = \left(g^{h(m)w} \, y^{rw} \, mod \, p\right) mod \, q$, where $w = s^{-1}$.

2.3DH: Diffie-Hellman

The Diffie–Hellman key agreement algorithm [13] is one of the most popular and widely used method used by two communicating entities, say Alice and Bob, to agree on a secret key over an insecure communication medium without pre-sharing any information.
The algorithm proceeds as follows: The two communicating entities say Alice and Bob select a prime number 'q' and its root known as generator 'g' [8].

- Alice selects a positive number X_a and calculates $Y_a = g^{X_a} \, mod \, q$.
- Similarly Bob selects a positive number X_b and calculates $Y_b = g^{X_b} \, mod \, q$.
- Alice sends Y_a (her public key) to Bob. And Bob sends Y_b (his public key) to Alice.
- Alice calculates $K_a = {Y_b}^{X_a} \, mod \, q$ and Bob calculates $K_b = {Y_a}^{X_b} \, mod \, q$.
- Therefore, Alice and Bob now share the same secret key $K = K_a = K_b$.

3. PROBLEM FORMULATION

The term security has become a latest and an utmost requirement for any organization whether it is personal or business level. With the rapid and ever-increasing computing power of the systems which doubles approximately every 18 to 24 months [14] [15], any field where data sharing, data storage and data usage is carried out demands for highest level of security. As the need to thwart the intruders and to safeguard our information has become a vital necessity, we have proposed some amendments in the well known public key cryptosystem "Diffie-Hellman". We are proposing the method to extend the security level by improving the key value sender and receiver generate simultaneously. Also, a simple encryption/decryption will be carried out using this improved key.

3.1Proposed Work

Our proposed work for key improvement is as follows. At the beginning, the two communicating entities choose a large prime number (q) and its primitive root (g), also known as generator. Then the sender and the receiver choose a secret number and calculate their public keys. Then they exchange their public keys with each-other and do come calculations to generate their secret keys. This secret will be same to both entities. Then sender and receiver calculate a value which is '2 to the power of secret key'. Lastly, in the newly calculated value both entities add the initially chosen prime number (q). As the secret key generated earlier is known only to sender and receiver, the value generated with performing '2 to the power of secret key' will be greater and unknown to third party. Moreover, as the secret key value is unknown to intruder, performing the addition of prime number, further makes it more secure and creates randomness, because even if the value of 'q' is public, the secret key value is unknown and intruder will not be able to deduce as to which value the prime number is added.
Further, the encryption and decryption is performed by opening file and reading each character in file, adding the new improved secret key value to each character value and then save file in a location. To decrypt the file the encrypted file is selected and opened, read each encrypted character, and subtract the new improved secret key value from each character value and then decrypted file is saved at a location.

3.2Methodology

The proposed work proceeds in the step by step fashion as follows:
The two communicating entities say Alice and Bob select a prime number 'q' and it root known as generator 'g'.

Sender:

1. Sender selects a positive number X_a.
2. Sender calculates its public key as $Y_a = g^{X_a} \, mod \, q$.
3. Sender gets the public key of receiver by exchanging public keys and calculates the secret key as $K_a = {Y_b}^{X_a} \, mod \, q$.
4. Sender calculates $P_a = 2^{K_a}$, where P_a is any variable to store value.
5. Sender performs the addition of prime number (q) to value in step 4, as $P_a = P_a + q$.

Receiver:

1. Receiver selects a positive number X_b.
2. Receiver calculates its public key as $Y_b = g^{X_b} \, mod \, q$.
3. Receiver gets the public key of sender by exchanging public keys and calculates the secret key as $K_b = {Y_a}^{X_b} \, mod \, q$.

4. Receiver calculates $P_b = 2^{K_b}$, where P_b is any variable to store value.
5. Receiver performs the addition of prime number (q) to value in step 4, as $P_b = P_b + q$.

Therefore, Sender and Receiver now share the same secret key $P_a = P_b$, which will be larger and stronger than earlier secret key of K_a and K_b, which are also same.
The flowchart of proposed algorithm is depicted below.

4. EXPERIMENTAL RESULTS

We have used Microsoft Visual Studio 2010 with C#.Net environment to generate results of our proposed algorithm. Comparison is done based on a secret key value which is generated simultaneously by both entities without pre-sharing any information and the time both algorithms take to generate keys. We have done the analysis of both the classical DH and our proposed DH on the parameters of Average Times-Larger and Average Time Difference. Average Times-Larger indicates how many times the new-key values are larger than the old-key values. Times-Larger is calculated by division of new value with old value. It has no unit as it is a ratio of two quantities. Average Time Difference indicates the average difference between the calculation times of old and new key values. Its unit is microseconds (µs). We have shown the analysis of both new-key and old-key values in graphical form.
Tables 1, 2, 3 below show the values that are used in this experiment. 'Times Larger' value is calculated by division of new values by old values. Tables 1, 2, 3 depict the generation of secret key values.

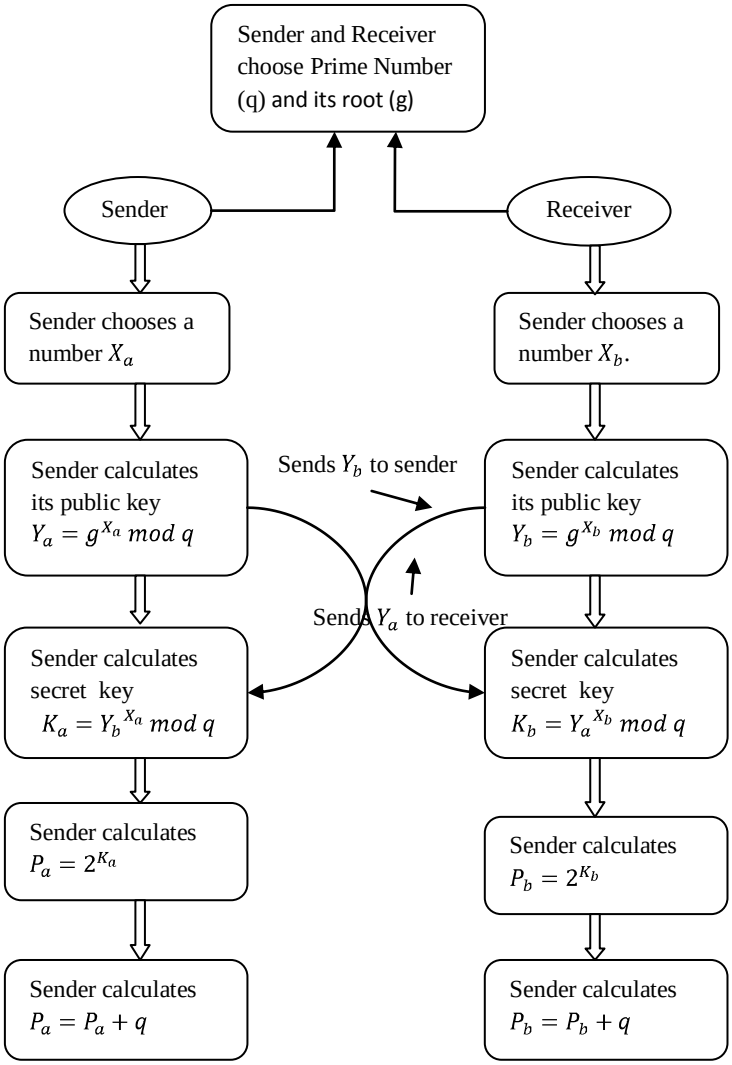

Fig 1: Flowchart of Proposed Algorithm

Table 1 Times-Larger and Secret Key Values with Prime =7 and Generator =3

Prime (q)	Generator (g)	Sender Choice	Receiver Choice	Old Value	New Value	Times Larger
7	3	3	5	6	71	11.833
7	3	2	4	2	11	5.5
7	3	30	12	1	9	9
7	3	8	11	4	23	5.75
7	3	7	13	3	15	5
				Total Old	Total New	TL= TN/TO
				16	129	8.0625

Table 2 Times-Larger and Secret Key Values with Prime =13 and Generator =2

Prime (q)	Generator (g)	Sender Choice	Receiver Choice	Old Value	New Value	Times Larger
13	2	4	6	1	15	15
13	2	7	3	5	45	9
13	2	11	8	3	21	7
13	2	14	13	4	29	7.25
13	2	5	5	2	11	8.5
				Total Old	Total New	TL= TN/TO
				15	127	8.466

Table 3 Times-Larger and Secret Key Values with Prime =19 and Generator =2

Prime (q)	Generator (g)	Sender Choice	Receiver Choice	Old Value	New Value	Times Larger
19	2	7	2	6	83	13.833
19	2	8	2	5	51	10.2
19	2	12	3	1	21	21
19	2	5	11	2	23	11.5
19	2	1	2	4	35	8.75
				Total Old	Total New	TL= TN/TO
				18	213	11.833

Total Old (TO) in the tables 1, 2, and 3 depicts the summation of all five secret key values that were generated with old DH.
Total New (TN) in the tables 1, 2, and 3 depicts the summation of all five secret key values that were generated with enhanced DH.
TL in tables 1, 2, and 3 is an abbreviation for Times-Larger. It is the ratio of Total New (TN) values to Total Old (TO) values. It depicts how many times, the total of new values is larger than the total of old values.
The comparison for the values in Tables 1, 2, and 3 is shown in graphical form in the Figures 2, 3, and 4 respectively. From the graphs we observed that the plot for the 'New values' is higher than the 'Old values'. It is evident from the graph plot of 'Times Larger' which is always above the corresponding value of both 'New and Old', that values generated by our proposed algorithm are much improved and better than its earlier counterparts. This is because, in our proposed scheme we are performing '2 to the power of old secret key' and then adding the Prime number value to the value obtained by raising '2 to the power of old secret key', which is shown in steps 4 and 5 for both sender and receiver in our methodology.

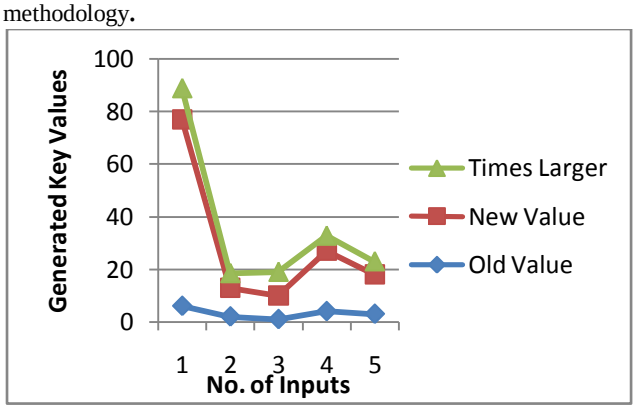

Fig 2: Graph plot of Table 1 values

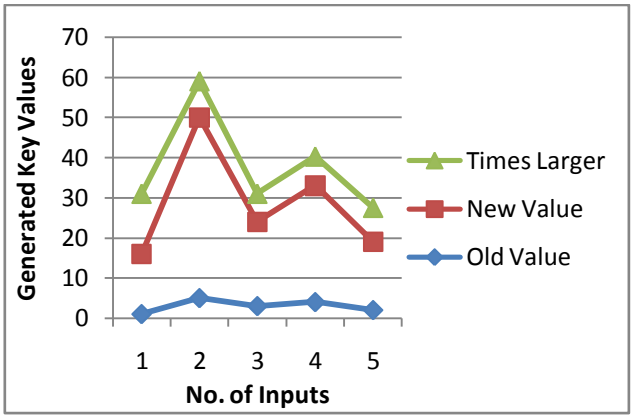

Fig 3: Graph plot of Table 2 values

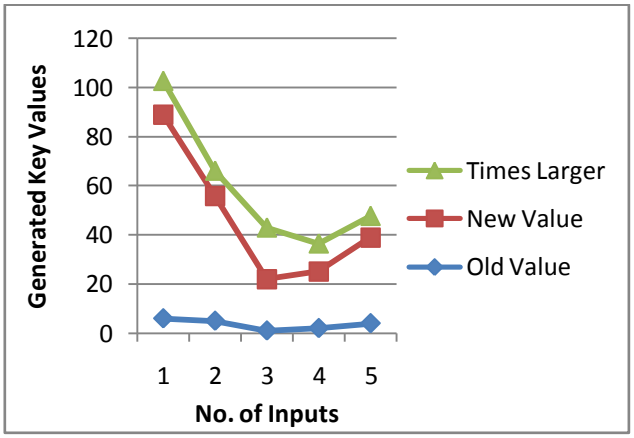

Fig 4: Graph plot of Table 3 values

In Table 4, the average of Times-Larger values of 15 different input values as shown in the Tables 1, 2, and 3 is calculated. The total of Times-Larger comes out to be 149.116, and the average of the resultant value comes out to be 9.941. Thus, according to the result of Table 4 values, the new key values on an average are approximately 10 times larger than the earlier key values. This is a significant rise in key values.

Table 4 Average Times-Larger between 15 Old and New Key Values

Old Value	New Value	Times Larger
6	71	11.833
2	11	5.5
1	9	9
4	23	5.75
3	15	5
1	15	15
5	45	9
3	21	7
4	29	7.25
2	17	8.5
6	83	13.833
5	51	10.2
1	21	21
2	23	11.5
4	35	8.75
	Total	149.116
	Average	149.116/15=9.941

The Table 5 shows the Average Time Difference between the time taken for calculating the old and the new key values. The time values that have been calculated are in microseconds (µs). Time Difference shows the difference between 'Time for Old' and 'Time for New'. Total represents, the summation of all the Time Difference values, and the Average Time Difference is calculated by dividing Total with the number of values taken. The average of time difference comes out to be 24.16 µs. So, on an average our scheme takes 24.16 microseconds more than the previous one.

Though the time values for our proposed algorithm are just minimally larger but considering the significant rise in the new key values i.e. approximately 10 times more than the old ones, this slight increase in time difference which on an average is 24.16 microseconds can be considered as negligible.

Table 5 Average Time Difference between Time Taken for Old Values and New Values

Prime (q)	Generator (g)	Sender Choice	Receiver Choice	Time for Old (µs)	Time for New (µs)	Time Difference (µs)
7	3	3	5	423	444	21
7	3	2	4	384	414	30
13	2	4	6	423	432	9
13	2	7	3	402	440	38
19	2	8	2	389	419	30
19	2	7	2	393	410	17
					Total	145
					Average	145/6 =24.16

The results of simple encryption/decryption scheme are shown in the Figures 5, 6, and 7 below. Figure 5 shows the plain text file named as 'hello.txt'. This plain text file will then be encrypted, and the result of encryption is the garbled text as shown in Figure 6. The encrypted file is named as 'hello (encrypted).txt'.

The encrypted file in Figure 6 will then be decrypted and the result of the decryption is shown in Figure 7. The decrypted file is named as 'hello (decrypted).txt'. The gibberish text in Figure 6 is changed to plaintext in Figure 7. The result for Figure 5 and 7 is identical, i.e. the file is encrypted and decrypted correctly.

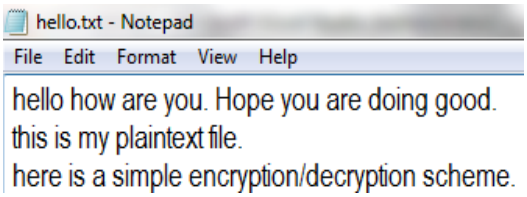

Fig 5: Plaintext

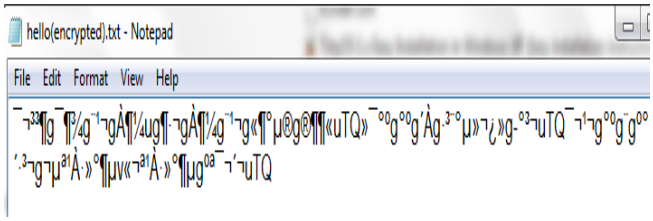

Fig 6: Encrypted Text

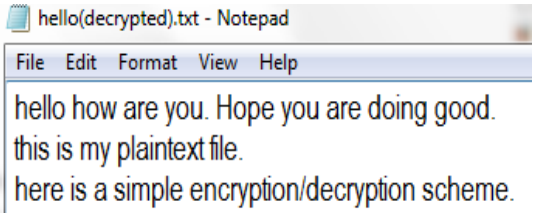

Fig 7: Decrypted Text

5. CONCLUSION

The field of security has been continuously evolving and new techniques to thwart the adversaries from accessing our information are also being discovered rapidly. DH public key algorithm is a well known key exchange algorithm and it is widely used in different protocols such as SSL, IPSec and SSH. Enhancing the security of DH means, improving the security of the protocols where DH is applicable. The security in itself is not permanent. The continuously evolving technology and the increasing computational power raise the probability of a system to be compromised. So, we have proposed some amendments in the widely used algorithm of DH. The amendments that we suggested include: raising the shared key value generated by classical DH, as power of 2; and performing the addition of prime number to the result of "2 to the power of shared secret". These amendments increase the shared key values to approximately 10 times on an average, than the values generated with classical DH algorithm. The calculated average time difference for generating key values which comes out to be 24.16 microseconds (µs) is almost negligible. This is also favorable considering the significant increase in the new key values. We have done the analysis of both the classical DH and our proposed DH on the parameters of Average Times-Larger and Average Time Difference. Our proposed algorithm is definitely an improvement over original DH algorithm. The future work prospective is to minimize the average difference between calculation times.

6. REFERENCES

[1] Crowley, E. 2003. Information system security curricula development. In Proceedings of the 4th conference on Information technology curriculum.

[2] H. Vartiainena, "Simple Model Of Secure Public Communication", Theory and Decision, 2009.

[3] S. Kim, G. Lee, "Secure verifiable non-interactive oblivious transfer protocol using RSA and Bit commitment on distributed environment", Future Generation Computer Systems, 2009.

[4] Pfleeger, C. 2006 Security in Computing. Prentice Hall.

[5] Bishop, M. 2005 Introduction to Computer Security. Addison-Wesley Professional.

[6] Tarte, J. 2003 The Need for Information Security in Today's Economy. Infosec Reading Room. Sans Institute.

[7] F. Leprevost, F. 2005. Public-Key Cryptography: An Overview of some Algorithms. In Optical and Digital Techniques for Information Security, B. Javidi

[8] Wang, J. 2009. Public-Key Cryptography and Key Management. In Computer Network Security: Theory and Practice, J. Wang

[9] Salomon, D. 2005. Public-Key Cryptography. In Coding for Data and Computer Communications, D. Salomon

[10] Raskind, W. and Blum, E.K. 2011. Computer Security and Public Key Cryptography. In Computer Science: The Hardware, Software and Heart of It, E. K. Blum, A.V. Aho

[11] M. Nikodem, "DSA Signature Scheme Immune to the Fault Cryptanalysis", Lecture Notes in Computer Science, 2008.

[12] Z. Shao, "Batch verifying multiple DSA-type digital signatures", Computer Networks, 2001.

[13] B. Tsaban, "Fast generators for the Diffie-Hellman key agreement protocol and malicious standards", Information Processing Letters, 2006.

[14] G. E. Moore, "Cramming More Components onto Integrated Circuits", Proceedings of the IEEE, 1998.

[15] "Moore's Law"- http://en.wikipedia.org