Diffie-Hellman:Key Exchange and Public Key Cryptosystems

Sivanagaswathi Kallam

A paper presented for the degree of Master of science

Math and Computer Science Department
Indiana State University
TerreHaute,IN,USA
9/30/2015

Contents

Abstract

In cryptography, key exchange is a strategy by which cryptographic keys are exchanged between two gatherings and those keys are utilized as a part of some cryptographic algorithms like AES. Utilizing those keys sender and recipient exchange encrypted messages. Public key cryptography gives a secured strategy to exchange secret keys. The key exchange issue is the means by which gatherings exchange the keys or data in a communication channel so that nobody else other than sender and recipient can get those. This paper presents Diffie-Hellman key exchange, a procedure which is one of the first public key cryptographic protocols used to build up a secret key between two gatherings over a frail channel. The protocol itself is constrained to exchange of the keys i.e, we are not sharing data while the key exchange, we are making a key together. We start with implementation of algorithm i.e, by building up a mutual secret between two gatherings that can be utilized for secret communication for exchanging information over a public channel. Having no entity authentication mechanism, protocol is effectively assaulted by the man-in-the-middle attack and impersonation attack in practically speaking. Diffie-Hellman is appropriate for utilization in information communication however is less frequently utilized for information storage or archived over long period of time.

Diffie-Hellman: Key Exchange and public key cryptosystems

Sivanagaswathi Kallam

29 September 2015

1 Introduction

The subject of key exchange was one of the first issues addressed by a cryptographic protocol. This was before the innovation of public key cryptography. In human advancement, people around the world attempted to hide data in composed structure when composing was created. This is presumably the first and primitive type of encryption however is stand out to be a one half portion of cryptography; the other half is the capacity to reproduce the first message from its hidden structure. Cryptography is not concealing a message, so that nobody can discover it, yet rather to leave the message out in public in a manner that nobody aside from the proposed beneficiary comprehends the message. The initially recorded utilization of cryptography for correspondence was by the Spartans who (as right on time as 600 BC) utilized a cipher device called "the scytale" to send secret communication between military officers. The scytale comprised of a wodden baton wrapped with a piece of parchment inscribed with the message. Once unwrapped the material contracted and seemed to contain some unlimited imprints; in any case, when wrapped around another stick of indistinguishable measurements the first content shows up.

Military employments of cryptography were the primary inspirations driving the investigation of cryptography in the past times. It was a secret endeavor, generally attempted by enormous governments, who could conceal all the efforts and make smokescreens important to shroud many individuals, exercises and dynamic analysts.

In those days the huge majority of the cryptosystems were private or symmetric key cryptosystems. In this two clients Alice and Bob select a key ahead of time, which is their private key, then they utilize the key in a private key cryptosystem to convey information over people in general

channel. Military foundations and discretionary workplaces typically have staffs, methods and conventions set up to handle this key choice by two clients and approaches to change these keys intermittently. Secret or private key cryptography is still the foundation of cutting edge cryptography, yet it misses the mark concerning today's necessities. We can clarify this with a case. For suppose the bank called BankAtlantic in Boca Raton, U.S.A needs to exchange a lot of cash to a somewhat obscure bank, State Bank of India, Chittaranjan, India, on the web. It is not possible for the BankAtlantic to officially negotiate with all banks on earth regarding private keys for secret communication and has techniques set up to change those keys occasionally. So other option is to send a trusted courier to Chittaranjan, India with the key before these two banks can exchange the cash. This is a major problem for online business, How can two entities unknown to one another agree upon a key?

The response to the inquiry raised above is the public key cryptography. We investigate public key cryptography regarding the Diffie-Hellman Key Exchange Protocol, which is the most primitive thought behind public key cryptography. In the Diffie-Hellman key exchange protocol, two clients unknown to one another can set up a private however arbitrary key for their symmetric key cryptosystem. Along these lines there is no requirement for Alice and Bob to meet ahead of time, or utilize a safe dispatch, or utilize some other secret means, to choose a key. The Diffie-Hellman key agreement protocol (1976) was the first practical method for setting up a shared secret over an unsecured communication channel.

By what method can two gatherings concur on a secret value when the greater part of their messages may be caught by an eavesdropper? The Diffie-Hellman algorithm accomplishes this, and is still generally utilized. With adequately huge inputs, Diffie-Hellman is exceptionally secure.

2 History

The primary researchers to find and publish the ideas of Public Key Cryptology were Whitfield Diffie and Martin Hellman from Stanford University, and Ralph Merkle from the University of California at Berkeley. As so frequently happens in the experimental world, the two gatherings were working autonomously on the same issue - Diffie and Hellman on public key cryptography and Merkle on public key distribution - when they got to know about one another's work and acknowledged there was collaboration in their methodologies. In Hellman's words: "We each had a key piece of the puzzle keeping in mind it's actual one of us first said X, and another of us first said

Y, and so on, it was the combination of forward and backward between us that permitted the disclosure..”

Figure 1: Ralph Merkle, Martin Hellman, Whitfield Diffie (1977)

The first published work on Public Key Cryptography was in a groundbreaking paper by Whitfield Diffie and Martin Hellman titled ”New Directions in Cryptography” in the November, 1976 version of IEEE Transactions on Information Theory, and which additionally referenced Merkle’s work. The paper depicted the key ideas of Public Key Cryptography, including the generation of digital signatures, and gave some algorithms for execution. This paper revolutionized the world of cryptography research, which had been to some degree controlled up to that point by genuine and saw Government confinements, and aroused many analysts around the globe to take a shot at down to earth executions of an public key cryptography algorithms.

Diffie, Hellman, and Merkle later obtained patent number 4,200,770 on their method for secure.

3 How Diffie-Hellman Works

3.1 What is Diffie-Hellman (DH)?

DH is a mathematical algorithm that permits two PCs to produce an indentical shared secret on both systems, despite the fact that those systems might never have communicated with one another. That shared secret can then be utilized to safely exchange a cryptographic encryption key. That key then encrypts traffic between the two systems.

3.2 Why do we care about DH ?

We care about Diffie-Hellman in light of the fact that it is a standout amongst the most widely recognized protocols utilized as a part of networking today. This is genuine, despite the fact that most by far of the time the user has no clue it is working. DH is usually utilized when you encrypt data on the Web utilizing either SSL (Secure Socket Layer) or TLS (Transport Layer Security). The Secure Shell (SSH) protocol also uses DH.Obviously, in light of the fact that DH is a piece of the key exchange mechanism for IPSec, any VPN based on that technology uses DH too.

Truly a VPN or SSL system could be being used for a considerable length of time without the system head knowing anything about Diffie-Hellman.In any case, I think that an understanding of underlying protocols and processes helps a great deal when trouble-shooting a system.

3.3 The process

Diffie-Hellman is not an encryption mechanism as we regularly consider them, in that we don't commonly utilize DH to encrypt data. Rather, it is a strategy for secure exchange of the keys that encrypt data. DH performs this protected exchange by making a "shared secret" (once in a while called a "Key Encryption Key" or KEK) between two devices. The shared secret then encrypts the symmetric key for secure transmittal. The symmetric key is some of the time called a "Traffic Encryption Key" (TEK) or "Data Encryption Dey" (DEK).

The procedure starts when every side of the correspondence generates a private key (depicted by the letter A in Figure). Every side then produces an public key (letter B), which is a derivative of the private key. The two systems then exchange their public keys. Every side of the correspondence now has its own private key and the other systems's public key (see the area named letter C in the diagrams).

I should also explain the box labeled OPTIONAL: CA Certifies Public Keys. It is not regular, but rather the ability exist inside of the Diffie-Hellman protocol to have a Certificate Authority ensure that the public key is without a doubt originating from the source in which you believe. The reason for this accreditation is to prevent man-in-themiddle (MITM) attacks. These attacks include intercepting both public keys and afterward sending to both beneficiaries the attacker's fake public keys. The "man in the middle" can potentially intercept encrypted traffic, decrypt it, duplicate or alter it, re-encrypt it with the bogus key, and forward it on to its destination.

When the key exchange is finished, the procedure proceeds. The Diffie-

Hellman protocol produces a "shared secret" an indentical cryptographic key shared by every side of the correspondence. Figure portrays this operation in the "DH Math" box.

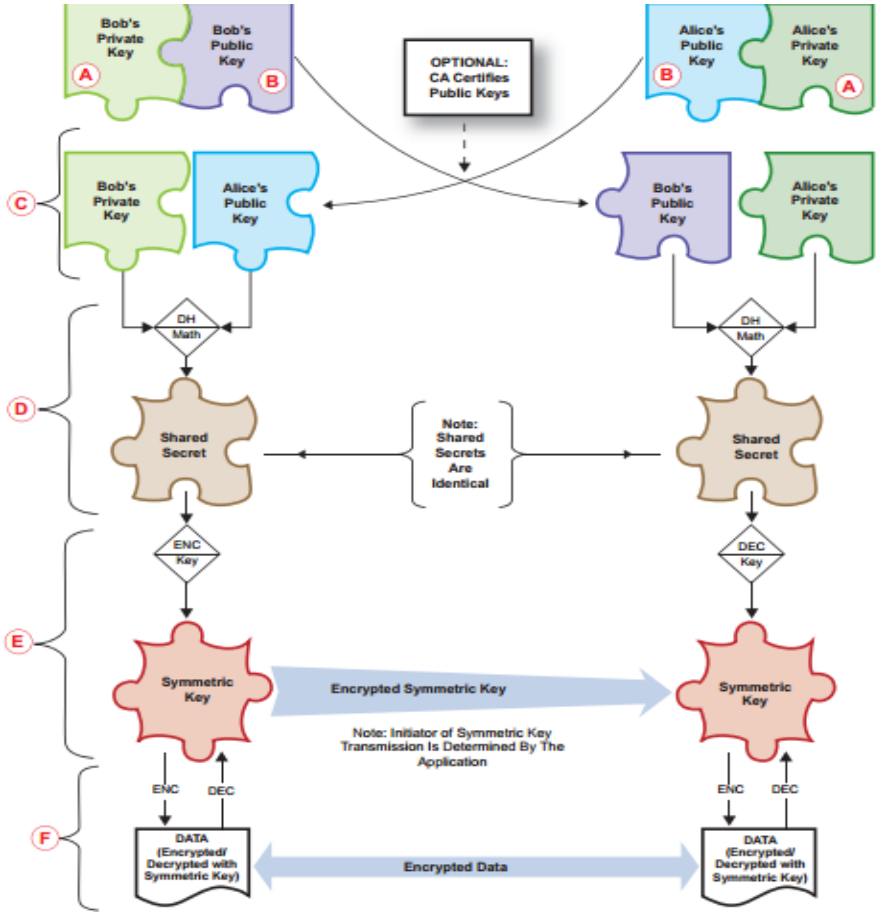

Figure 2: Diffie-Hellman Key Exchange

By running the mathematical operation against your own private key and the other side's public key, you produce a value. At the point when the far off end runs the same operation against your public key and its own private key, that end also creates a value. The critical point is that the two qualities produced are indentical. They are the "shared secret" that can encrypt data between systems.

At this point,the Diffie-Hellman operation could be viewed as complete.The shared secret is, after all, a cryptographic key that could encrypt traffic.In any case, fulfillment as of right now is exceptionally uncommon, on the grounds that the shared secret is an uneven key by its mathematical nature, and all asymmetric key systems are inherently slow. On the off chance that the two

sides are passing next to no movement, the mutual mystery may scramble real information.But any attempt at bulk traffic encryption requires a symmetric key system, for example, DES, Triple DES or Advanced Encryption Standard (AES). In most real uses of the DH protocol (SSL, TLS, SSH, and IPSec specifically),the shared secret encrypts a symmetric key for one of the symmetric algorithms, then transmits it safely, and the inaccessible end decrypts it with the shared secret.

Which side of the correspondence creates and transmits the symmetric key fluctuates. In any case, it is more regular for the initiator of the correspondence to be the one that transmits the key. I ought to additionally call attention to that some kind of arrangement ordinarily strikes choose the symmetric algorithms,the mode of the algorithms (e.g., cipher block chaining, or CBC), hash functions (MD5, SHA-1, etc.), key lengths, refresh rates, and so on. That arrangement is taken care of by the application, and is not a piece of Diffie-Hellman, but it is obviously an important task, since both sides must support the same schemes for encryption for it to function.This additionally indicates why key-administration arranging is so vital – and why poor key administration so frequently prompts failure of systems.

When secure exchange of the symmetric key is finished, information encryption and secure communication can happen (note that passing the symmetric key is the general purpose of the Diffie-Hellman operation). Figure depicts data encrypted and decrypted on every end of the communication by the symmetric key. Changing the symmetric key for expanded security is straightforward as of right now. The longer the time a symmetric key is in use, the less demanding it is to perform a fruitful cryptanalytic attack against it. In this manner, changing keys frequently is important.

3.4 Mathematical Background

The original implementation of Diffie-Hellman protocol uses the multiplicative group of integers modulo p, where p is a prime number and g a primitive root modulo p. What does this mean?Group is a set of elements together with a binary operation that fulfills certain properties. It is called multiplicative to denote that the group operation is multiplication (as opposed to being addition). In Definition 1 we put this into more mathematical form.
Definition 1 A structure (G, $\star$) is a group if the following four properties hold:

1. $\star$ is an operation on G, that is, a rule that joins to each pair (a, b) ϵ $G \times G$ a unique element $a \star b \epsilon$ G.

2. $\star$ is associative, that is, $(a \star b) \star c = a \star (b \star c)$.

3. There exists a neutral element $e \epsilon G$ with respect to $\star$, that is, $\forall\ a \epsilon G : a \star e = e \star a = a$. Note that the neutral element is unique and is quite often denoted by 1 in multiplicative groups.

4. Every element $a \epsilon G$ has an inverse $a^{-1} \epsilon G$ such that $a \star a^{-1} = a^{-1} \star a = e$, where e is the neutral element. Note the inverse of a is quite often denoted by a^{-1} and it is unique.

Let m be a fixed non-negative integer. The set of all residue classes mod m is denoted by Z_m, that is,

$$Z_m = \{\bar{0}, \bar{1}, \cdots, \bar{m-1}\}$$

where $\bar{x} = \{\ y \epsilon Z \mid \exists k\ \epsilon Z : \text{y} = \text{x} + \text{km}\ \}$
Remark 1 The structure $(Z_m, \cdot)$, where

$$\bar{a} \cdot \bar{b} = \bar{ab}$$

is a commutative monoid, that is, $\cdot$ is commutative ($\forall a \forall b : a \cdot b = b \cdot a$), associative and the neutral element exists. The structure $(Zm, \cdot)$ is not necessarily a group, because some residue classes might not have an inverse.
Definition 2 The greatest common divisor of a and b is the largest positive integer that divides both a and b, that is, integer c such that there exists k_1,$k_2\ \epsilon Z$: a = $k_1 c$ and b = $k_2 c$. It is denoted by GCD(a,b) or (a, b). If (a, b) = 1, a and b are relatively prime or coprime.
Theorem 1 An element $\bar{a}$ has inverse in monoid $(Z_m, \cdot)$ iff (a, m) = 1 iff the equation $ax \equiv 1\ mod\ m$ has a solution iff the Diophantine Equation ax-my = 1 has a solution.
Definition 3 The structure $(G, \star)$ is called Abelian group if $\star$ is commutative, associative operation on G and the neutral element and inverses exist.
Definition 4 The elements of the set $Z^\star{}_m = \{\bar{a} \epsilon Z_m \mid (a, m) = 1\}$ are called reduced residue classes modulo m.
Theorem 2 The structure $(Z^\star_m, \cdot)$ is Abelian group.
The group G = $(Z^\star_m, \cdot)$ is the multiplicative group of integers modulo m and it has $\Phi(m)$ elements (its order, ord(G) is $\Phi(m)$), where Φ is Euler's totient function (that gives the number of positive integers less or equal to m that are coprime to m).
Euler's totient function has the following properties:

- $\Phi(1) = 1$
- $\Phi(p^k) = (p-1)p^{k-1}$,for any prime number p and $k \geq 1$

- if m and n are coprime, then $\Phi(mn) = \Phi(m)\Phi(n)$

Remark 2 If p is a prime number,$Z^{\star}{}_p = \{\bar{1}, \cdots, p - \bar{1}\}$ and $\Phi(p) = p - 1$.
Definition 5 A group $(G, \star)$ is a cyclic group, if
$\exists g \, \epsilon G \prec g \succ = \{g^k \mid k \, \epsilon Z\} = \mathrm{G}$
Element g is called a generator of G.
Remark 3 Given a prime number p, the group $(Z^{\star}{}_p, \cdot)$ is always a cyclic group that is generated by a single element. A generator of this cyclic group is called a primitive root modulo p or a primitive element of $Z^{\star}{}_p$.
Definition 6 The order of a group element g, denoted by ord(g), is the smallest positive integer k such that $g^k = e$, where e is the neutral element of the group.
Remark 4 It is easy to see that for a generator g of a cyclic group $Z^{\star}{}_p$, it always holds that ord(g)= ord($Z^{\star}{}_p$)= p-1.

3.5 Steps

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p.

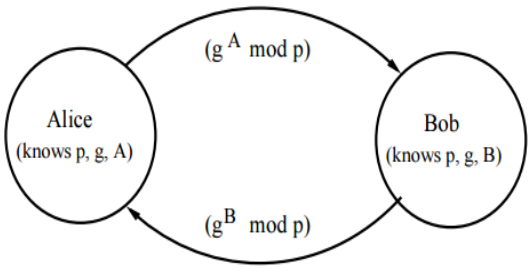

Figure 3: Steps of DH

1. Alice and Bob agree on a prime number p and a base g.

2. Alice chooses a secret integer a, then sends Bob

$$A = g^a \, mod \, p$$

3. Bob chooses a secret integer b, then sends Alice

$$B = g^b \, mod \, p$$

4. Alice computes
$$K_1 = B^a \, mod \, p$$
5. Bob computes
$$K_2 = A^b \, mod \, p$$
6. Alice and Bob now share a secret ie., both Bob and Alice can use this number as their key.

3.6 Diffie-Hellman Correctness and its Proof

1. Alice has computed
$$A = g^a \, mod \, p$$
$$K_1 = B^a \, mod \, p$$
2. Bod has computed
$$B = g^b \, mod \, p$$
$$K_2 = A^b \, mod \, p$$
3. Alice has
$$K_1 = B^a \, mod \, p$$
$$= (g^b)^a \, mod \, p$$
$$= (g^a)^b \, mod \, p$$
$$= A^b mod \, p$$
4. Bod has
$$K_2 = A^b \, mod \, p$$
$$= (g^a)^b \, mod \, p$$
$$= (g^b)^a \, mod \, p$$
$$= B^a \, mod \, p$$
5. Therefore
$$K_1 = K_2$$

4 Illustration with Examples

4.1 Illustartion with colors

The following diagram shows the general thought of the key exchange by utilizing colors rather than an large number. The procedure starts by having the two gatherings, Alice and Bob, agree on an arbitrary starting color that does not should be kept secret; in this example the color is yellow. Each of them chooses a secret color - red and aqua respectively - that they keep to themselves.

The vital piece of the procedure is that Alice and Bob now combine their secret color with their commonly shared color, bringing about orange and blue mixtures respectively, then freely exchange the two mixed colors. At last, each of the two combine the color they got from the partner with their own particular private color. The outcome is a last color mixture (brown) that is indentical with the partner's color mixture.

On the off chance that another gathering had been listening in on the exchange , it is computationally troublesome for that individual to decide the common secret color;in fact, when utilizing large numbers rather than colors, this activity is outlandish for cutting edge supercomputers to do in a sensible measure of time.

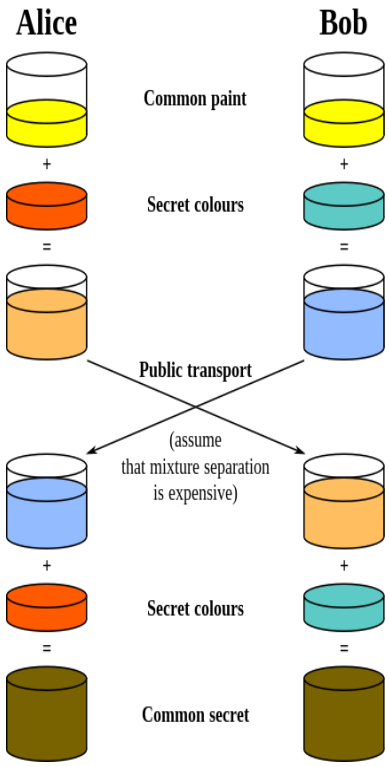

Figure 4: Illustration with Colors

4.2 Secrecy Chart

The chart below depicts who knows what, again with non-secret values in blue, and secret values in red. Here Eve is a eavesdropper;she watches what is sent in the middle of Alice and Bob, however she doesn't alter the contents of their communications.

- g = public (prime) base, known to Alice, Bob, and Eve.

$$g = 5$$

- p = public (prime) modulus, known to Alice, Bob, and Eve.

$$p = 23$$

- a = Alice's private key, known only to Alice.

$$a = 6$$

- b = Bob's private key known only to Bob.

$$b = 15$$

- A = Alice's public key, known to Alice, Bob, and Eve.

$$A = g^a \, mod \, p = 8$$

- B = Bob's public key, known to Alice, Bob, and Eve.

$$B = g^b \, mod \, p = 19$$

Alice

Known	Unknown
$p = 23$	b
$g = 5$	
$a = 6$	
$A = 5^a \bmod 23$	
$A = 5^6 \bmod 23 = 8$	
$B = 19$	
$s = B^a \bmod 23$	
$s = 19^6 \bmod 23 = 2$	
$s = 2$	

Bob

Known	Unknown
$p = 23$	a
$g = 5$	
$b = 15$	
$B = 5^b \bmod 23$	
$B = 5^{15} \bmod 23 = 19$	
$A = 8$	
$s = A^b \bmod 23$	
$s = 8^{15} \bmod 23 = 2$	
$s = 2$	

Eve

Known	Unknown
$p = 23$	a
$g = 5$	b
	s
$A = 8$	
$B = 19$	
$s = 19^a \bmod 23 = 8^b \bmod 23$	

Figure 5: Secrecy Chart

4.3 Examples

4.3.1 Example1

1. Alice and Bob agree on p = 23 and g = 5.

2. Alice chooses a = 6 and sends

$$5^6 \, mod \, 23 = 8$$

3. Bob chooses b = 15 and sends

$$5^{15} \, mod \, 23 = 19$$

4. Alice computes

$$19^6 \, mod \, 23 = 2$$

5. Bob computes

$$8^{15} \, mod \, 23 = 2$$

Then 2 is the shared secret.

4.3.2 Example2

Domain parameters

$$p = 29$$

$$\alpha = 2$$

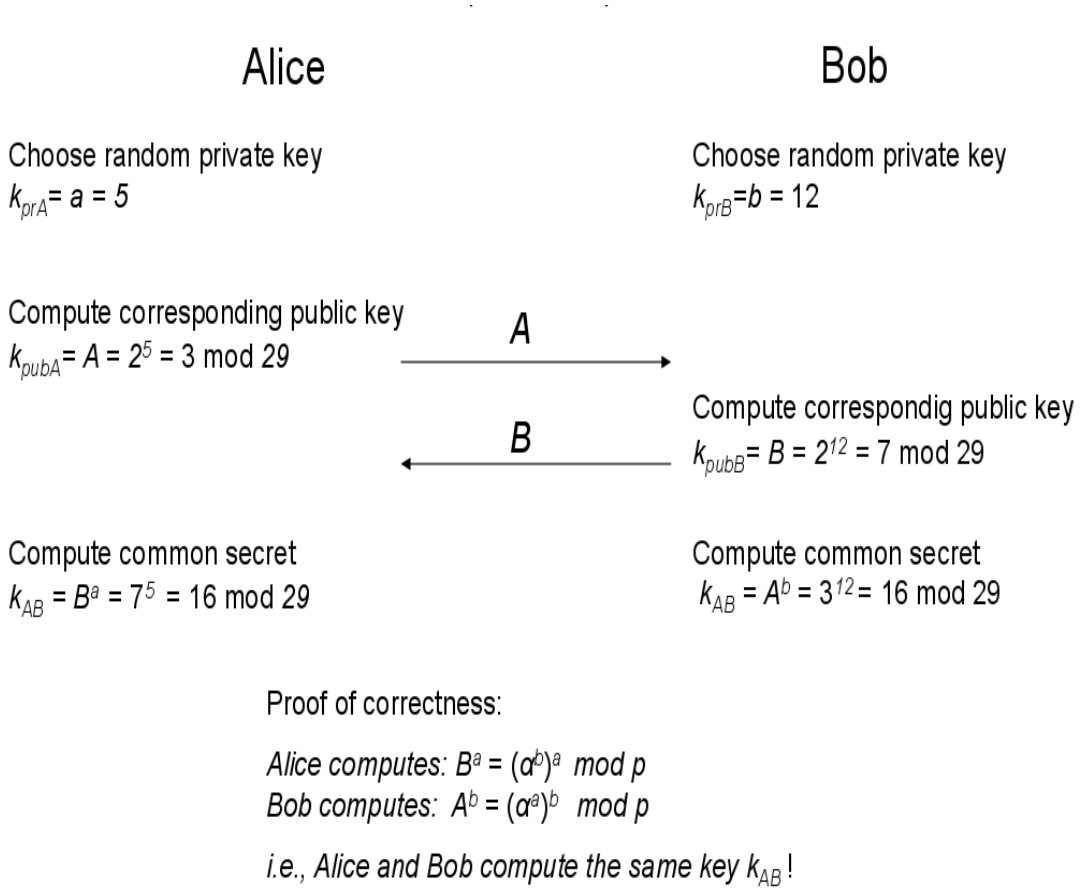

Figure 6: Example of DH

4.3.3 Example3

Let's assume that Alice wants to establish a shared secret with Bob.

1. Alice and Bob agrees on a prime number, p, and a base, g, in advance. For our example, let's assume that p=23 and g=5.
2. Alice chooses a secret integer a whose value is 6 and computes
$$A = g^a \ mod\ p = 8$$
3. Bob chooses a secret integer b whose value is 15 and computes
$$B = g^b \ mod\ p = 19$$
4. Alice sends A to Bob and Bob sends B to alice.

5. To obtain the shared secret, Alice computes

$$s = B^a \, mod \, p = 2$$

6. To obtain the shared secret, Bob computes

$$s = A^b \, mod \, p = 2$$

The algorithm is secure because the values of a and b, which are required to derive s are not transmitted across the wire at all.

5 Issues and Security

5.1 Man-In-The-Middle Attacks

This algorithm , especially in its early forms, has a noteworthy weakness as man - in - the - middle vulnerability. In this attack, a malicious third party, usually referred to as "Eve" (for "eavesdropper") recovers Alice's public key and sends her own public key to Bob. At the point when Bob transmits his public key , Eve interruptson and substitutes the value with her own public key and after that sends it to Alice. At this point, Alice would have go to a agreement on a common secret key with Eve rather than Bob. This exchange should be possible in opposite, and it is feasible for Eve to decrypt any messages conveyed by Alice or Bob, and after that read and perhaps alter them before the re - encryption with the suitable key and transmitting them to alternate party.This weakness is available on the grounds that Diffie-Hellman key exchange does not authenticate the members. Possible solutions include the use of digital signatures and other protocol variants.

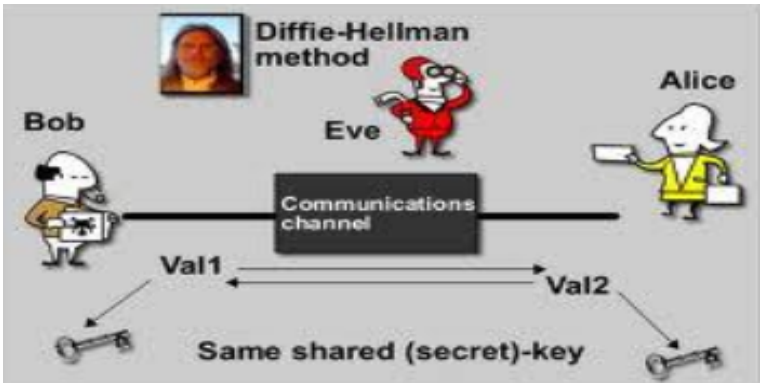

Figure 7: Man-In-The-Middle Attack

5.2 Security Against Attacks

The basic Diffie-Hellman protocol we have shown is not secure against a man-in-the-middle attack. In fact, impossible to achieve security against such an attacker unless some information is shared in advance E.g., private-key setting Or public-key setting.

To address this issue, generally a process of authentication will be expected to guarantee that, at whatever point Alice wishes to send a message to Bob, the beneficiary must be Bob and not an Eve, and the other way around. It is also important - and generally the norm - to discard the keys after use, so that there will be no long - term keys that can be revealed to bring about issues later on. Different concerns ordinarily rotate around upgrading the mathematics involved. That is, to properly generate the randomly choose values with the goal that they are (1) large enough to achieve computational infeasibility for attackers, and (2) random enough, as pseudo-random numbers can greatly ease Eve due to their eventual predictability.

"Generally talking, the fundamental thought is as per the following. Preceding execution of the protocol, the two gatherings Alice and Bob each acquire an public/private key pair and a certificate for the public key. During the protocol, Alice computes a signature on certain messages, covering the public value

$$g^a \, mod \, p$$

Bob proceeds in a similar way. Despite the fact that Eve is still ready to intercept messages in the middle of Alice and Bob, she can't forge signatures without Alice's private key and Bob's private key. Henceforth, the upgraded enhanced protocol defeats the man-in-the-middle attack."

6 Advantages and Disadvantages

Its advantages are

- The security factors with respect to the fact that solving the discrete logarithm is very challenging, and
- That the shared key (i.e. the secret) is never itself transmitted over the channel.

The algorithm has its share of drawbacks including

- The fact that there are expensive exponential operations involved, and the algorithm cannot be used to encrypt messages - it can be used for establishing a secret key only.

- There is also a lack of authentication.
- There is no identity of the parties involved in the exchange.
- It is easily susceptible to man-in-the-middle attacks. A third party C, can exchange keys with both A and B, and can listen to the communication between A and B.
- The algorithm is computationally intensive. Each multiplication varies as the square of n, which must be very large. The number of multiplications required by the exponentiation increases with increasing values of the exponent, x or y in this case.
- The computational nature of the algorithm could be used in a denial-of-service attack very easily.

7 Psuedo Code and Output Screens

7.1 Psuedo Code

```
//read prime no
    String prime=request.getParameter("t1").trim();
    //alice secret value
    String alice=request.getParameter("t2").trim();
    //bob secret value
    String bob=request.getParameter("t3").trim();
    //generating primitive root for given prime no
    String primitive_root = PrimitiveRoot.getG(Integer.parseInt(prime));
    //convert all primitive root in array
    String values[ ] = primitive_root.split(",");
    //prime no
    BigInteger prime_no = new BigInteger(prime);
    //choosing first value from primitive root array as gbase
    BigInteger gbase = new BigInteger(values[0]);
    //Alice, compute A = gbase.pow(alice_pri_key).mod(p) and send to bob
    BigInteger A = gbase.pow(Integer.parseInt(alice)).mod(prime_no);
    //Bob, compute B = gbase.pow(bob_pri_key).mod(p) and send to alice
    BigInteger B = gbase.pow(Integer.parseInt(bob)).mod(prime_no);
    //now alice compute B.pow(alice_pri_key).mod(p) to generate common
secret key
    BigInteger alice_compute = B.pow(Integer.parseInt(alice)).mod(prime_no);
```

//now bob compute A.pow(bob_pri_key).mod(p) to generate common secret key

BigInteger bob_compute = A.pow(Integer.parseInt(bob)).mod(prime_no);

//result printing

StringBuilder sb = new StringBuilder();

sb.append("Prime No : "+prime_no);

sb.append("Primitive Root Modulo Of "+prime_no+" : "+primitive_root);

sb.append("Choosen Primitive Root (g) : "+gbase);

sb.append("Alice Send Bob : "+A);

sb.append("Bob Send Alice : "+B);

sb.append("Alice Compute Secret Key : "+alice_compute);

sb.append("Bob Compute Secret Key : "+bob_compute);

7.2 Output Screens

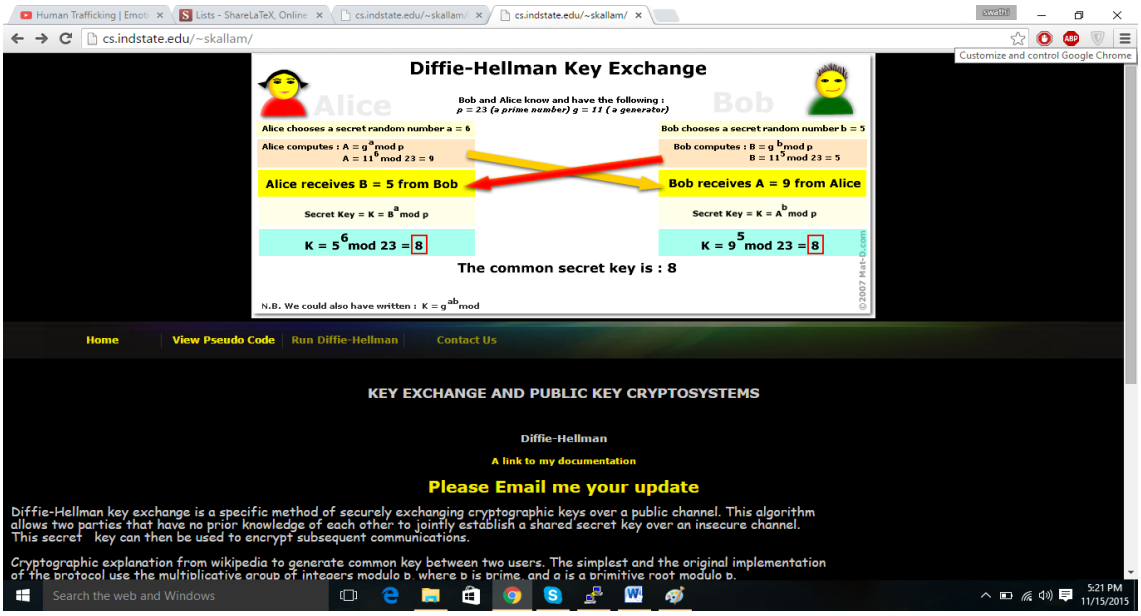

Figure 8: Home Page

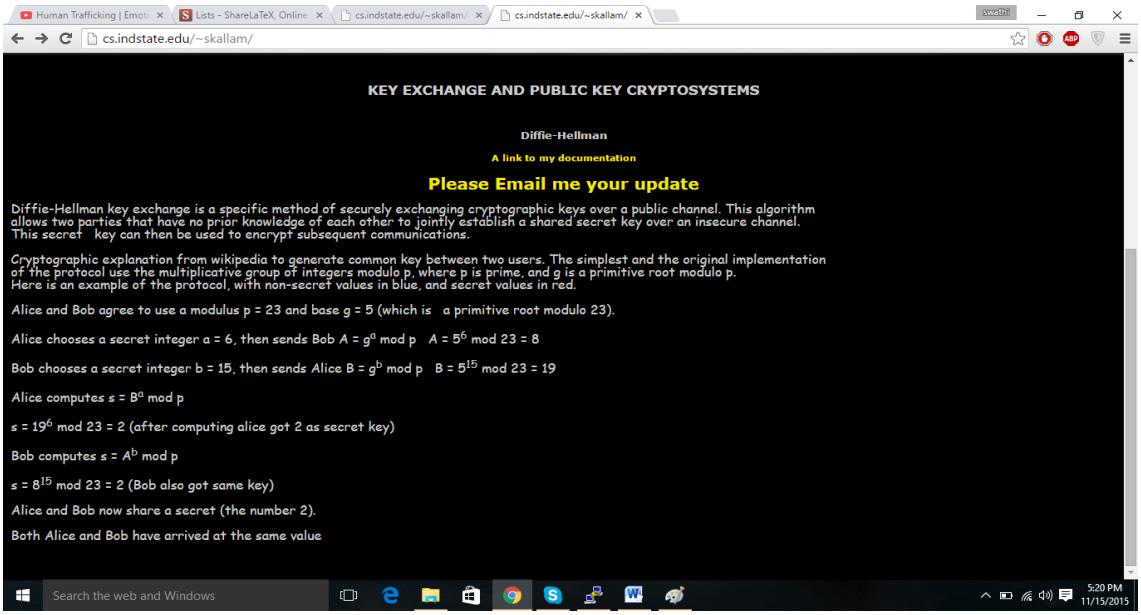

Figure 9: Home Page

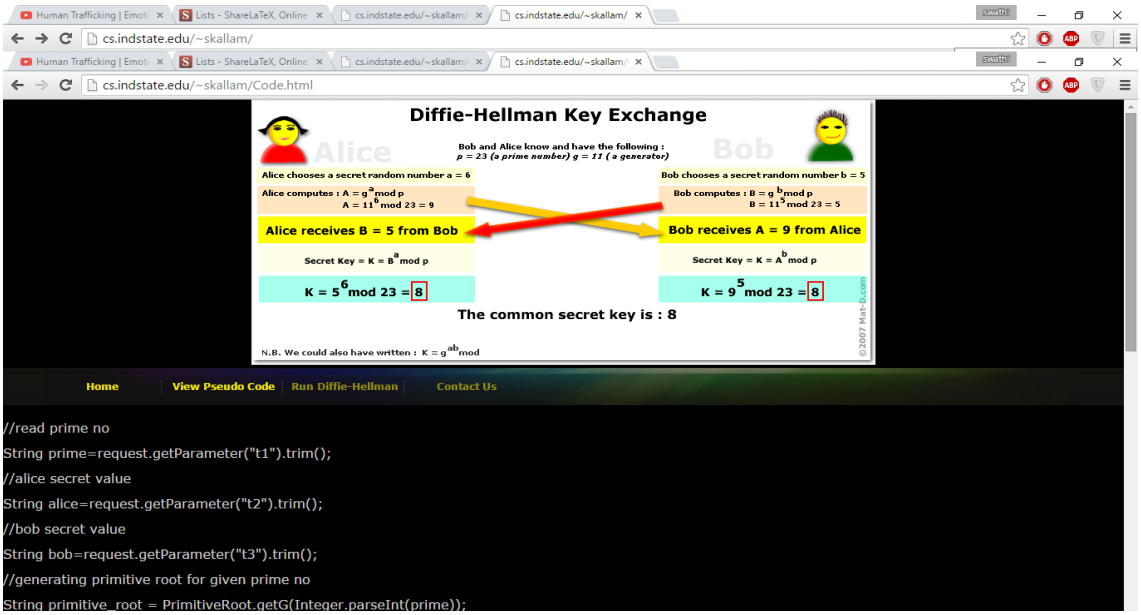

Figure 10: Psuedo Code

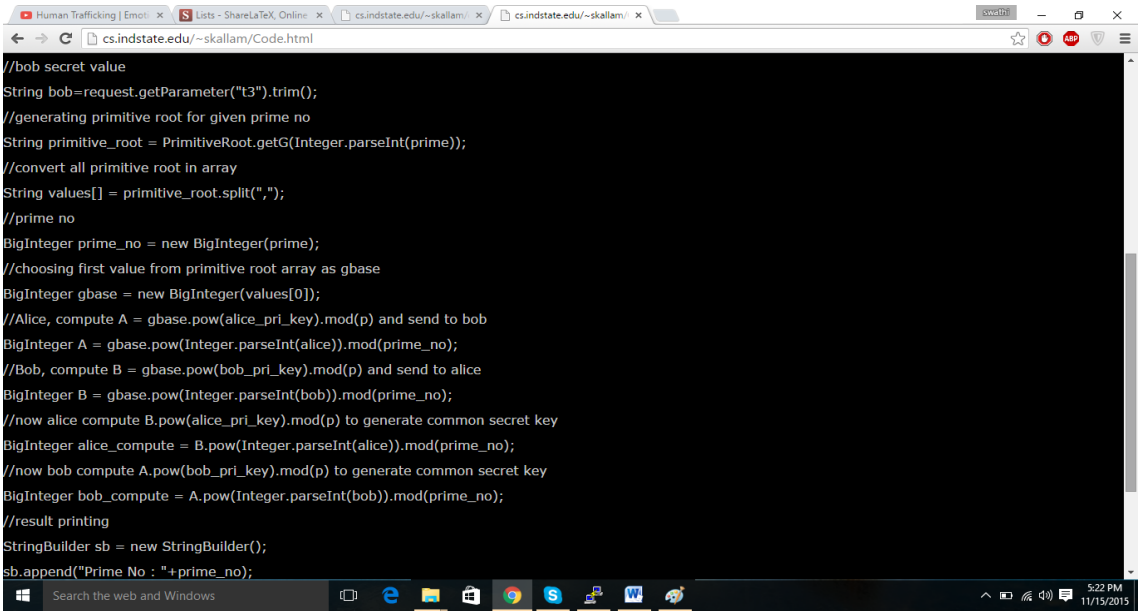

Figure 11: Psuedo Code

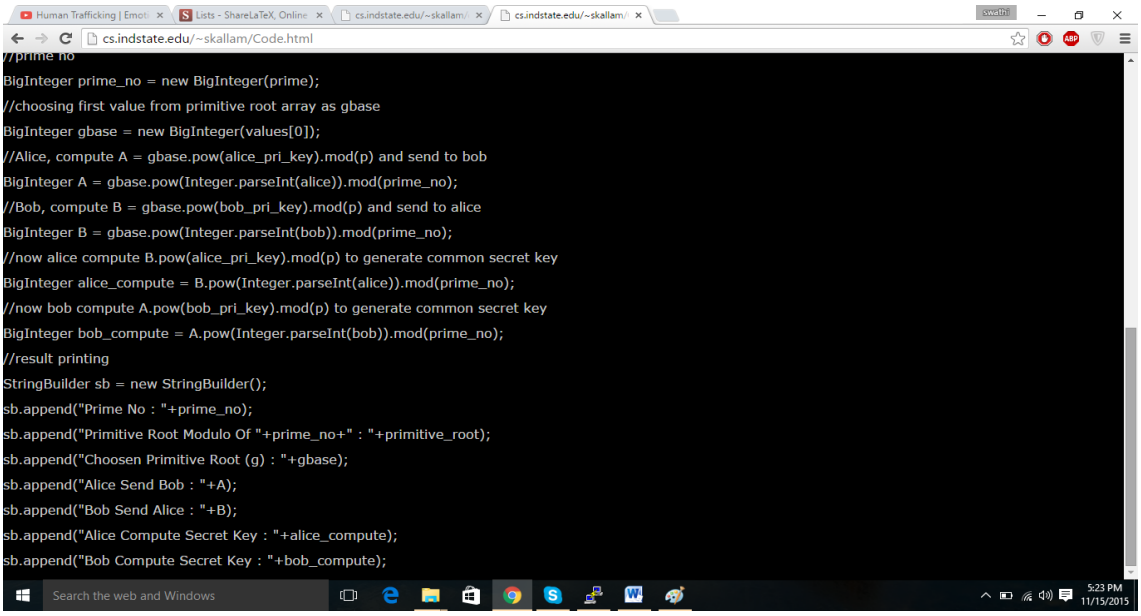

Figure 12: Psuedo Code

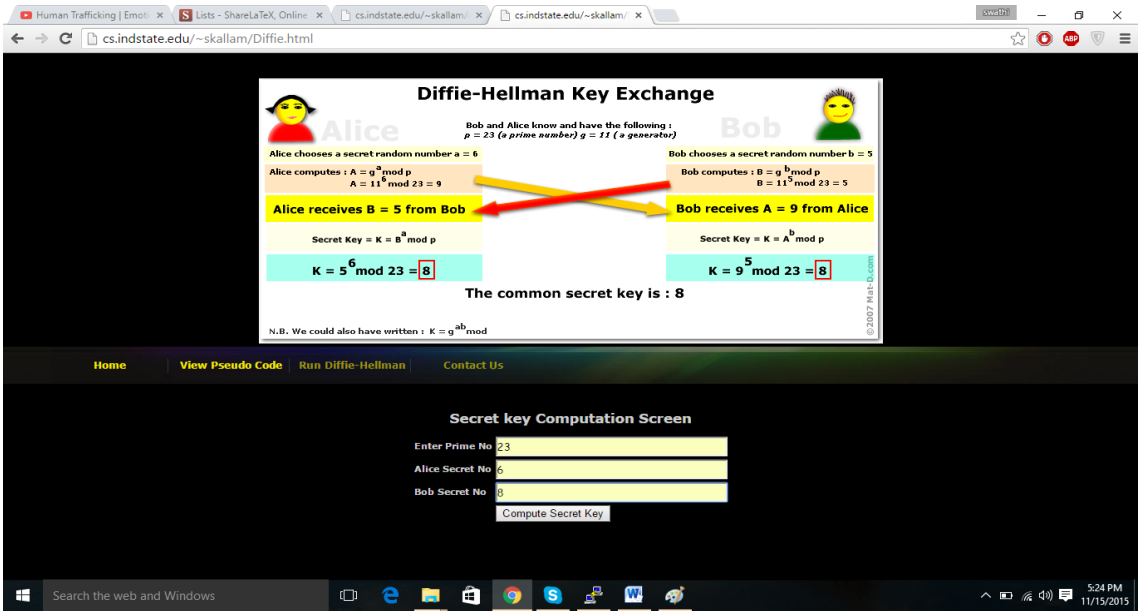

Figure 13: Execution Page

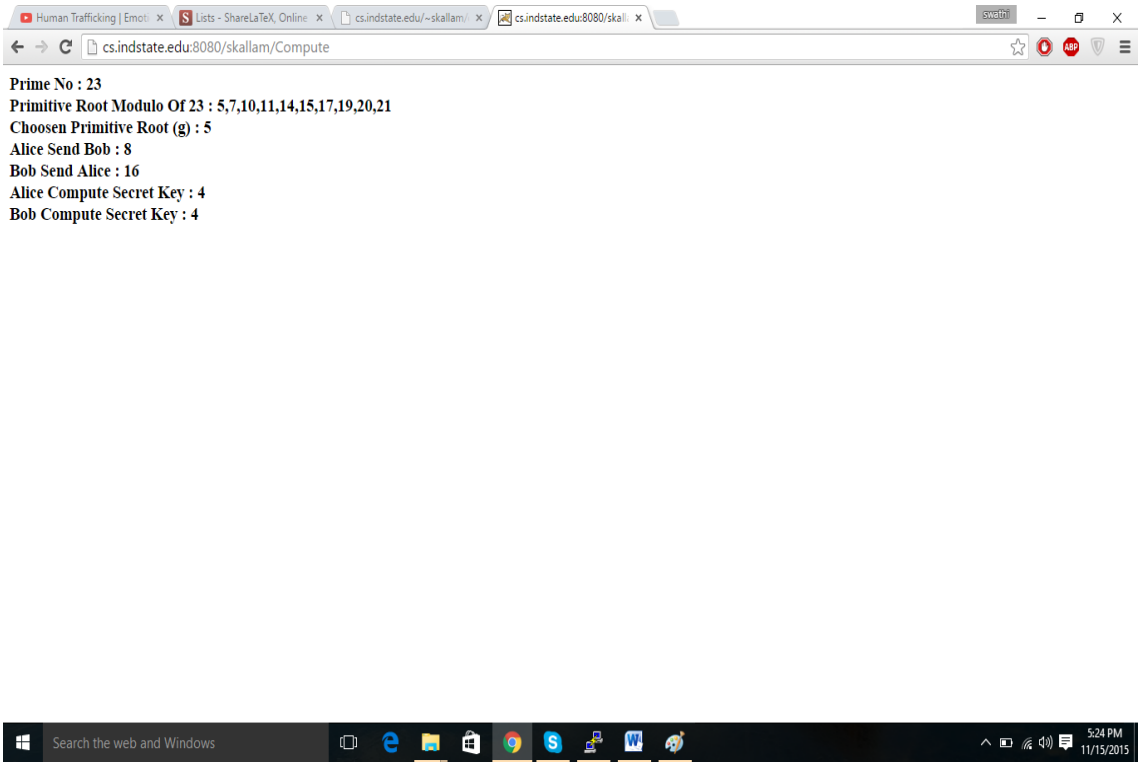

Figure 14: Output Screen

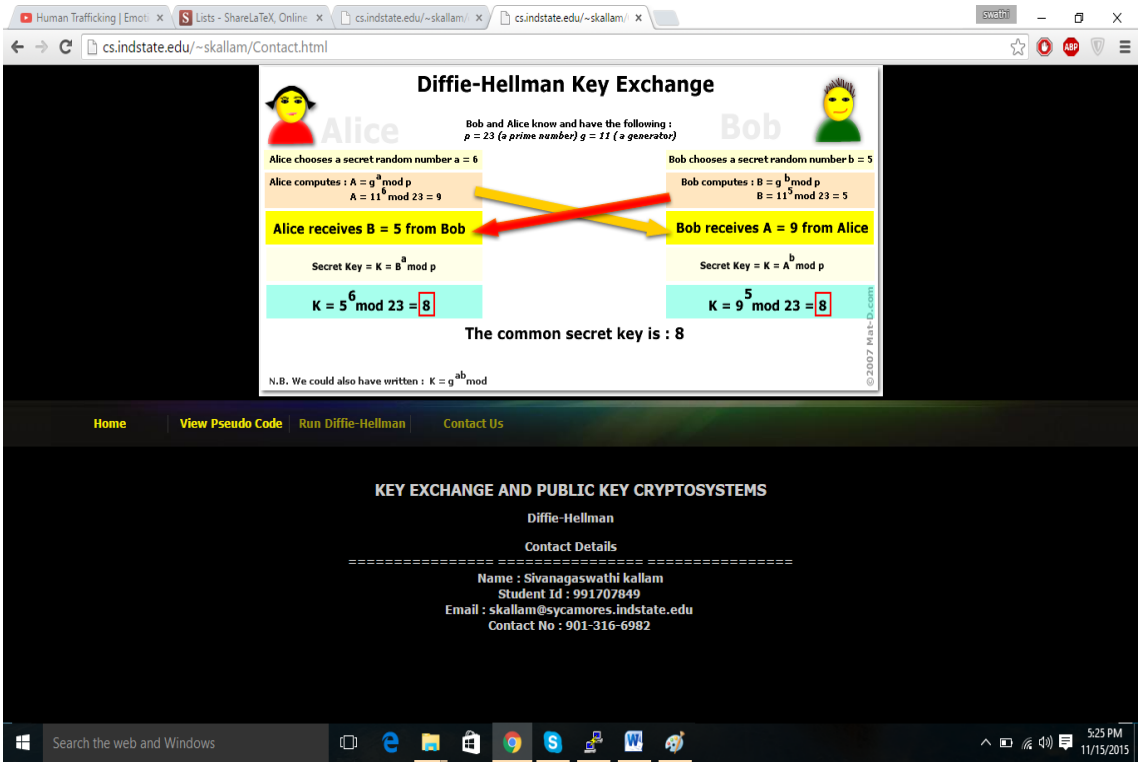

Figure 15: Contact Us Page

8 Future of DH

In spite of the fact that Diffie-Hellman is an public key algorithm, specialists say it don't scale well for future.As of right now it is expressed that Diffie-Hellman keys shorter than 900 bits are not sufficiently secure. To make Diffie-Hellman keys, which now can go to 1,024 bits, secure for the following 10 to 20 years, associations would need to grow to key lengths of no less than 2,048 bits, as per Stephen Kent, chief researcher at BBN Technologies.In the long run, key sizes would need to grow to 4,096 bits. Researchers from the NIST's security technology group expect, that it is exceptionally conceivable, that Diffie-Hellman will be broken inside of 10 years or somewhere in the vicinity.

The cryptographic security standards utilized as a part of public-key infrastructures, RSA and Diffie-Hellman, were presented in the 1970s. And although they haven't been broken, their time could be running out. That is one reason the National Security Agency needs to move to elliptic-bend cryptography (ECC) for cybersecurity. ECC, a complex mathematical algorithm used to secure information in transit, may replace Diffie-Hellman in light of the fact that it can give much more prominent security at a littler

key size. ECC takes less computational time and can be utilized to secure data on smaller machines, including mobile phones, smart cards and wireless devices.

9 Conclusion

Designing a Key exchange algorithm with 100% Accuracy is not possible. Our Algorithm utilizes basic scientific ideas making execution simpler and in addition avoidance from common Attacks.Security change is useful in light of the fact that Diffie Hellman Algorithm is the premise of a few security standards and services on the internet, and if the security of the Diffie Hellman algorithm is compromised, such frameworks will collapse. Diffie Hellman key trade approach for key distribution gives off an impression of being one of the favored systems utilized as a part of practice today.

The Diffie-Hellman key exchange algorithm has turned out to be a standout amongst the most fascinating key distribution schemes being used today. Nonetheless, one must know about the way that in spite of the algorithm is safe against passive eavesdropping, it is not necessarily protected from active attacks. Diffie-Hellman algorithm should be complemented with an authentication mechanism.This methodology for key distribution gives off an impression of being one of the favored routines utilized as a part of practice today.

References

[1] Wikipedia, (2009, October 23). Diffie-Hellman problem
`http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_problem`

[2] Stewards B. Living Internet: Public Key Cryptography, (PKC) History
`http://www.livinginternet.com/i/is_crypt_pkc_inv.htm#diffie`

[3] Wikipedia (2009, November 12). Public-Key Cryptography
`http://en.wikipedia.org/wiki/Public-key_cryptography`

[4] Hickey, K. Government computer news, (2007, Aug 03). Encrypting the future
`https://gcn.com/Articles/2007/08/03/Encrypting-the-future.aspx?Page=1`

[5] Wikipedia, (2009, November 15). Diffie-Hellman key exchange
http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

[6] Wikipedia, (2009, November 12). Alice and Bob
http://en.wikipedia.org/wiki/Alice_and_Bob

[7] Keith Palmgren, CISSP (2006, August).Diffie-Hellman Key Exchange - A Non-Mathematician's Explanation
http://academic.regis.edu/cias/ia/palmgren_-_diffie-hellman_key_exchange.pdf

[8] Hellman M. (2002, May). An Overview of Public Key Cryptology
http://www-ee.stanford.edu/~hellman/publications/31.pdf