International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 01 | Jan 2022 www.irjet.net p-ISSN: 2395-0072

Symmetric Key Encryption Decryption Technique Using Image Based Key Generation

Ms. Prateeksha Arya[1], Mr. Anurag Jain[2], Mr. Rajneesh Pachouri[3]

[1]*M.Tech Research Scholar Department of Computer Science Engineering AIST, Sagar*
[2]*Assistant Professor, Department of Computer Science Engineering AIST, Sagar*
[3]*Assistant Professor, Department of Computer Science Engineering AIST, Sagar*

Abstract - — *Security plays a key role in protecting information from unauthorized or accidental access, although information is in negotiation and where the information is stored. To minimize the damage caused by various attacks, banks, governments, and other security agencies use passwords simultaneously, after which the password will be automatically destroyed. Over the years some have developed many authentication key exchange protocols such as DH-PAKE, J-PAKE etc, but they all suffer from some serious security issues. We provide a legal way to overcome these various issues in the key exchange agreement. We use one-time private key (OTPK) in the context of password authentication key exchange (PAKE), which allows for mutual verification, session key agreement, and opposition to various critical attacks. And enhance more security in our protocol we have used strong session keys which is generated by imaged based key generation process (True random number generation method).*

Key Words: — *Symmetric, TTP, RSA, ECC, DOS, Cryptography.*

1. INTRODUCTION

A Certified Key Exchange Key (PAKE) enables two communication elements to authenticate and set meeting key using unusually effective passwords. The main PAKE meeting was introduced by Bellovin and Merritt in 1992 [21] known as the Encrypted Key Exchange (EKE). Two-party privacy policy based on key trade guarantee (two-PAKE) is equally useful for customer service structures. In any case, in large cases of customer writing where the client needs to speak with many different clients, the Two-PAKE meeting is a major challenge in management that is important for the number of confidential names the customer should keep. Gong, Lomas, Needham, and Saltzer [22] proposed a secret three-party clause based on a key exchange meeting using the employee's public key. Subsequently, Steiner, Tsudik and Waider [23] proposed a PAKE (three-PAKE) team meeting between two clients without the public service key. From a later date, Lee et al. [24] proposed a three-functional PAKE meeting. Wang and Mo [25] showed that Lee et al. The treaty cannot protect itself from pantomime attacks. Wang and Mo also proposed another way to deal with the attack.

Security on PCs is a protection against unauthorized or unplanned disclosure while data is being transferred and you remember that data is not available. Verification agreements provide two things to ensure that your partner is the one you are trying to talk to through an insecure network. These agreements can be considered in three dimensions: quality, efficiency and security.

One time private key

Although there are a variety of techniques that are needed to convey the comfort of the data from sender to receiver. all with data transmission from sender to receiver security plays an significant role because the chances of an attack within the network are high. so to overcome these limitations there are security strategies used for conveying facts in a comfortable way. Authentication is also one of the ways in which facts can be safely sent. One such concept is to provide solid security using key generating using a one-time secret key. As we know the key is an important part of data verification when the sender and receiver use his or her authentication key, but if those keys cannot be validated then such strategies are not secure.

In the idea of a generating a key OTPK is used during the production of the key by the sender or receiver or by any third party the key is generated to ensure either data encryption or coding using the key and as soon as the sender and receiver are authorized and securely sent the data is destroyed. Literature Survey Since the last few years, so many PAKE (password authentication key exchange) protocol have been proposed.

2. Literature Survey

Since the last few years, so many PAKE (password authentication key exchange) protocol have been proposed. In previous operations, the PAKE protocol has been used in a variety of security solutions. Some of these are discussed below:

The Optimistic fair exchange protocol uses a trusted third party, but in a very limited way: a third party is required only in cases where one participant attempts to cheat or rapidly crashes; therefore, for most transactions, a third party will not require to be involved at all. Compared to a protocol used by an external online company, the optimistic approach greatly reduces the burden on the external company, which in turn reduces the cost and security involved in duplicating the service in order to maintain availability. A valid protocol allows two parties to exchange digital signatures online in an appropriate manner, so that each party receives the other's signature, or no one receives it. The most widely used digital signatures are exchanges between the two organizations are the signatures of RSA, DSS, Schnorr, Fiat-Shamir, GQ, and Ong-Schnorr. Some protocols hoping for a good exchange can

easily leave one player hanging for a long time, without knowing if the exchange will end, and without doing anything about it. This can not only be a major disruption, it can also lead to real losses in the case of time-sensitive data such as stock quotes. In fact, fair exchanges include the following separate but related issues: contract signing agreements, guaranteed email systems, non-disclosure agreements [7], [12] and e-mail payment schemes in electronic trading. Agreements for all these issues have their pros and cons.

A. Verifiable Escrows Based Protocol [2]

Verrifiable Escrow Based Protocol [2] A valid escrows-based protocol is a valid protocol that allows two players to exchange digital signatures so that each player receives the other's signature, or no player receives it. These rules of law [2] ensure timely termination of fair trade. A trusted foreign party is simply needed in cases where one actor crashes or tries to cheat. Here a trusted third function is used as an "escrow service". The basic premise is that Alice, the founder, secretly writes her signature under the public key of a trusted foreign company. So Bob, the respondent, can have it removed to encrypt a trusted third party company. In conjunction with this escrow program the standard "cut and select" proof is used which makes it valid. In the sense that the actor getting the escrow can verify that the escrow mark of the preferred form has the correct form attached. This protocol uses three sub-protocols: launcher withdrawal protocol, recipient resolution protocol, and launcher resolution protocol. The protocol may also be used to encrypt data to maintain the integrity of the data while it is being exchanged online.

1) Eligibility: Since TTP is offline its interference to the protocol can be minimized. In using a trusted third party players should not compromise their privacy. The procedure can agree to any standard mark scheme such as RSA, DSS, Schnorr, Fiat-Shamir, GQ, and Ong-Schnorr, .etc signatures. without modification.

2) Incorrect: Encryption creator has the ability to control instances where encryption can be removed by encryption by TTP. Calculation and communication costs are often expensive. In particular, the system does not work well, as expensive cutting and selection methods [20] are used to prove the validity of the encrypted signature.

B. Park et al.'s RSA-Based Multi signature Protocol[15]

Another drawback is that it requires the participating members to execute considerable amounts of computations during the interactive zero-knowledge proof. B. Park et al.'s RSA-Based Multi signature Protocol[15] For e-commerce applications the fair exchange must be assured. In this protocol [15] a method of constructing an efficient fair-exchange protocol by distributing the computation of RSA signatures is described.

By using the features of the multi-signature model, the protocol was developed without the need for proof that he was not in the exchange protocol, so that the calculation was reduced. Only in the protocol setting phase, the use of zero-proof information is required. In this way fairness is ensured by dividing the private key of the RSA into two parts. The signer holds both parts while TTP holds only one part.

1) Eligibility: This program uses a number of signatures that are compatible with the standard basic signature system, making it easy to integrate the exchange feature evenly with existing e-commerce programs. Evidence of ignorance is not used in the exchange protocol, of this method which greatly enhances efficiency.

2) Improper: This legal process is not secure, because a reliable but curious TTP can easily retrieve a user's private key after the end of his or her registration. Dodis and Reyzin [19] violated these rules by pointing to this problem. If this protocol is not implemented successfully either of the two groups can demonstrate the suitability of the intermediate outcomes to the outsider.

C. A Verifier-based Password-Authenticated Key Exchange Protocol via Elliptic Curves [26]

This is an important prerequisite for protection against contract signing, in which case the obligations that form part of the contract may be of benefit to the unscrupulous person or outsider. C. Key Password-Protected Exchange Protocol Based on Elliptic Curves [26] Advanced in the cryptography curve of the elliptic curve, Li et al. [27] and Yoon et al. [28] has proposed two password-authorized key exchange agreements without a public server key. They claim to be safe from multiple potential attacks, securely review user passwords without a complex process, and provide transparent key authentication in the form of a session key agreement. Unfortunately, Li et al.'s [27] protocol is at risk of offline dictionary attacks and malware attacks. At that time the law of Yoon et al. is under attack from an offline dictionary and fails to provide retrospective privacy.

In this paper, Junhan YANG, Tianjie CAO conducts a detailed analysis of errors and suggests a key exchange-based protocol with elliptic curves protected from various known attacks. In this paper, Junhan YANG, Tianjie CAO indicated that Li et al's adherence to the law. [27] is at risk of being attacked by offline dictionary and personal attacks. Additionally, we have shown that the law of Yoon et al. [28] is still at risk of being attacked by an offline dictionary and is failing to provide backwards. We also developed a key-based key exchange protocol using elliptic curves, which are protected from offline dictionary attacks and server interference attacks. In addition, the proposed protocol may also provide uniform authentication, prior confidentiality and retrospective secrecy.

D. Improved Protocol for Essential Exchange of Three Groups of Mobile Trading Areas [29].

Recently, Yang et al. [26] proposed a 3PAKE encryption key protocol based on Elliptic curve cryptography. Their 3PAKE protocol works well because it requires minimal computer costs and minimal communication costs, which are well suited for mobile trading platforms. However, the 3PAKE protocol by Yang et al. is at risk of similar attacks and attacks of impersonation. Zuowen Tan introduced a tool to solve such security issues. A detailed analysis shows that our proposed protocol is a secure 3PAKE protocol and is more efficient.

In this paper, we have shown that the law of Yang et al. [26] is at risk of malignant and similar attacks. We propose an advanced three-party key exchange protocol based on elliptic

curve discrete logarithm. We are introducing a timestamp to keep track of session verification session up to date. The improved system eliminates the weaknesses of the protocol of Yang et al. The analysis shows that the proposed protocol is protected from CDHP and ECDLP thinking. In addition, the improved protocol works much better than the protocol of Yang et al.

E. Cryptanalysis of Other Client-to-Client Key Exchange Protocol Protocols [30].

Protocols allow two clients to create the same session key based on their passwords. In a secure C2C-PAKE protocol, no computer-generated enemy reads anything. About session keys shared between two clients. Especially the participating server should not learn anything.

about session keys. Server risk reassurance is another desirable protection feature of the C2C-PAKE protocol. It means that risking any client A password must not allow an external enemy to be able to share a session key with A. Recently, Kwon and Lee [31] proposed four C2C-PAKE agreements in a three-company arrangement, along with Zhu et al. proposed [32] the C2C-PAKE protocol in the cross-realm setting. All proposed protocols are said to be resistant to server compromises. However, in this paper, we show that the agreements of Kwon and Lee [31] and the protocol of Zhu et al [32] have server compromise attacks, and that a malicious server can attack a person in the middle and can listen to the connection between the two Clients.

3. Proposed Work

The problem with typing public keys is that if the file size is too long it takes longer to encrypt or decrypt [18]. So we used cryptography for compatible keywords to send large files to unprotected networks. And the problem with compatible cryptography is how to share a regular session key, because sharing a regular session key triggers a variety of attacks. Therefore, in order to minimize these types of session-sharing keyword problems we have developed a more efficient protocol. In this protocol, we have suggested that if users or two agencies want to share their key for the same session and establish a 2-factor authentication session, for that purpose they need to register with a trusted third party. In the first group login with a password on TTP, TTP verifies that password when the password is activated and 1 authentication is done in another way to retrieve the message with the wrong password. The first group selects another group (from a list of already registered members on the server) of the contact and sends its identity (User domain name) and a trusted third party ID. TTP is the same as the ID from the website and if it works and sends this ID to the second party using another media (email or mobile), then wait for a second response. The second person identifies his or her identity and, if he or she is interested, then enters the login process with his or her password. TTP verifies that password if the password is valid and 1-factor authentication is verified. Following this verification process, TTP generates a unique random number by a random liar and is transmitted to both parties by other registered media (email or mobile).

The protocol for signing the proposed contract here operates in three stages.

Stage 1: Registration

In the process of registering TTP (a trusted third party company) creates a registration form for the user, the user completes all the required information and sends it to TTP. TTP verifies all useful information and stores it on its website. The user generates the password as instructed by a trusted third party and sends it to TTP, after which TTP stores this password on its website to authenticate the user at the time of signing.

1. User name
2. Mobile number
3. Email address
4. Address etc.

Stage 2: Signing

Here is an example of some of the important details required in the registration form such as:

Category, registered users enter a trusted third name and password. Here at this stage each user needs to generate digital signatures to validate. A trusted third party verifies the password if the password is valid and goes to the key generating process otherwise return the invalid user message.

Stage 3: Key generation

The third phase of our protocol is an important generation, in this phase the third person is responsible for generating a double random number (keys), the first key is generated by a randomly generated production process for verification and after verifying that the second key is generated by a random generator process (primary generation based on image below paragraph 4.1) .As shown in Figure 1 the construction of the proposed project. Here we use the concept of the PAKE protocol using OTPK. The first is the development and signing of both parties requesting TTP digital signatures. If the signatures of both parties have signed an exchange of signatures using TTP and if the signature is the same as the contract there is a transaction between the two parties. In fairness between the two groups TTP is used and the best security is using multiple TTPs.

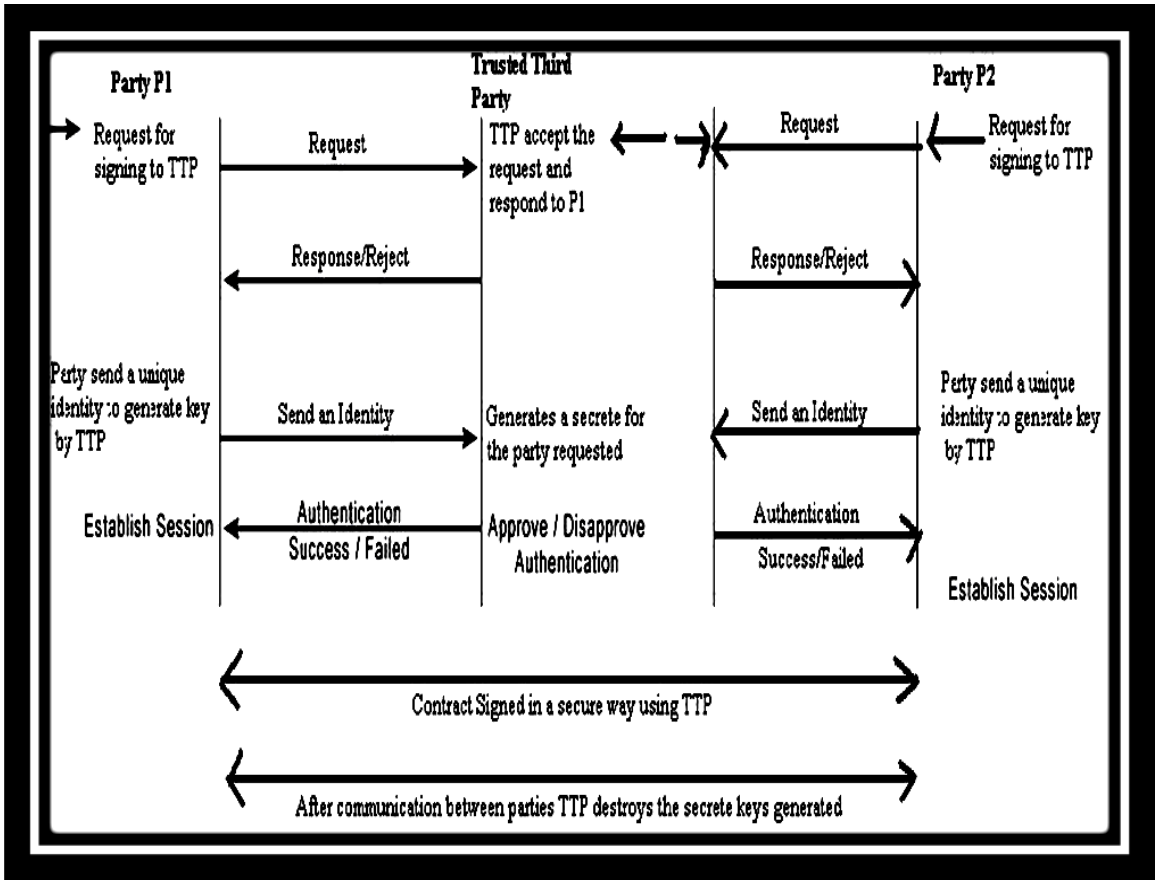


Figure 1 Architecture working of contract signing using OTPK

Image based key generation

Here give the process of key production using the image. Figure 2 shows an example image and figure 3 shows the corresponding binary value of the image. The corresponding Mkey value hash key generated by the image.

We suggest a new way to generate random real numbers based on an image that generates a key of 256 bits or more in the key exchange algorithm. Real random numbers are always secure and good, compared to random random numbers.

This method of operation is very simple, expensive and convenient for the transfer of shared session keys. Our proposed method rapidly enhances the security of exchange key protocols in an insecure channel and may be used for public cryptosystem keys.

Pseudo-Random Number Generators (PRNGs)

As the word 'pseudo' suggests, fake numbers do not happen the way you would expect, at least not if you are used to selling trucks or lottery tickets. In fact, PRNG algorithms use mathematical formulas or predefined tables to produce seemingly random numbers. A good example of PRNG is a linear approach. A lot of research has gone into the theory of counterfeit numbers, and modern algorithms for producing false numbers are so good that the numbers look exactly like they did.

True Random Number Generators (TRNGs)

Compared to PRNGs, TRNGs extract randomly into virtual environments and present them to a computer. You may think of it as death on a computer, but often people are using a physical phenomenon that is easier to connect to a computer than it is. The visual event can be very simple, such as a small variation of another person's mouse movement or the amount of time between keystrokes. In practice, however, you have to be careful about which source you choose. For example, it would be a trick to use keystrokes in this fashion, as the keys are often interrupted by a computer program, which means that a few key keys are collected before being sent to the system waiting for them. In a system waiting for key keys, it will look like the keys are pressed almost simultaneously, and there may not be much random after

Image to Binary format

Here a small image (black and white) can be used to produce TRNGs as well as an image from paint software (eg MS-Paint from Windows). The pixel value of an image can be obtained with the help of simple functions from NET and converted to a unit number of characters. To convert an image to a binary

format, we check the RGB value per pixel. Then we compare those values in pixels.The corresponding values (0s and 1s) are written in the text file from left to right or in any other format. If there is a slight change in the image it leads to a significant difference in the random numbers generated.

The steps to generating a key through image are discussed below:

1. Scan pixel values from top to bottom and left to right.
2. Addition to produce a random number that includes 0 and.
3. We can use any rule to get random numbers like XOR, map, dump etc., we can use the map in this area.
4. Random value can be done by combining columns only or rows only or rows and columns.
5. The same different values can be generated by two groups from the same image for confirmation.

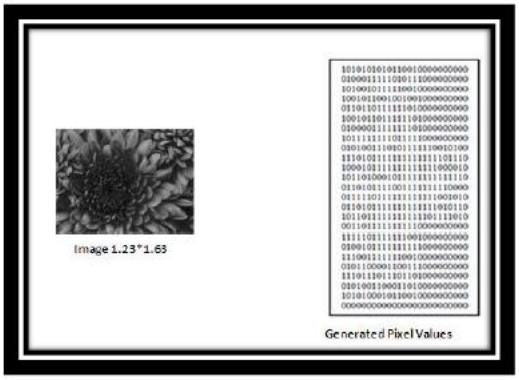


Figure 2: Image based key generation

Generated key is:
00101101001011010010111111000001110000011100011100110011001100110100100010001000100010001001110000000100000001000001101010101010101010101101010111010101110101100010100100101001001010011110

Corresponding Mkey is:
5144c34bb38013d162675d6c090d526b

Proposed Protocol
In table 1 show some notations which is used in our proposed protocol.

Table 1. Notation table for Algorithm

P1	Party 1
P2	Party 2
IDp1	Identity of p1
IDp2	Identity of p2
K	Common shared key of p1&p2
Pw1	Password of party p1
Pw2	Password of party p2
Ek(M)	Encryption of message m using shared key k
Dk(M)	Decryption of message m using shared key k
r	Random number generated by TTP
Si	Secret key generated by TTP using TRNG(True random number generation)
H(r)	One way hash function.
Mkey	Master key which is generated by hash
TTP	Trusted third party

In our proposed protocol we have work on OTPK (one time private key) in the context of password authentication key exchange (PAKE) protocol. Our protocol works on three steps which are discussed below. In step 1 we show the communication between party P1 and trusted third party, in step 2 shows the communication between trusted third party and party P2 and step 3 shows the communication between party P1 and party P2.

Step 1.Communication between party P1 and trusted third party (TTP) or Server

Party P_1 ⟷ TTP

a) User login: Party P_1 login with Pw_1 and required information to the trusted third party, TTP verify the password and if password is valid then print the message user successfully login otherwise print the invalid password message.

b) Send identities: PartyP1 send the identities (IDp1& IDp2) to the trusted third party, and TTP recognized the identities for further communications.

c) Generate random number r (one time password): When the above two steps is successfully done the TTP generate random number 'r1' for the party P1 with the timestamp (5 minute). That random number r1 sends to the party via other media (email).
Note: timestamp (5 minute) means r automatically destroys in 5 minute.

d) Party P_1 perform function:
In this stage party P_1 perform some functions like:

i. Generate master key (M_{key})

$$M_{key} = H(r_1 + Pw_1)$$

ii. Keep this random number in memory.
iii. Send Mkey to the TTP.

e) Verification:
TTP matched the Mkey with own calculated Mkey, if that is valid then server generated imaged based key (this method describe above in section image based key generation) Si and calculated hash H(Si) of this key and send to the party P1.

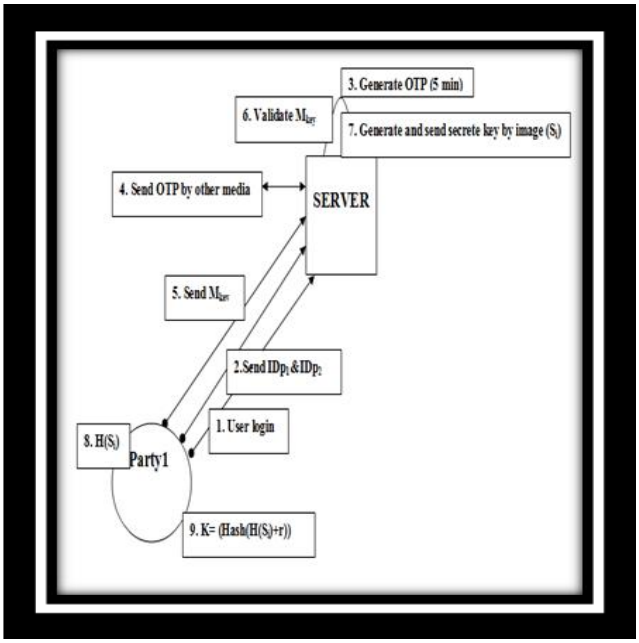
f) Session key(K) generation:

K=Hash(H(Si) + r1)

Party P1 generated his common session key (K) by the concatenate of H(Si) and random number r1.

Step 2. Communication between party P_2 and trusted third party (TTP) or Server

Figure 1working module between party P_1 and server



Step 2. Communication between party P_2 and trusted third party (TTP) or Server

Party P_2 ⟷ TTP

a) User login:

Party P_1 login with Pw_1 and required information to the trusted third party, TTP verify the password and if password is valid then print the message user successfully login otherwise print the invalid password message.

b) Send identities:

PartyP_1 send the identities (IDp_1& IDp_2) to the trusted third party, and TTP recognized the identities for further communications.

c) Generate random number r (one time password):

When the above two steps is successfully done the TTP generate random number 'r_1' for the party P_1 with the timestamp (5 minute). That random number r_1 sends to the party via other media (email).

Note: timestamp (5 minute) means r automatically destroys in 5 minute.

d) Party P_1 perform function:

In this stage party P_1 perform some functions like:

i. Generate master key (M_{key})
$M_{key}= H(r_1+Pw_1)$
ii. Keep this random number in memory.
iii. Send M_{key} to the TTP.

e) Verification:

TTP matched the M_{key} with own calculated M_{key}, if that is valid then server generated imaged based key (this method describe above in section image based key generation) S_i and calculated hash $H(S_i)$ of this key and send to the party P_1.

f) Session key(K) generation:

$K=Hash (H (S_i) + r_1)$

Party P_1 generated his common session key (K) by the concatenate of $H(S_i)$ and random number r_1.

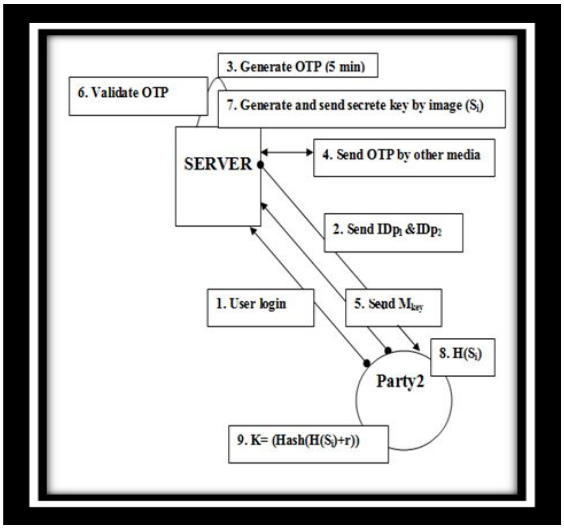


Figure.2 working module between party P_2 and server

Step 3. Communication between party P_1 and party P_2.

Party P_1 ⟷ party P_2

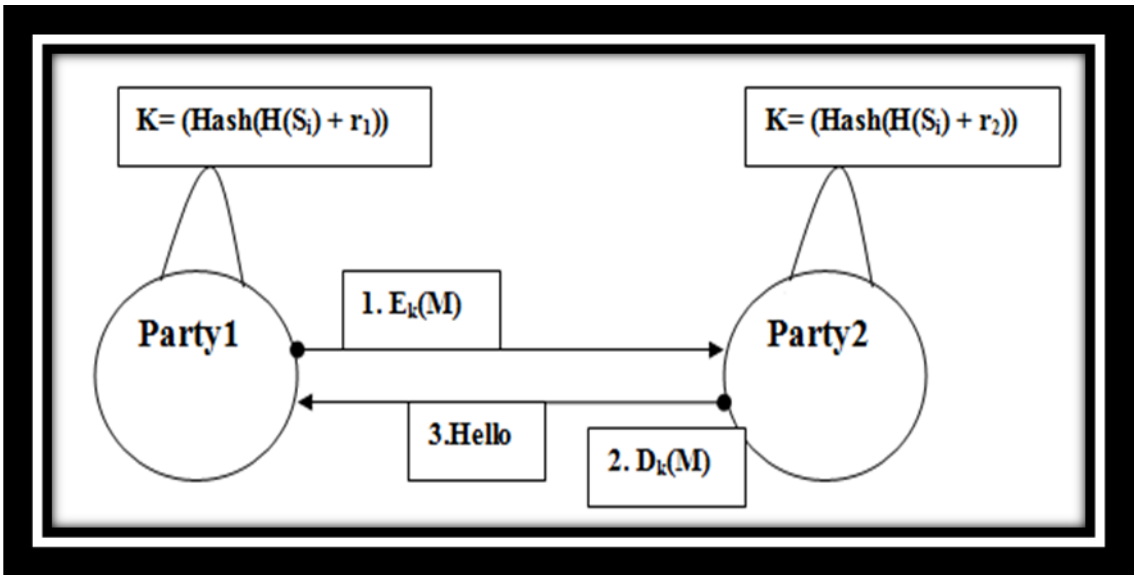


Figure 3 working module between party P_1 and party P_2

a) Encryption:

Party P_1 encrypted the message (M) using the common session key K. If party P_2 successfully decrypted the message (M) then confidentiality as well as integrity proof.

b) Decryption:

Party P_2 decrypted the message (M) using the common session key K. if party P_2 successfully decrypted the message then party P_2 confident that message is coming from party P_1. After that party P_2 send confirmation message "Hello" to the party P_1.

4. Results and Discussion

The table 2 shows the comparison of basic features, security, and efficiency between these referenced and our protocols. In the category of basic features, the properties such as transparent trusted third party or not, off-line or on-line trusted third party are considered. Here two main security requirements are compared: fairness and timeliness. The protocol which guarantees the two parties obtain or not obtain the other's signature simultaneously is fair. This property implies that even a fraud party who tries to cheat cannot get an advantage over the other party.

A trusted third function company involved during each protocol session but not during each transaction, is said to be online. Off-line TTP - A reputable third-party company that only engages in protocols in the action of a business misconduct or in the action of a network error, is said to be offline.

Table 2. Comparison of different Contract Signing Protocols

Parameters	Protocols				Proposed Protocol
	Escrows Based Protocol [19]	Park et. al,'s RSA based protocol [21]	Bao et. al.'s Protocol [22]	Contract Signing Protocol based on RSA [20]	Proposed Protocol
Fairness	YES	YES	YES	YES	YES
Timeliness	YES	YE	YES(weak)	YES	YES
Multiple TTP	YES	YES	YES	YES	YES
Replay attack	YES	YES	YES	YES	NO
Confidentiability	NO	NO	NO	YES	YES
Additional Authentication	NO	NO	NO	NO	YES
Storage Cost	MORE	MORE	MORE	MORE	LESS

In table 2 us show time calculations of the different images. If the size of image is big then it takes more time to

calculate the session key. So server takes some less size image for the generation of key.

In table 3 shown time calculations of the different images. If the size of image is big then it takes more time to calculate the session key. So server takes some less size images for the generation of key. As shown in the table is the size of the key generated using an image of different sizes and their respective time in mills seconds.

Table 3. Time Computation of Image key generation

Images	Image Size	Image key size(bits)	Time in ms
Image i	18KB	7498	209
Image ii	835KB	786435	510
Image iii	512KB	569855	620
Image iv	760KB	634680	438
Image v	778KB	634800	621
Image vi	537KB	567447	580
Image vii	769KB	634678	950
Image vii	619KB	648989	549

5. Conclusion and Future Scope

The importance of network security continues to grow with the rapid growth of computer technology, and these days it is becoming increasingly important in the computer world.

Fifty years ago, cybercrime was never heard of again. New technologies present new challenges, as computer technology advances rapidly; cybercrime situations threaten computer security. Nowadays, cybercrime is on the rise; To deal with the current level of crime, computer security becomes a basic necessity. Active computer security is now more important than ever, and the need to increase awareness is urgent.

Their exciting discovery on this site has led to excellent security standards that have contributed to major developments in the computer world. In order to make further progress, we must accept the fact that security is not stable, and that risks remain, so we do everything we can to minimize those potential and technological risks.

In this article, we are dealing with authentication which is one of the most important aspects of computer security. We then focus on the one-time private key in the context of the password authentication protocol. By using our protocol we have exchanged a regular session key for strong two-factor authentication. The proposed method here does not require much storage does not keep the key or data for a while so the chances of various attacks on the network are reduced such as replay attacks or identity theft attacks. To make the session key more robust we used TRNG (a real random number generator) in the key generation. The Session key generated is a combination of an image generated by the master key and OTP (one-time password), which creates a strong session key, to minimize all chances of attack. Once a session key is established there is no TTP involvement, so the parties do not share their information on TTP. After verification is done OTP will destroy the TTP side. OTPK does not allow any party to use its signatures repeatedly in signing a contract because as soon as TTP confirms the production keys groups will be lost and the parties are required to produce different signatures in exchange for different agreements. The concept of two-factor authentication using an image is as effective as verification is concerned and the types of attacks that are difficult to accomplish can be easily prevented by the process. Various types of attacks like replay attacks, DOS attacks, internal attacks, external attacks; fake password attacks are easily prevented. The concept of two-factor authentication can be used by multiple groups so that when communicating groups can easily share their data securely.

We plan to implement the proposed theoretical solutions in key exchange password verification programs, and look forward to seeing our protocol in practice. Our main idea is very common; we expect it to be integrated into existing password verification protocols and to be used in the future.

REFERENCES

1. G. Wang, “An abuse-free fair contract signing protocol based on the RSA signature,” IEEE Transactions on Information Forensics andSecurity, vol. 5, no. 1, pp. 158–168, Mar 2010.
2. N. Asokan, V. Shoup, and M. Waidner, “Optimistic fair exchange of digital signatures,” IEEE J. Sel. Areas Commun., vol. 18, no. 4,pp. 591–606, Apr. 2000.
3. M. Abadi, N. Glew, B. Horne, and B. Pinkas, “Certified e-mail with a light on-line trusted third party: Design and implementation,” inProc. 2002 Int.World Wide Web Conf. (WWW’02), 2002, pp. 387–395, ACM Press.
4. G. Ateniese, “Efficient verifiable encryption (and fair exchange) of digital signature,” in Proc. ACMConf. Computer andCommunications Security (CCS’99), 1999, pp. 138–146, ACM Press.
5. G. Ateniese and C. Nita-Rotaru, “Stateless-receipient certified e-mail system based on verifiable encryption,” in Proc. CT-RSA’02,2002, vol. 2271, LNCS, pp. 182–199, Springer-Verlag.
6. F. Bao, R. H. Deng, and W. Mao, “Efficient and practical fair exchange protocols with off-line TTP,” in Proc. IEEE Symp. Securityand Privacy, 1998, pp. 77–85.

7. S. Gürgens, C. Rudolph, and H. Vogt, "On the security of fair nonrepudiation protocols," in **Proc.** ISC'03, 2003, vol. 2851, LNCS, pp.193–207, Springer-Verlag.
8. G. Wang, "Generic non-repudiation protocols supporting transparent off-line TTP," J. Comput. Security, vol. 14, no. 5, pp. 441–467,Nov. 2006.
9. S. Micali, "Simple and fast optimistic protocols for fair electronic exchange," in Proc. PODC'03, 2003, pp. 12–19, ACM Press.
10. J. Zhou, R. Deng, and F. Bao. Some remarks on a fair exchange protocol. In: Public Key Cryptography (PKC'00), LNCS 1751, pp. 46-57. Springer-Verlag, 2000.
11. C. Boyd and E. Foo, "Off-line fair payment protocols using convertible signatures," in Proc. ASIACRYPT'98, 1998, vol. 1514, LNCS,pp. 271–285, Springer-Verlag.
12. M.A. Strangio, "An Optimal Round Two-Party Password Authenticated Key Agreement Protocol," Proceeding of the First IEEE International Conference on Availability, Reliability, and Security (ARES'06), pp. 216–223, April. 2006.
13. F. Bao, G. Wang, J. Zhou, and H. Zhu, "Analysis and improvement of Micali's fair contract signing protocol," in Proc. ACISP'04,2004, vol. 3108, LNCS, pp. 176–187, Springer-Verlag.
14. M. Bellare and R. Sandhu, The Security of Practical Two-Party RSA Signature Schemes 2001 [Online]. Available: http://wwwcse.ucsd. edu/users/mihir/papers/
15. J. M. Park, E. Chong, H. J. Siegel, and I. Ray, "Constructing fair exchange protocols for e-commerce via distributed computation ofRSA signatures," in Proc. PODC'03, 2003, pp. 172–181, ACM Press.
16. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proc. 1st ACM Conf.Computer and Communications Security (CCS'93), 1993, pp. 62–73, ACM press.
17. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM,vol. 21, no. 2, pp. 120–126, Feb. 1978.
18. O. Markowitch and S. Kremer. An optimistic non-repudiation protocol with transparent trusted third party. In: Information SecurityConference (ISC'01), LNCS 2200, pp. 363-378. Springer-Verlag, 2001.
19. Y. Dodis and L. Reyzin, "Breaking and repairing optimistic fair exchange from PODC 2003," in Proc. ACM Workshop on DigitalRights Management (DRM'03), 2003, pp. 47–54, ACM Press.
20. A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in Proc. CRYPTO'86,1987, vol. 263, LNCS, pp. 186–194, Springer-Verlag.
21. S. M. Bellovin and M. Merrit, Encrypted key exchange: password-based protocols secure against dictionary attacks, Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 72-84, 1992.
22. L. Gong, M. Lomas, R. Needham and J. Saltzer, Protecting poorly chosen secrets from guessing attacks, IEEE Journal on Selected Areas in Communications, Vol. 11, No. 5, pp. 648-656, 1993.
23. M. Steiner, G. Tsudik and M. Waidner, Refinement and extension of encrypted key exchange, ACM Operating Systems Review, Vol. 29, No. 3, pp. 22-30, 1995.
24. S.-W. Lee, H.-S. Kim and K.-Y. Yoo, Efficient verifier-based key agreement protocol for three parties without server's public key, Applied Mathematics and Computation, 167(2), pp. 996-1003, 2005.
25. R-C Wang and K-R Mo, Security enhancement on efficient verifier-based key agreement protocol for three parties without server's public key, Int. Math. Forum, 1(17-20), pp. 965 – 972, 2006.
26. Junhan YANG, Tianjie CAO "A Verifier-based Password-Authenticated Key Exchange Protocol via Elliptic Curves", Journal of Computational Information Systems 7:2, pp. 548-553, 2011.
27. W.M. Li, and Q.Y. Wen. Efficient verifier-based password-authentication key exchange protocol via elliptic curves. In Proceedings of 2008 International Conference on Computer Science and Software Engineering, pages 1003-1006, 2008.
28. E.J. Yoon, and K.Y. Yoo. Robust User Password Change Scheme based on the Elliptic Curve Cryptosystem. In FundamentaInformaticae, pages 483-492, 2008.
29. Zuowen Tan "An Enhanced Three-Party Authentication Key Exchange Protocol for Mobile Commerce Environments", JOURNAL OF COMMUNICATIONS, VOL. 5, NO. 5, pp. 436-443, MAY 2010.
30. Tianjie Cao, Tao Quan, Bo Zhang "Cryptanalysis of Some Client-to-Client Password-Authenticated Key Exchange Protocols", JOURNAL OF NETWORKS, VOL. 4, NO. 4, pp. 263-270, JUNE 2009.
31. T. Kwon and D. H. Lee, "Three-party password authenticated key agreement resistant to server compromise", WISA 2006, LNCS 4298, pp. 312-323, 2007.
32. H. Zhu, T. Liu, J. Liu and G. Chang, "An efficient client to client password-authenticated key exchange resilient to server compromise," 13th IEEE International Symposium on Pacific Rim Dependable Computing, pp. 405-408, 2007.
33. Yalin Chen, Jue-Sam Chou, Chun-Hui Huang "Cryptanalysis on Four Two-Party Authentication Protocols", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 2, 2010.