Analysis on Secure Key Agreement Protocol Using Trusted Third Party Information Sensitivity Systems

Gnanasekaran P, Umadevi V

***Abstract*— The study of security models for sensitive data systems has been taken on for years. Throughout this century, the thought of seeking the system security to the supply of system development lifecycle received Brobdingnagian improvement within the system and software system assurance domain. This paper expounds the understanding security by illustrating information security study development progress since pre-computer age and presents an outline of Internet and cyberization security by summarizing the established order of cyberization. Then a security model referred to as PDRL, which incorporates six core security attributes of sensitive data systems, is planned to safeguard the protection of sensitive data systems within the whole system life-cycle. Within the past, many key agreement protocols square measure planned on watchword based mostly mechanism. These protocols square measure prone to wordbook attacks. Storing plain text version of watchword on server isn't secure continuously. During this paper we have a tendency to utilize the service of a trustworthy third party, i.e., the Key Distribution server (KDS) for key agreement between the hosts. Now-a-days in massive operating environments 2 party key agreement protocols square measure being seldom used. During this planned theme, rather than storing plain text version of watchword we have a tendency to store a technique hash of the watchword at the server. Each host and server agrees upon family of independent unidirectional hash functions, victimization that host authentication is completed once a bunch applies for session key with KDS. Host establishes just once key with server victimization that server authentication is completed. Thanks to this man-in-the middle attacks square measure defeated. The planned protocol relies on Diffie-Hellman key exchange protocol.**

***Index Terms*— Key Agreement, Diffie-Hellman, Sensitive System, Detector Network, Security Model**

I. INTRODUCTION

Information security is outlined by a group of necessities for storage, process and distribution of data. This set defines a security policy (Bellare Mihir & Rogaway Phillip, 1995). If the protection policy is enforced, then the system of data security may be given within the type of the subsequent international diagram (GDIS, see Fig. 1):

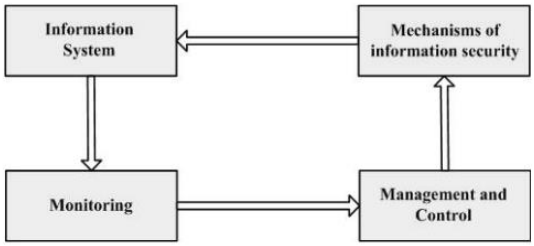

Fig1. Global diagram of information security subsystem.

Gnanasekaran P , Research Scholar, Jairams Arts and Science College, Karur, Affiliated to Bharathidasan University, Trichy,Tamilnadu, India.
Umadevi V, Professor & Research Director, Jairams Arts and Science College, Karur, Affiliated to Bharathidasan University, Trichy,Tamilnadu, India.

The implementation of necessities of a security policy consists of the subsequent steps:

• The separation of objects for the implementation of requirements;
• The identification of objects for the implementation of necessities and also the identification of necessities that ought to be glad for these objects;
• The implementation of necessities of the protection policy via obtainable security mechanisms.

The main goal of cryptography is to alter secure communication during hostile surroundings. 2 parties Pi and Pj, wish to soundly communicate over a network occupied by a full of life mortal. Usually, Pi and Pj can wish to make sure the privacy and believability of the information they send to every different. They're going to write and attest their transmissions. However before Pi and Pj will use these tools they're going to got to have keys. Indeed, while not keys, cryptography merely cannot get off the bottom. Key agreement is one in every of the elemental cryptanalytic primitive once secret writing and digital signature. Such protocols permit 2 or additional parties to exchange info among themselves over associate adversarially controlled insecure network and agree upon a typical session key, which can be used for later secure communication among the parties. Thus, secure key agreement protocols function basic building block for constructing secure, complex, higher-level protocols. Key institution is also generally divided into key transport and key agreement.

Secret communications with secret keys implies that solely sure parties ought to have copies of the key. Though secret keys will assure USA of confidentiality, authentication of users, and message integrity, during an international world we have a tendency to should be able to firmly distribute keys at a distance during a timely manner (Menezes A, Oorschot P. van & Vanstone S, 1996).

If security is to be maintained, key distribution should be as solid because the cryptanalytic methodology and should be able to make sure that solely sure parties have copies of the keys (Schneier Bruce, 1994). Obviously, key distribution may be a vital downside. Key institution protocols involving authentication usually need a set-up section whereby authentic and presumably secret initial keying material is distributed. Most protocols have as associate objective the creation of distinct keys on every protocol execution. In some cases, the initial keying material pre-defines fastened key which can result each time the protocol is dead by a given try or cluster of users. Systems involving such static keys are insecure beneath known-key attacks.

Key pre-distribution schemes are key institution protocols whereby the ensuing established keys are utterly determined a priori by initial keying material. In distinction, dynamic key institution schemes are those whereby the key established by a set try (or group) of users varies on future executions.

Dynamic key institution is additionally noted as session key institution. During this case the session keys ar dynamic, and it's sometimes meant that the protocols are proof against known-key attacks. Several key institution protocols involve a centralized or sure party, for either or each initial system setup and on-line actions (i.e., involving time period participation). This party is noted by a range of names looking on the role compete, including: sure third party, sure server, authentication server, key distribution center (KDC), key translation center (KTC), and certification authority (Stallings Williams, 2004 ; Mel H.X., Baker Doris.M. & Burnett Steve, 2004).

It is typically desired that every party during a key institution protocol be able to verify truth identity of the other(s) that might presumably gain access to the ensuing key, implying prevention of any unauthorized extra parties from deducing an equivalent key. During this case, the technique is alleged (informally) to produce secure key institution. This needs each secret of the key and identification of these parties with access thereto (Bellare Mihir & Rogaway Phillip, 1995).

In a secure system, words may be simply guessed if user chooses their own password in plain text (L. Gong, M. A. Lomas, R. M. Needham & J. H. Saltzer, 1993). Storing plain text version of word on server isn't secure. This weakness exists in much all wide used systems. The projected protocol is secure against wordbook attacks as we have a tendency to use only once keys with server. This protocol is additionally secure against malicious business executive attacks, wherever a number misuses the knowledge in one protocol run to a different. Projected protocol conjointly provides good forward secrecy i.e. even though one secret is disclosed future session keys won't be disclosed. As we have a tendency to don't use any Public Key Infrastructure (PKI), giant process power isn't needed. Since this is often a third-party key agreement protocol each host needn't share secret info with different host.

In this paper in Section two, we have a tendency to review short comings of existing protocols. In section three we have a tendency to discuss our new third-party Key Agreement Protocol. Formal security analysis of projected protocol is finished in Section four. Finally a terminal remark is finished in Section five.

II. RELATED WORK

DH-BPAKE (M. Strangio, 2006) could be a 2 party key agreement protocol supported Diffie-Hellman (1976) and Encrypted key exchange protocols that were planned by Strangio (S. Bellovin & M. Merritt, 1992). This protocol isn't appropriate for big networks wherever we tend to cannot assume each party shares a secret (password) with every alternative party. Straightforward attested Key Agreement (SAKA) protocol planned by Her-Tyan Yeh et al (2002) is additionally a 2 party key agreement protocol that based on parole based authentication and Diffie-Hellman key agreement. User authentication is one in every of the foremost necessary security services in secure communications. It's necessary to verify the identities of the communication parties before they begin a replacement affiliation. Password-based mechanism is that the most generally used methodology for user authentication since it permits folks to decide on and bear in mind their own parole with none assistant device. This protocol is easy and value effective, however is being seldom utilized in massive networks.

STW protocol could be a 3 party Encrypted key exchange protocol planned by Steiner et al (1995). Since this can be a 3 party key agreement protocol, each the hosts share a secret key solely with trusty third party. Peal et al (Y. Ding & P. Horster, 1995) have evidenced that this protocol is at risk of undetectable on-line approximation attacks.

According to sculpture C.L. et al (2000), this protocol is additionally at risk of offline approximation attacks. Associate in nursing aggressor tries to use a guessed parole in internet dealing. Host verifies the correctness of his guess victimization responses from server. If his guess fails he should begin a replacement dealing with server victimization another guessed parole. A failing guess can't be detected and logged by server, as server isn't ready to depart Associate in nursing honest request from a malicious request. In off-line approximation attacks Associate in nursing aggressor guesses a parole and verifies his guess offline. No participation of server is needed; therefore server doesn't notice the attack. If his guess fails, the aggressor tries once more with another parole, till he finds the correct one. Among these categories of attacks, the off-line parole approximation attack is that the most comfy Associate in nursing promising one for an aggressor. It not noticeable and has no communication value. Storing a noticeable text version of the shared parole at the server could be a constraint that can't (or ought not) invariably be met. Specially, think about the matter of a user work in to a laptop that doesn't trust a secure key server for authentication. It's inadvisable for many hosts to store passwords in either plain kind or in an exceedingly reversibly encrypted kind. LSH 3-PEKE protocol was planned by Chun-Li sculpture et al (C. L. Lin, H. M. Sun, & Hwang, 2000). This protocol is securing against each the offline approximation attack and undetectable on-line approximation attacks however conjointly satisfies the protection properties of good forward secrecy.

The most necessary demand to forestall undetectable on-line approximation attacks is to produce authentication of host to server. Within the STW 3- Peke, there's no verifiable data for server to demonstrate host. On the contrary, if there's any verifiable data for server combined with parole can lead to offline approximation attacks. LSH 3-PEKE uses server public keys for this purpose. However this can be not a satisfactory resolution all the days and is impractical for a few environments.

Communication parties have to be compelled to get and verify the general public key of the server, a task that puts a high burden on the user. In fact, key distribution services while not public-keys area unit very often superior in observe than PKI.

III. PROPOSED 3-PARTY KEY AGREEMENT PROTOCOL

Our projected protocol withstands all on-line (Y. Ding & P. Horster, 1995) and offline dead reckoning attacks (C. L. Lin, H. M. Sun, & Hwang, 2000), and doesn't makes use of PKI. Each host and server agrees upon family of independent unidirectional hash functions mistreatment that host authentication is finished once it applies for session key. Host establishes just one occasion key with server mistreatment

that server authentication is finished. Instead of storing an obvious text version of secret we have a tendency to store a technique hash of secret at server. A unidirectional perform may be a perform f such for every x within the domain of f , it's straightforward to cipher y = f (x) , however it's computationally impracticable to search out any x given f (x) .

3.1 Notations

In this research, we use the following notations

A, B	Full principal names
S	Trusted Third Party
$E_K(X)$	Encryption of plaintext block X under key K
$D_K(X)$	Decryption of plaintext block X under key K
K_{AB}	A and B share Key K
$K_{AB}^{N}(X)$	One way hash of X using key KAB N once generated by A and received by B Password of A
P_A	One way hash of password of A
$H(P_A)$	Generator of cyclic group g
P	Large prime number
$A \rightarrow B\ M$	A sends message "M" to B

3.2 Proposed Protocol

In this subsection, we describe the steps involved in detail.

i. A chooses a random number ra and generates $R_A = g^{ra} \pmod p$ then encrypts R_A with $H(P_A)$. After calculating the values sends it to server along with IDs of participating entities.

$$A \rightarrow S \quad ID_A, ID_B, H(P_A)[R_A]$$

ii. After receiving the values sent by A, server S decrypts the packet to get R_A by previously distributed one way hash of password of A. server randomly chooses $rs1$ and $rs2$ and computes ephemeral key with A as follows

$$K_{AS} = (R_A)^{rs1} \pmod p = (g^{ra})^{rs1} \bmod p$$

S generates $g^{rs1} \pmod p$ and $g^{rs2} \pmod p$ and encrypts with $H(P_A)$ and $H(P_B)$ respectively. Using these quantities server establishes ephemeral keys with A and B respectively and server authentication is done. S sends the values to A

$$S \rightarrow A \quad H(P_A)(g^{rs1} \bmod p), H(P_B)(g^{rs2} \bmod p)$$

iii. A decrypts this packet with $H(P_A)$ to get $g^{rs1} \pmod p$ and establishes ephemeral key with $S = (g^{rs1})^{ra} \bmod p$ .A calculates one way as KA S function $F(P, K)$ using Which server authenticates A, since only A knows P_A it can compute this function. As this is a commutative one way hash function (S. Bellovin & M. Merritt, 1993), server need not know host password to evaluate this function. Using one way hash of host password server can calculate predicate function and authenticate host. A sends the following values to B

$$A \rightarrow B \quad F(P, K_{AA}), H(P)(g^{rs2} \bmod p)_{SB}$$

iv. After receiving the values B decrypts it with $H(P_B)$ to get $(g^{rs2} \bmod p)$ .B chooses randomly rb and generates $R_B = g^{rb} \pmod p$ .Then computes ephemeral key for authenticating server as $K_{BS} = (g^{rs2})^{rb} \bmod p$. B calculates one way function $F_B(P_B, K_{BS})$, using which server authenticates B. Password of B and ephemeral session key K_{BS} are seeds for this function. Since only B knows P_B it can compute this function and sends the values to S.

$$B \rightarrow S \quad F_A(P_A, K_{AS}), F_B(P_B, K_{BS}), H(P_B)[R_B]$$

v. Server decrypts it with $H(P_B$ to get R_B and computes ephemeral key $K_{BS} = (g^{rb})^{rs2} \bmod p$. For authentication of A and B server evaluates one way functions $F_A(...)$, $F_B(...)$ server need not know host passwords to evaluate these functions. Using one way hash of host password it can evaluate this function as it is a commutative one way hash function. If it results into true then it confirms that host is genuine. It defines a predicate as $T(H(P), F(P, K), K)$. This evaluates to true if and only if the genuine password P was used to create both $H(P)$ and $F(P, K)$. K can be K_{AS} , K_{BS} for A and B respectively. Encrypts R_B and R_A with K_{AS} , K_{BS} respectively and computes one way hash function using K_{AS} (one time key shared between A and server). Using this host A authenticates the server. Similarly S computes one way hash function $H_{KBS}(R_A, R_B)$ using K_{BS} (one time key shared between B and server) and authenticates B and sends the values to B.

$$S \rightarrow B \quad E_{KAS}(R_B), E_{KBS}(R_A), H_{KAS}(R_A, R_B), H_{KBS}(R_A, R_B)$$

vi. After receiving this B decrypts $E_{KBS}(R_A)$ with K_{BS} and gets R_A . Since K_{BS} is shared between server and B, it ensures B that R_A value is from authentic source. B computes one way hash $H_{KBS}(R_A, R_B)$ using K_{BS} as key and authenticates server. B computes session key with A as $K_{AB} = (R_A)^{rb} \pmod p$. B computes a one way hash $H_{KAB}(N_{AB})$ using K_{AB} and N_{AB} as seeds, where N_{AB} is a random number. This one way hash is used for key confirmation (assures that both parties posses same session key). Since N_{AB} is transmitted in plain there is no need of decryption. One way hash suffices decryption. After computing all the values it sends to A.

$$B \rightarrow AE_{KAS}(R_B), H_{KAS}(R_A, R_B), H_{KAB}(N_{AB}), N_{AB}$$

vii. A decrypts $E_{KAS}(R_B)$ using K_{AS} to get R_B . Since K_{AS} is shared between server and A, it ensures A that R_B value is from authentic source. A computes session key with B as $K (R)^{ra} \pmod p$ .Using K_{AB} and N_{AB} A computes one way hash $H_{KAB}(N_{AB})$ and verifies that B posses same key (K_{AB}) as A. Using K_{AB} , A once again calculates one way hash $H_{KAB}(H_{KAB}(N_{AB}))$ and sends to B.

$$A \rightarrow B \quad H_{KAB}(H_{KAB}(N_{AB}))$$

viii. Finally, after receiving this B computes this one way hash using K_{AB} and verifies that A posses same session key (K_{AB}) as B.

The detail is explained in Fig-2.

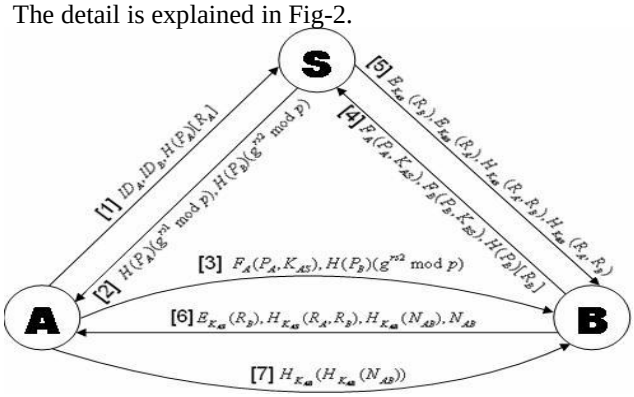

Fig 2: Proposed Protocol.

3.3 Commutative One Way Hash Functions

Both host and serveragree upon family of commutative one-way hash functions $\{H_0, H_1, H_2 ....H_N\}$ (S. Bellovin & M. Merritt, 1993). Let H *(P)* be defined as $H_0(P)$,A member of a family of commutative one way hash functions. Host A calculates one way hash of it password as $H_0(P)_A = (P)^{h}0(\text{mod } p)$, where h_0 is a random number. We assume that one way hash of password $H_0(P)$ of every host is distributed to server. Since one way hash is irreversible nobody can compute P from $H_0(P)$. Host *A* calculates its one way function as

$F(P, K) = H_K(P) = (P^K AS)(\text{mod } p)$ and sends to server. Server Knows only one way hash of

AA AS AS A A

password P_A i.e. $H_0(P_A)$ using which it calculates predicate function of *A* as,

$H_K(H(P)) = (P^{h0})^K AS \ (\text{mod } p)$. Server computes $H(H(P)) = (P^K AS)^{h}0(\text{mod } p)$.

AS 0 *A* *A* $0^K AS$ *A* *A*

Here $H_K AS(P) = (P^K AS)(\text{mod } p)$ is sent by the host. Now server checks $H_K(H(P))$ equals

A *A* *AS 0* *A*

$H_0(H_{K\,AS}(P_A))$ or not. If these two are equal it confirms server that host is genuine. Much better implementation of commutative one way hash functions can be found.

IV. INFORMATION SYSTEMS SECURITY ANALYSES

In this section, we provide formal information systems security analysis of our protocol. Hosts are not forced to store plain text version of password at server as a result this protocol is not vulnerable to password file compromise attacks (S. Bellovin & M. Merritt, 1993). Though $H(P)$ is compromised there is no way to recover *P* from $H(P)$. Even $H(P)$ is compromised nobody can mimic the host to server as only genuine host can compute one way function- $F_A(...)$, $F_B(...)$ etc., Because only host knows password, which is seed for this function.

This protocol provides host authentication and server authentication as a result man-in-the middle attacks are averted. Server authentication is done through one time keys it defeats malicious insider attacks (T. Gene & H. Van, 1993). This is a type of attack where a genuine host turns out to be hostile in subsequent protocol run and misuses the information that it has already acquired in previous protocol run.

Online guessing attacks are not possible since R_A , R_B are encrypted with one time keys. Dictionary attacks and offline guessing attacks are not possible since there is no verifiable information present in the protocol runs to verify attacker's guess. This protocol also provides perfect forward key secrecy. It also provides Key non-disclosure, Key integrity, and Key confirmation. We use one way hash functions for authentication and key confirmation as conventional encryption and decryption makes protocol design messy (T. Gene & H. Van, 1993). One way hash function suffices decryption. N_{AB} in last step multiplies key space to be searched in case of brute force attack. To guard further against dictionary attacks one way function- $F_A(...)$, $F_B(...)$ may be encrypted with K_{AS} , K_{BS} respectively. Even if $H(P)$ is compromised it is equivalent to breaking Diffie-Hellman protocol (1976). Since R_A , R_B are encrypted with $H(P_A)$ and $H(P_B)$ respectively this averts identity mis-binding attacks.

V. CONCLUSION

We propose a third party protocol secure against on-line and information attacks. It provides host and server authentication. Hosts aren't forced to store plain text version of countersign at server. Projected protocol doesn't build use of any public key infrastructure. Rather than independent away hash functions digital signatures also can be used for host authentication purpose. The technologies of the analysis, the synthesis and also the correction of the design of data security area unit investigated. Diagrams security area unit used because the basic construction components in information security systems. Within the comparison with ancient approaches we tend to introduce Associate in nursing integrated approach for the analysis of security systems. It permits to hold out a stratified decomposition of Associate in nursing data security scheme from GDIS to DIS of separate elementary actions. The correction of the design of a security scheme, first of all, is predicated on a division of the system into isolated domains. Samples of correct the divisions into the isolated domains are unit given. At intervals the isolated domain it's potential to construct effective to hold out its stratified decomposition on levels of the bottom. Besides, it's potential to outline segments of EDIS wherever data security risks area unit accepted. All rules of a security policy within the isolated domain have to be compelled to be applied if potential to any or all actions that they treat. Associate in nursing existence of DIS that isn't connected with rules of a security policy is feasible. For instance product with a redundant security measures are often employed in a security scheme

REFERENCES

[1] Menezes A.,Oorschot P. van and Vanstone S. (1996) "*Handbook of Applied Cryptography*", CRC Press.

[2] Schneier Bruce(1994). "*Applied Cryptography: Protocols and Algorithms*", John Wiley and Sons.

[3] Stallings Williams(2004). "*Cryptography and Network Security*", 3rd Edition, Pearson Education.

[4] Mel H.X.,Baker Doris.M. and Burnett Steve(2004). "*Cryptography Decrypted*", Addison- Wesley.

[5] Bellare Mihir, Rogaway Phillip(1995). "Provably Secure Session Key Distribution-The Three Party Case". *In Proceedings of the 27th annual ACM symposium on Theory of computing STOC '95*, ACM Press.
[6] L. Gong, M. A. Lomas, R. M. Needham, and J. H. Saltzer June (1993). "*Protecting Poorly Chose Secrets From Guessing Attacks*". Selected areas in communications, vol. 11, no. 5, pp. 648–656.
[7] M. Strangio, (2006). "An Optimal Round Two-Party Password-Authenticated Key Agreement Protocol". *In The First International Conference on Availability, Reliability and Security*, p. 8.
[8] W. Diffie and M. Hellman, (1976). "New Directions In Cryptography". *IEEE Transactions on Information Theory IT-11*, pp. 644–654.
[9] S. Bellovin and M. Merritt,(1992) "Encrypted Key Exchange: Password Based Protocols Secure Against Dictionary Attacks". *In Proceedings IEEE Symposium on Research in Security and Privacy*, pp. 72–84.
[10] Y. Her-Tyan and S. Hung-Min, (2002) "*Simple Authenticated Key Agreement Protocol Resistant To Password Guessing Attacks*", ACM SIGOPS Operating Systems Review, vol. 36, no. 4, pp. 14–22.
[11] M. Steiner, G. Tsudik, and M. Waidner,(1995) "*Refinement And Extension Of Encrypted Key Exchange". ACM Operating System Review*, vol. 29, no. 3, pp. 22–30.
[12] Y. Ding and P. Horster, (1995) "*Undetectable On-Line Password Guessing Attacks*". ACM Operating System Review, vol. 29, no. 4, pp. 77–86.
[13] C. L. Lin, H. M. Sun, and Hwang, (2000). "*Three-Party Encrypted Key Exchange: Attacks And A Solution*". ACM Operating System Review, vol. 34, no. 4, pp. 12–20.
[14] S. Bellovin and M. Merritt,(1993). "Augmented Encrypted Key Exchange: A Password Based Protocols Secure Against Dictionary Attacks And Password File Compromise*". In 1st ACM Conf. on Computer and Communications Security*. ACM Press, pp. 244–250.
[15] T. Gene and H. Van,(1993). "On Simple and Secure Key Distribution". *In Proceedings of the 1st ACM conference on Computer and communications security CCS 93*. ACM Press, pp. 49–57.

Gnanasekaran P , Research Scholar, Jairams Arts and Science College, Karur, Affiliated to Bharathidasan University, Trichy,Tamilnadu, India.

Umadevi V, Professor & Research Director, Jairams Arts and Science College, Karur, Affiliated to Bharathidasan University, Trichy,Tamilnadu, India.