Survey: Study on Authentication used in Internet Cryptography

Peter Sequeira
Student MCA, TIMSCDR
Mumbai, India

Swati Verma
Student MCA, TIMSCDR
Mumbai, India

Abstract: Authentication and encryption are two intertwined technologies that help to insure that your data remains secure. *Authentication* is the process of insuring that both ends of the connection are in fact who they say they are. This applies not only to the entity trying to access a service (such as an end user) but to the entity providing the service, as well (such as a file server or Web site). *Encryption* helps to insure that the information within a session is not compromised. This includes not only reading the information within a data stream, but altering it, as well. While authentication and encryption each has its own responsibilities in securing a communication session, maximum protection can only be achieved when the two are combined. For this reason, many security protocols contain both authentication and encryption specifications.

Keywords: cryptography, cipher, public key, private key, security, algorithms, password, encryption, digital signatures

I. INTRODUCTION

Key authentication is used to solve the problem of authenticating the keys of the person (say "person B") to whom some other person ("person A") is talking to or trying to talk to. In other words, it is the process of assuring that the key of "person A" held by "person B" does in fact belong to "person A" and vice versa.
This is usually done after we assume that the keys have been shared among the two sides over some secure channel, although some of the algorithms share the keys at the time of authentication also. The simplest solution for this problem is for the two users concerned to meet face-to-face and exchange keys. However, for systems in which there are a large number of users or in which the users do not personally know each other (e.g., Internet shopping) this is not practical. There are various algorithm for both symmetric keys and asymmetric public key cryptography to solve this problem.

Public-key digital certificate has been widely used in public-key infrastructure (PKI) to provide user public key authentication. However, the public-key digital certificate itself cannot be used as a security factor to authenticate user. In this paper, we propose the concept of generalized digital certificate (GDC) that can be used to provide user authentication and key agreement. A GDC contains user's public information, such as the information of user's digital driver's license, the information of a digital birth certificate, etc., and a digital signature of the public information signed by a trusted certificate authority (CA). However, the GDC does not contain any user's public key. Since the user does not have any private and public key pair, key management in using GDC is much simpler than using public-key digital certificate. The digital signature of the GDC is used as a secret token of each user that will never be revealed to any verifier. Instead, the owner proves to the verifier that he has the knowledge of the signature by responding to the verifier's challenge. Based on this concept, we propose both discrete logarithm (DL)-based and integer factoring (IF)-based protocols that can achieve user authentication and secret key establishment.

Authentication is the first step in access control, and there are three common factors used for authentication: something you know, something you have, and something you are. This article provides you with good understanding of the three factors of authentication and how they can be used together with multifactor authentication.

One of the first steps of access control is the identification and authentication of users. There are three common factors used for authentication:

- Something you know (such as a password)
- Something you have (such as a smart card)
- Something you are (such as a fingerprint or other biometric method)

Identification occurs when a user professes an identity (such as with a username), and authentication occurs when users prove their identity. For example, users are authenticated when they provide both their username and correct password. Permissions, rights, and privileges are then granted to users based on their proven identity.

The **first type** of authentication is accepting proof of identity given by a credible person who has first-hand evidence that the identity is genuine. When authentication is required of art or physical objects, this proof could be a friend, family member or colleague attesting to the item's provenance, perhaps by having witnessed the item in its creator's possession. With autographed sports memorabilia, this could involve someone attesting that they witnessed the object being signed. A vendor selling branded items implies authenticity, while he or she may not have evidence that every step in the supply chain was authenticated. This hear-say authentication has no use case example in the context of computer security.
The **second type** of authentication is comparing the attributes of the object itself to what is known about objects of that origin. For example, an art expert might look for similarities in the style of painting, check the location and form of a signature, or compare the object to an old photograph. An archaeologist might use carbon dating to verify the age of an artifact, do a chemical analysis of the materials used, or compare the style of construction or decoration to other artifacts of similar origin. The physics of sound and light, and comparison with a known physical environment, can be used to examine the authenticity of audio recordings, photographs, or videos. Documents can be verified as being created on ink or paper readily available at the time of the item's implied creation. Attribute comparison may be vulnerable to forgery. In general, it relies on the facts that creating a forgery indistinguishable from a genuine artifact requires expert knowledge, that mistakes are easily made, and that the amount of effort required to do so is considerably greater than the amount of profit that can be gained from the forgery.

In art and antiques, certificates are of great importance for authenticating an object of interest and value. Certificates can, however, also be forged, and the authentication of these poses a problem. For instance, the son of Han van Meegeren, the well-known art-forger, forged the work of his father and provided a certificate for its provenance as well; see the article Jacques van Meegeren. Criminal and civil penalties for fraud, forgery, and counterfeiting can reduce the incentive for falsification, depending on the risk of getting caught. Currency and other financial instruments commonly use this second type of authentication method. Bills, coins, and cheques incorporate hard-to-duplicate physical features, such as fine printing or engraving, distinctive feel, watermarks, and holographic imagery, which are easy for trained receivers to verify.

The **third type** of authentication relies on documentation or other external affirmations. In criminal courts, the rules of evidence often require establishing the chain of custody of evidence presented. This can be accomplished through a written evidence log, or by testimony from the police detectives and forensics staff that handled it. Some antiques are accompanied by certificates attesting to their authenticity. Signed sports memorabilia is usually accompanied by a certificate of authenticity. These external records have their own problems of forgery and perjury, and are also vulnerable to being separated from the artifact and lost.

In computer science, a user can be given access to secure systems based on user credentials that imply authenticity. A network administrator can give a user a password, or provide the user with a key card or other access device to allow system access. In this case, authenticity is implied but not guaranteed.

Consumer goods such as pharmaceuticals, perfume, fashion clothing can use all three forms of authentication to prevent counterfeit goods from taking advantage of a popular brand's reputation (damaging the brand owner's sales and reputation). As mentioned above, having an item for sale in a reputable store implicitly attests to it being genuine, the first type of authentication. The second type of authentication might involve comparing the quality and craftsmanship of an item, such as an expensive handbag, to genuine articles. The third type of authentication could be the presence of a trademark on the item, which is a legally protected marking, or any other identifying feature which aids consumers in the identification of genuine brand-name goods. With software, companies have taken great steps to protect from counterfeiters, including adding holograms, security rings, security threads and color shifting ink.

Passwords are the most widely used form of authentication. Users provide an identifier, a typed in word or phrase or perhaps a token card, along with a password. In many systems the passwords, on the host itself, are not stored as plain text but are encrypted. Password authentication does not normally require complicated or robust hardware since authentication of this type is in general simple and does not require much processing power. Password authentication has several vulnerabilities, some of the more obvious are: Password may be easy to guess. Writing the password down and placing it in a highly visible area. Discovering passwords by

eavesdropping or even social engineering. The risk of eavesdropping can be managed by using digests for authentication. The connecting party sends a value, typically a hash of the client IP address, time stamp, and additional secret information. Because this hash is unique for each accessed URI, no other documents can be accessed nor can it not be used from other IP address without detection. The password is also not vulnerable to eavesdropping because of the hashing. The system is, however, vulnerable to active attacks such as the-man-in-the middle attack. To avoid the problems associated with password reuse, one-time passwords were developed. There are two types of one-time passwords, a challenge-response password and a password list. The challenge-response password responds with a challenge value after receiving a user identifier. The response is then calculated from either the response value (with some electronic device) or select from a table based on the challenge. A one-time password list makes use of lists of passwords which are sequentially used by the person wanting to access a system. The values are generated so that it is very hard to calculate the next value from the previously presented values. For example, the S/Key system calculates values xi starting from initial value R: x1=f(R), x2=f(f(R)), ..., xn=f(xn-1). The f() is chosen so that f-1 is very difficult. First the xn is used, then the xn-1 is used. [Applied Cryptography Second Edition: protocols, algorithms, and source code in C, p. 53]. It is important to keep in mind that Password systems only authenticate the connecting party. It does not provide the connecting party with any method of authenticating the system they are accessing, so it is vulnerable to spoofing or a man-in-middle attack.

Public-key cryptography: Public key cryptography is based on very complex mathematical problems that require very specialized knowledge. Public key cryptography makes use of two keys, one private and the other public. The two keys are linked together by way of an extremely complex mathematical equation. The private key is used to decrypt and also to encrypt messages between the communicating machines. Both encryption and verification of signature is accomplished with the public key. The advantage of public-key cryptography is that the public key is readily available to the public. In fact, public-keys are often published to public directories on the Internet so that they can be easily retrieved. This simplifies key-management efforts. The integrity of the public key is of the utmost importance. The integrity of a public key is usually assured by completion of a certification process carried out by a certification authority (CA). Once the CA has certified that the credentials provided by the entity securing the public key are valid, the CA will digitally sign the key so that visitors accessing the material the key is protecting will know the entity has been certified. Basically, the public-key authentication process includes the following: Client selects some random numbers and sends the results to the server as a message: Message 1. The server then sends different random numbers back to the client based on Message 1. The Clients then computes the new value and sends Message 2 to the server. The Server then uses the clients public key to verify that the values returned could have only been computed using the private key. This authenticates the client to the server. If the client wants to authenticate the server, the same procedure is repeated with changed roles.

Zero-knowledge proofs: Zero-knowledge proofs make it possible for a Host to convince another Host to allow access without revealing any “secret information”. The hosts involved in this form of authentication usually communicate several times to finalize authentication. The client will first create a random but difficult problem to solve and then solves it using information it has. The client then commits the solution using a bit-commitment scheme and then sends the problem and commitment to the server. The server then asks the client to either prove that the problems are related or open the committed solution and prove that it is the solution. The client complies with the request. Typically, about ten successful exchanges will be required to take place before the authentication process is complete and access is granted. The zero-knowledge proof can be made to be non-interactively. In this instance only one message from client to server is needed. This method utilizes a one-way hash function where the committing answers are based on the output of that hash function. The number of proofs needed is generally larger (64 or more), to avoid brute-force attacks. The zero-knowledge proof of identity has it share of problems. Perhaps the most vulnerable one is that while Host A thinks he is proving his identity to Host B, it is possible for Host B to simultaneously authenticate to a third party, Host C, using Host A’s credentials.

Digital Signatures: In many instances it is not necessary to authenticate communicating parties; for instance when downloading application updates or patches from the Internet. From a security pointof-view, the server does not need to screen who is downloading the software. The user downloading the software does not necessarily care what particular server it is downloading from. However, the user may want to be assured that the downloadable data is genuine and not a Trojan Horse or other malicious or invalid information. In this instance a digital signature would best serve to authenticate the downloadable data. A digital signature is a digest calculated from a signed document (typically a one-way hash function) which is then signed (encrypted with private key). The client verifies the digest signature by decrypting it with the server’s public key and

compares it to the digest value calculated from the message received. The signature can also be used by the server to verify data the client is sending.

Secure Sockets Layer: Secure Sockets Layer (SSL), developed by Netscape Communications, provides a secure method of communication for TCP connections, especially for HTTP connections. SSL work in this manner: after a TCP connection is established, the client sends a client hello message to which the server responds with a server hello message. The hello messages establish connection attributes which include the protocol version, a session identifier, the cipher suite used, and the compression method in addition to random values for both the server and the client. After the hello messages are exchanged, the server will send its certificate. When the server has been authenticated, depending on the cipher suite used, the server may then request a certificate from the client. After receiving the client hello, the server instructs the client to start using encryption and finishes the initial handshake. The application transfer can now take place. When the client and the server decide to resume a previous session or duplicate an existing session, only the hello messages are exchanged. If the server does not find a matching session identifier, it will assume the connection is a new one. The advantage of resuming previous session is that it saves processing time, which may have a considerable effect on server performance.

IP Sec: The IP Authentication header provides strong authentication and integrity for IP datagrams. Depending on the signing algorithm used, it may also provide non-repudiation, excluding those fields that are changed during transmit, like hop count or time to live. The authentication header has fields for the next header, payload length, security parameters index (SPI: identifies security association (SA) between two hosts), sequence number, and authentication data. [Atkinson, R., "Security Architecture for the Internet Protocol", RFC 1825,] The authentication is transport-protocol independent, so there may be data from more than one different protocol, for instance TCP and UPD. The authentication data is calculated with a message digest algorithm.

To avoid replay attacks, the 32-bit sequence number is not allowed to wrap around; one must establish a new SA and generate new keys. This happens once in 232 packets so, if 1460 byte TCP segments are transferred one can transfer 5.7 TB of data using one SA.
Secure Shell: Secure Shell (SSH) is a protocol for providing secure remote login and other secure network services over an insecure network. With SSH (version 2) each host has a host key, during the connection establishment the client can verify he is talking to the right server. The server keys can be stored locally on the clients or they may be distributed by using a key distribution protocol. [Ylonen, T., Kivinen, T.,, Saarinen, M., Rinne, T., Lehtinen, S., "SSH Protocol Architecture"] After a reliable byte stream is established between the client and the server, host authentication takes place using the transport layer functions. Both ends send version identification. The key exchange begins with both the client and server sending a key exchange initialization packet. The initialization packet contains a list of algorithms for key exchange, keys, encryption, MAC, and the level of compression supported. The server and client may negotiate a different set of algorithms for each direction of data flow. For each category, the best algorithm is chosen that both the client and server support. Kerberos Kerberos authentication was developed at the Massachusetts Institute of Technology (MIT). There are two main components: a ticket, which is used for user authentication and securing data, and an authenticator that is used to verify that the user is the same user to whom the ticket was initially granted. When a user logs into a system, the system connects to the Kerberos server where it retrieves a session key to be used between the user and the ticket granting service (TGS). This is encrypted with a key based on the user's password. If the user provides the right password the end system is able to decrypt the session key. After this is done, the user password is erased from memory to avoid being compromise. The ticket (Ticket granting ticket: TGT) expires after a set amount of time. When a user wants to connect to a service to which he does not already have a ticket, the user connects to the TGS and gets a ticket that can only be used to access the particular service the ticket was granted for. The user can now connect through an encrypted channel to the server. After the ticket expires, the user must request a new one from the TGS. The major issue with Kerberos is its scalability. The Kerberos server must store secret keys for each of the users and each of the TGSs. Kerberos can get very complex in enterprise implementations where trust relationship need to be in place between multiple organizations.

Although Government efforts to broaden the social base has recently been characterized as a limited success.

II. RELATED WORK

- In 2012 a simple and intuitive model for expressing the semantics of privacy-friendly authentication and accountability technologies such as anonymous credentials systems and verifiable encryption. It allows for expressing the precise relations as well as the authentication and accountability properties between parties.

- The concepts cover in the model comprises pseudonyms, attribute-based authentication, as well as conditional release of information. As a result, the model can express the relevant primitives for privacy-preserving authentication and accountability at the same time. Many standards exist for authentication, ranging from simple static passwords stored on a single machine to complicated distributed systems.
- Organizations concerned about protecting their digital assets from sophisticated cyber attacks have begun relying on two-factor authentication as a defense against unauthorized access. These protocols were proven secure in the random oracle model. Katz, Ostrovsky, and Yung (KOY) demonstrated the first efficient PAKE protocol with a proof of security in the standard model. It also achieves mutual authentication in three rounds. In their work , Groce and Katz mentioned their framework will significantly improve efficiency when basing the protocol on lattice assumptions. Katz and Vaikuntanathan first instantiated the KOY/GL PAKE protocol under lattice assumptions. The most technically complex characteristic of their work is the construction of a lattice-based CCA-secure encryption scheme with an associated approximate smooth projective hash system.
- In order to plug into the JG/GK's structure, we use an approximate lattice-based SPH and an error correcting code (ECC) to do the job of an exact lattice-based SPH. In 2012 by Wang, Y.G. observed that the previous papers in this area present attacks on protocols in previous papers and propose new protocols without proper security justification (or even a security model to fully identify the practical threats), which contributes to the main cause of the above failure. Consequently, Wang offered three kinds of security models, specifically Type I, II and III, and further planned four concrete schemes, only two of which, i.e. PSCAb and PSCAV, are claimed to be secure under the harshest model, i.e. Type III security model. PSCAb requires Weil or Tate pairing operations to defend against offline guessing attack and may not be suitable for systems where pairing operations are considered to be too expensive or infeasible to implement. Moreover, PSCAb suffers from the wellknown key escrow problem and lacks some desirable features such as local password update, reparability and user anonymity.
- As for PSCAV, in Appendix B, we will demonstrate that it still cannot achieve the claimed security goals and is vulnerable to an offline password guessing attack and other attacks under the Type III security model. In 2011 a password based authentication using Elliptic Curve Cryptography (ECC) for smart card. Since the secret key of the AS is a long-term key, it requires further security. When the secret key of the AS is compromised, the entire operation of the AS will be disrupted. It is necessary to replace or alter the long term secret key.
- Password-authenticated secret sharing (PASS) methods, first commenced by Bagherzandi et al. at CCS 2011, permit users to allocate data among several servers so that the data can be recovered using a single human-memorizable password, but no single server (or even no collusion of servers up to a certain size) can mount an off-line dictionary attack on the password or learn anything about the data. Further in 2012 present a concrete 2PASS protocol and prove that it meets our definition. Given the strong safety measures guarantees, our protocol is amazingly proficient: in its most efficient instantiation under the DDH assumption in the random oracle model.
- In 2011 the TW-KEAP is an efficient protocol for sharing a session key to protect communication in an insecure network. It is based on the concept of the Diffie-Hellman key exchange protocol which allows the key exchange without session key appearing in the message. The TW KEAP could support lawful interception because the corresponding server is involved in the key exchange procedure to derive the session key.
- In 2011, Maryam Saeed has recommended a new two party IJSER International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013 1082 ISSN 2229-5518 IJSER © 2013 http://www.ijser.org validation protocol without the server's public key in which the limitations of PAKE1 and PAKE2 protocols has been overcome and new authentication protocols has been implemented which can provide several security attributes while it has a remarkable computational efficiency and lower number of rounds.
- In Maryam Saeed, Hadi Shahriar Shahhoseini, "An Improved two party Password Authenticated Key Exchange Protocol without Server's Public Key", IEEE 3rd International Conference on

Communication Software and Networks (ICCSN-2011), pp. 90-95, 2011 is proved that the Hitchcock et al.'s protocol is exposed to momentary key compromise masquerade, Key Compromise Impersonation (KCI) attacks and off-line dictionary while it does not provide the mutual authentication and forward secrecy attributes. It is also shown that SPAKEI and SPAKE2 protocols are vulnerable to password compromise impersonation and Denial-of-Service (DoS) attacks while they do not provide the mutual authentication property.

- To remove the above disadvantages, an efficient secure two-party P AKE protocol is designed to provide several securities attributes while the efficiency is also improved. In 2010 Songs projected extremely recently a password based authentication and key establishment protocol using smart cards which attempts to solve some weaknesses found in a previous scheme suggested by Xu, Zhu, and Feng. In 2009, Lee et al. showed that Juang et al.'s design is not protected against stolen-verifier attack. Furthermore, Juang's method does not convince the user anonymity. To solve this problem, Kyungkug Kim proposed an improved anonymous authentication and key exchange proposal. Then, we demonstrate that the offered scheme is safe and sound against various well-known attacks.

III. DATA COLLECTED

SECURITY ANALYSIS: The security analysis is discussed with respect to the security features which the proposed protocol should satisfy. It is desirable for a two party P AKE protocol to possess the following security attributes:

a. Forward secrecy: If the user's password or the server's private key is divulged, the secrecy of previously established session keys should not be revealed.

b. Known session key security: Disclosure of one session key should not reveal other session keys. c. Resilience to Denning-Sacco attack: Disclosure of session key should not enable an attacker to calculate or guess the password.

d. Resilience to password compromise impersonation attack: Password compromise of any user A should not enable an attacker to share any session key with A by impersonating himself/herself as any other entity.

e. Resilience to Unknown Key Share (UKS) attack: User A should not be coerced into sharing a key with an attacker while he thinks that his key is shared with another user B.

f. Resilience to off-line dictionary attack: If an attacker could guess a password, he should not be able to check his guess offline.

g. Resilience to undetectable on-line dictionary attack: If the attacker could guess a password in an on-line transaction, he should not be able to check the correctness of his guess by using responses from the server and the server is also able to detect an honest request from a malicious request.

h. Resilience to replay attack: An attacker or originator, who captured the exchanged data, should not be able to reuse it maliciously.

i. Resilience to ephemeral key compromise impersonation attack: Disclosure of the ephemeral key of any user A should not enable adversary to share session key with A by impersonating any other participant.

j. Resilience to Key Compromise Impersonation (KCI) attack: Disclosure of the user A's private key should not enable the attacker to masquerade as other participants to A.

k. Resilience to malicious server attack: If an attacker runs on a malicious server and tempts people to register with that server, he/she must not be capable to acquire the passwords of users and impersonate himself/herself as users in login to another server.

l. Resilience to man-in-the-middle attack: The attacker captures and changes the transferred messages between the user and server while two participants are unaware of being attacked by the attacker.

Table 2 gives a brief summary of the least preferred modes of collaboration. It is clearly evident that IT Institutes do not wish to collaborate with industry on their internal issues. Faculty & staff selection, training and development are viewed as in-house activities, where industry's participation is not invited.

In view of government's increased interest in education and the latter being included as crucial issues in WTO debate, there has been increased spending on development of higher education in India

The cash rich private institutions are today less dependent on industry for funding and infrastructure support. They not only have donors but also generate income from consulting, executive education initiatives to support their functions.

IV. RESEARCH & OBSERVATIONS

Cryptographic Prerequisites: Cryptographic mechanisms are fundamental to authentication protocols. Suppose that we have some message text P which we wish to transmit over the network. P is generally referred to as plaintext or a datagram. A cryptographic algorithm converts P to a form that is unintelligible to anyone monitoring the network. This conversion process is called encryption. The unintelligible form is known as ciphertext or a cryptogram. The precise form of the cryptogram C corresponding to a plaintext P depends on an additional parameter K known as the key. The intended receiver of a cryptogram C may wish to recover the original plaintext P. To do this, a second key K 1 is used to reverse the process. This reverse process is known as decryption.

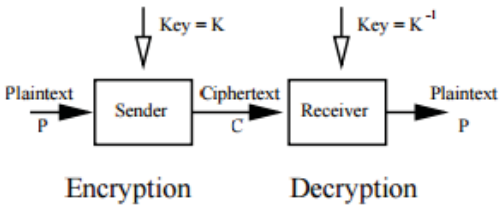

Figure 1: Encryption and Decryption

The classes of encryption and decryption algorithms used are generally assumed to be public knowledge. By restricting appropriately who has access to the various keys involved we can limit the ability to form ciphertexts and the ability to determine the plaintexts corresponding to ciphertexts.

Communication with public key cryptography: Public key cryptography, by using a different key for decrypting than encrypting solves problems of key distribution. If Alice and Bob wish to communicate, Alice sends Bob her public key and Bob gives his public key to Alice. Alice then encrypts her message to Bob with Bob's public key, knowing that only Bob, the possessor of Bob's private key, can decrypt the message. Likewise, Bob encrypts his messages to Alice with Alice's public key. Public keys may be stored in a database or some well-known repository so that the keys do not have to be transmitted. Not only does public key cryptography solve key distribution, it also solves the problem of having [n(n-1)]/2 keys for n users. Now we only need 2n keys (n public and n private).

*Symmetric Key Cryptography***:** In symmetric key cryptography the encryption key K and the decryption key K 1 are easily obtainable from each other by public techniques. Usually they are identical and we shall generally assume that this is the case. The key K is used by a pair of principals to encrypt and decrypt messages to and from each other. Of course, anyone who holds the key can create ciphertexts corresponding to arbitrary plaintexts and read the contents of arbitrary ciphertext messages. To ensure security of communication this key is kept secret between the communicating principals. Following established convention we shall use the notation Kab to denote a key intended for communication between principals A and B using a symmetric key cryptosystem.

*Classical Cryptography***:** Classical cryptography has used symmetric keys. Typically classical ciphers have been either substitution or transposition ciphers (or a mixture) and have worked on text characters. A substitution cipher substitutes a ciphertext character for a plaintext character. A transposition cipher shuffles plaintext characters. The precise substitutions and transpositions made are defined by the key. Examples include simple, homophonic, polyalphabetic and polygram substitution ciphers and simple permutation ciphers (e.g. where successive groups of N characters are permuted in the same way). Elements of transposition and substitution are included in modern day algorithms too. It is not our intention to survey classical approaches to cryptography.

*Modern day Cryptography***:** Modern day symmetric key algorithms are principally block ciphers or stream ciphers. A block cipher will encrypt a block of (typically 64 or 128) plaintext bits at a time. The best known block cipher is the ubiquitous Data Encryption Standard, universally referred to as DES. This has been a hugely controversial algorithm. The controversy has centered on whether the effective key length (56 bits – reduced from 128 at the insistence of the National Security Agency) is really sufficient to withstand attacks from modern-day computing power, and over the design of elements called S-boxes (the design criteria were not made public). The reader is referred to for details. It is worth noting that the algorithm is remarkably resistant to attack using the published state-of-theart cryptanalysis technique known as differential cryptanalysis discovered

by Biham and Shamir in 1988. As revealed by Coppersmith in 1994 this was because the technique was known to the designers of DES back in 1974! Of course, in this survey we can only comment on what is publicly known.
Other examples of block ciphers are MADRYGA (efficient for software implementation and operates on 8-bit blocks), NEWDES (operates on 64- bit blocks but with a 120-bit key), FEAL-N, RC2 and RC4 (by Ronald Rivest) and IDEA (by Lai and Massey). Schneier has written a readable account of the IDEA algorithm. A very good overview of block ciphers (and others) can be found in Schneier's general cryptography text.

Modes of Block Cipher Usage: There are several modes in which a block cipher can be used. Principal ones are:

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)

ECB is the simplest mode. Consecutive blocks of plaintext are simply encrypted using the algorithm. Thus, identical blocks of plaintext are always encrypted in the same way (with the same result). Its security needs to be questioned for specific contexts. An analyst may be able to build up a codebook of plaintext-ciphertext pairs (either known or because he can apply cryptanalytic methods to derive the plaintexts). Also, it is possible to modify messages (e.g. by simply replacing an encrypted block with another). Cipher Block Chaining (CBC) is a relatively good method of encrypting several blocks of data with an algorithm for encrypting a single block. It is one mode in which the widely used Data Encryption Standard (DES) can be employed. Block i of plain text is exclusively-ored (hereafter XORed) with block i - 1 of ciphertext and is then encrypted with the keyed block encryption function to form block i of ciphertext. For example, with initialisation block I the encryption of message block sequence P_1 P_1 …. P_n with key K denoted by E(K : P_1 P_2….. P_n) is given by

$E(K : P_1 P_2 \ldots\ldots P_n) = C_0 C_1 C_2 \ldots\ldots C_n$
where
$C_0 = I$

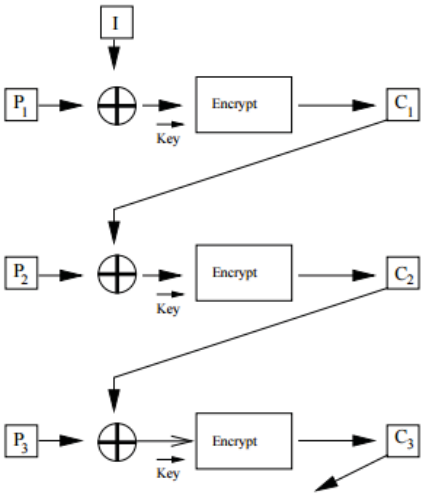

Figure 2: Cipher Block Chaining

VI. CONCLUSION

- User authentication can be handled using one or more different authentication methods. Some authentication methods such as plain password authentication are easily implemented but are in general weak and primitive.
- The fact that plain password authentication it is still by far the most widely used form of authentication, gives credence to the seriousness of the lack of security on both the Internet and within private networks.

- Other methods of authentication, that may be more complex and require more time to implement and maintain, provide strong and reliable authentication (provided one keeps its secrets secret, i.e. private keys and phrases).
- That being said, one of the key factors to be considered in determining which method of authentication to implement is usability. The usability factor cannot be ignored when designing authentication systems. If the authentication methods are not deemed usable by those forced to utilize them, then they will avoid using the system or persistently try to bypass them. Usability is a key issue to the adoption and maintenance of a security system.

VIII. FUTURE RESEARCH

➢ Here in this paper we will provide the literature survey on the basis of different PAKE techniques and the different ways of providing authentication to the user. We will only provide the survey of the work that had been done so far. In the next step we provide the simulation of the proposed work in the PAKE technique and analyse on the basis of different parameters.

➢ This is just an overall survey of what we have studied so far regarding different authentication techniques. In the next paper we implement an efficient algorithm for password authentication using one time private key which provides more security features as compared to the other existing techniques of authentication.

IX. ACKNOWLEDGMENT

Authors thank Mrs. Vinita Gaikwad, Prof. Pankaj Mudholkar, Mrs. Mira Gohil and faculty members at TIMSCDR, Mumbai for their helpful comments and suggestions.

REFERENCES

[1] Generalized Digital Certificate for User Authentication and Key Establishment for Secure Communications by Harn, L. Dept. of Comput. Sci. & Electr. Eng., Univ. of Missouri-Kansas City, Kansas City, MO, USA and Jian Ren
[2] Understanding the Three Factors of Authentication by Darril Gibson
[3] An Overview of Different Authentication Methods and Protocols, Richard Duncan, October 23, 2001
[4] Schneider, B., "Applied Cryptography Second Edition: protocols, algorithms, and source code in C", John Wiley & Sons, Inc., 1996.
[5] Atkinson, R., "IP Authentication Header", RFC 1826, NRL, August 1995. ftp://ftp.isi.edu/in-notes/rfc1826.txt
[6] Alvaro Retana, Don Slice, Russ White, “Advanced IP Network Design” Cisco Press, 1999
[7] A Comparison Study on Key Exchange-Authentication
[8] protocol, International Journal of Computer Applications (0975 – 8887),Volume 7– No.5, September 2010 by Razieh
[9] Mokhtarnameh, Nithiapidary Muthuvelu, Sin Ban Ho, Ian Chai
[10] A Survey of Two-Party Password Authentication Key Exchange Namita Raghuwanshi, Prof. Amit Saxena Truba Institute of Engg. & Technology, Bhopal, India
[11] J. Katz and V. Vaikuntanathan “Password-based Authenticated Key Exchange Based on Lattices”, In Advances in Cryptology, volume 5912 of LNCS, pp. 636–652. Springer, 2009