CHAPTER 2

LITERATURE REVIEW

Bellovin and Merritt (1992) proposed augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise. Since then, much effort has been made on the development and cryptanalysis of such 2PAKE (two-party password-based authenticated key exchange) protocol. Because 2PAKE protocol is suitable only for the client-server architecture, some researchers extended 2PAKE protocols into augmented encrypted key exchange protocol.

The new protocol, augmented encrypted key exchange (A-EKE), works by choosing particular functions $H(P)$ for host storage of passwords. Both sides use $H(P)$ which should be secret, as the shared password in the EKE exchange. However, under some comparatively-rare circumstances, $H(P)$ might be compromised, the user must send an additional message containing a different one-way function of the password. This value, together with $H(P)$ and the session key, is used by the host to validate the login sequence. Most existing 3PAKE protocols are designed for the client-client-server architecture, in which each client (user) shares his password with a trusted server and requests the server to authenticate the peer for establishing a session key.

Sung Bae Choi et al (2011) proposed some countermeasures to eliminate the security vulnerability of the S-3PAKE. These 3PAKE protocols allow users to communicate securely over public networks simply by using
easy-to-remember passwords. In the 3PAKE protocols, each user can exchange session keys with other users securely via the remote server. The remote server authenticates users by encrypting sending messages with personal passwords. Only valid users can decrypt the received messages with their own passwords and derive the correct common session keys for their subsequent communications.

The 3PAKE technology is widely deployed in various kinds of applications. S-3PAKE protocol is still insecure to undetectable on-line dictionary attacks. For this reason, S-3PAKE protocol cannot be used for practical application. It is important that security engineers should be made aware of this, if they are responsible for the design and development of 3PAKE systems. Improving the S-3PAKE protocol can be able to provide greater security and provides computation efficiency.

Maw-Jinn Tsaur, Wei-Chi Ku, Jenn-Wei Lin, Yu-Ze Shen proposed an An Improved Provably Secure Three-party Key Exchange Protocol. In this proposed protocol, the server $S$ can verify that $X$ is indeed generated by the protocol initiator $A$ and $Y$ by the responder $B$. Assume that the system parameters $g, M, N$, and the long-term public key of $S$, $gs$, where $s$ is the long-term private key of $S$, are known to each user during registration. The multiplication, division, and exponential modular operations are performed under modulo $p$. The Proposed protocol is an improved protocol that can resist password guessing attacks. Additionally, it offers a formally provable security theorem to evaluate the AKE security of the proposed protocol.

Cheng-Chi Lee, Shun-Der Chen and Chin-Ling Chen proposed a three-party encrypted key exchange (3PEKE) protocol with password authentication which is called CCLC-3PEKE. The protocol simultaneously
possesses round and computation efficiencies. However, the protocol is vulnerable to replaying attacks. Since the protocol is currently one of the most superior of all 3PEKE protocols, it seems valuable to remedy the security weakness and enhance their efficiency.

Feng Hao and Peter Ryan (1998) proposed a Password Authenticated Key Exchange protocol. Password-Authenticated Key Exchange (PAKE) establishes secure communication between two remote parties solely based on their shared password, without requiring a Public Key Infrastructure (PKI). Despite extensive research in the past decade, this problem remains unsolved. Patent has been one of the biggest brakes in deploying PAKE solutions in practice. Besides, even for the patented schemes like EKE and SPEKE, their security is only heuristic; researchers have reported some subtle but worrying security issues. This protocol proposed to tackle this problem using an approach different from all past solutions.

This protocol, Password Authenticated Key Exchange by Juggling (J-PAKE), achieves mutual authentication in two steps: first, two parties send ephemeral public keys to each other; second, they encrypt the shared password by juggling the public keys in a verifiable way. The first use of such a juggling technique was seen in solving the Dining Cryptographers problem in 2006. Here, the PAKE problem, and the protocol is zero-knowledge as it reveals nothing except one-bit information: whether the supplied passwords at two sides are the same. With clear advantages in security, the scheme has comparable efficiency to the EKE and SPEKE protocols.

Raphael et al (2008) proposed the Cryptanalysis of simple three-party key exchange protocol. Password-authenticated key exchange (PAKE) protocols allow parties to share secret keys in an authentic manner based on
an easily memorizable password. A two-party password-based authenticated key exchange (PAKE) protocol establishes a shared secret key between two parties, which is then used to secure subsequent communication between them. Authentication of parties during the PAKE session is based on knowledge of a shared low-entropy password. This concept is been extended to three parties, e.g. two clients and a trusted server or key distribution center (KDC). The idea of S-3PAKE is a treatment of the security against known protocol attacks. The security of a protocol is a difficult task to analyze and should be handled rigorously using the provable security approach, where a formal security model is defined and the security of the designed protocol is analyzed in the model, without assuming what specific attack will be mounted by the adversary.

The approach is based on formal security models that do not assume on any specific attack method that an adversary may use. Instead a communication model is defined that describes how parties within the protocol, as well as an adversary, communicate with each other, and what sort of information formalized via the notion of oracle queries, is available to or may be under the control of the adversary. Then, security properties of a protocol are defined as one or more games each intended to capture a security property, played by the adversary within the pre-defined communication model.

A protocol is secure with respect to the defined security properties if the adversary’s advantage in winning the game(s) is negligible, and further that the task of an adversary winning is reduced to computationally intractable assumption(s). Once proven secure, a protocol is guaranteed to resist attacks by any adversary who works within the communication model regardless of what specific attacks are mounted, as long as the assumptions remain
intractable. However, defining an appropriate model is not a trivial task, because not including some types of queries, e.g. the corrupt query, or improperly defining the adversarial game may result in a security proof that fails to capture valid attacks.

Abdalla and Pointcheval (2005) suggested a new variation of the computational DH assumption called Chosen based Computational Diffie Hellman (CCDH) and presented SPAKE-1 and SPAKE-2 simple password based authenticated key exchange protocols. Since then several three party password authenticated key agreement protocols are proposed based on CCDH assumption but most of them broken. Abdalla proposes two passwords based simple three party key exchange protocols via twin Diffie -Hellman problem and shown that the proposed protocol provide greater security and efficiency than the existing protocols.

The protocol is verified using Automated Validation of Internet Security Protocols and Applications (AVISPA) which is a push button tool for the automated validation of security protocols and the result shows that they do not have any security flaws. The heart of their method is a trapdoor test which can be used to implement an effective decision oracle for the twin Diffie-Hellman problem, without knowing any of the corresponding discrete logarithms. They applied the trapdoor test to many new variant protocols based on the Diffie-Hellman problem.

Her-Tyan Yeh et al (2003) proposed an Efficient Three-Party Authentication and Key Agreement Protocols resistant to Password Guessing Attacks. Three-party EKE is proposed to establish a session key between two clients through a server. However, three-party EKE can insecure against undetectable on-line and off-line password guessing attacks. It is proposed
that an enhanced three-party EKE to withstand the security risk in three-party EKE.

It also proposes a verifier-based three-party EKE, that is more secure than a plaintext-equivalent mechanism in which a compromise of the server’s database will not result in success in directly impersonating clients. The users use the server’s public key to encrypt some messages such as confounders or passwords and sends it to the server. After decrypting the message and authenticating the users, the server chooses the common session key and transfers it to the users encrypted by password or confounder. If the server’s private key is revealed, the attacker can get the password directly or by using a guessing attack to acquire the password and gain the common session key. This breaks the extended perfect forward secrecy.

Hung-Yu Chien (2011) proposed a Secure Verifier-Based Three-Party Key Exchange in the Random Oracle Model. A Three Party password Authenticated Key Exchange protocol (3PAKE) facilitates two clients to establish authenticated session keys via the help of a trusted server. This approach enhances the scalability of key agreement issue and facilitates users’ convenience in distributed environments. The scheme is efficient, and the key indistinguishability is proved relative to the computational Diffie-Hellman problem. It is the first provably secure verifier-based 3PAKE protocol. Three-party password authenticated key exchange protocol (3PAKE) protocol enables two clients to establish authenticated session keys via the help of a trusted server, where the sever maintains clients’ password as its secret key.

In this approach, each client only keeps one simple password instead of hard-to-memorize secrets. Therefore, it enhances the scalability of secret key establishment in distributed environments and facilitates users’
convenience. However, due to the complicated interactions in the three-party case and the low entropy of passwords, it is a challenge and an error-prone job to design efficient and secure 3PAKE protocols. The weaknesses of guessing attacks are neglected. In addition to the threats (like impersonation attack, replay attack, man-in-the-middle attack, etc.) that the general authentication schemes are faced with, the low-entropy passwords incur still potential security threats—the off-line guessing attack, the on-line guessing attack and the on-line un-detectable guessing attack.

Mihir Bellare et al (2000) proposed an Authenticated Key Exchange Secure against Dictionary Attacks, which is a password-based protocol for authenticated key exchange (AKE). It is designed to work despite the use of passwords drawn from a space so small that an adversary might well enumerate, off line, all possible passwords. While several such protocols are being suggested, the underlying theory has been lagging. This approach defines a model for this problem, one rich enough to deal with password guessing, forward secrecy, server compromise, and loss of session keys. This one model can be used to define various goals. AKE (with “implicit” authentication) is taken as the basic goal and for entity-authentication goals. Then it is proved with correctness for the idea at the center of the Encrypted Key-Exchange (EKE) protocol, security in an ideal-cipher model, of the two-flow protocol at the core of EKE.

Michel Abdalla Pierre-Alain Fouque David Pointcheval (2005) proposed Password-Based Authenticated Key Exchange in the Three-Party Setting. Password-based key exchange protocols assume a more realistic scenario in which secret keys are not uniformly distributed over a large space, but rather chosen from a small set of possible values (a four-digit pin, for example). They also seem more convenient since human-memorable
passwords are simpler. The vast majority of protocols found in practice does not account, however, for such scenario and are often subject to so-called dictionary attacks. Dictionary attacks are attacks in which an adversary tries to break the security of a scheme by a brute-force method, in which it tries all possible combinations of secret keys in a given small set of values. Several protocols are designed to be secure even when the secret key is a password.

The goal of this protocol is to restrict the adversary’s success to on-line guessing attacks only. In the attack, the adversary must be present and interact with the system in order to be able to verify whether its guess is correct. In order to limit the number of passwords that each user needs to remember, we consider in this protocol the password based authenticated key exchange in the 3-party model is considered, where each user only shares a password with a trusted server. The main advantage of this solution is that it provides each user with the capability of communicating securely with other users in the system, while only requiring it to remember a single password. This seems to be a more realistic scenario in practice than the one in which users are expected to share multiple passwords, one for each party with which it may communicate privately. Its main drawback is that the server is needed during the establishment of all communications.

Chin-Chen Chang and Ya-Fen Chang (2003) proposed a novel three-party encrypted key exchange protocol. In three-party key exchange protocols with password authentication, clients are allowed to share an easy-to-remember password with a trusted server such that two clients can communicate with each other through a common secret key without the existence of redundant keys. Such protocols are quite suitable for application when light-weight clients need secure communications. Each user shares an easy-to-remember password with a trusted server S, and S acts as a
coordinator between two communication parties to complete the mutual authentication. In 3PEKE protocol the trusted server holds a permanent and publicly known server’s public key to prevent both of the password guessing attacks that STW-3PEKE suffers from. However, the approach of applying server’s public keys puts a burden on the users because they have to verify the server’s public keys. However, the number of rounds needed in LSSH-3PEKE is two more than that in LSH- 3PEKE.

Taking round efficiency into consideration, trapdoor functions is employed to propose a brand new 3PEKE protocol, which not only has the same properties as LSSH-3PEKE but also provides the same round efficiency as LSH-3PEKE. On the other hand, the trapdoor functions can be constructed from one-way hash functions. If the poly-to one trapdoor functions is employed, it implies that an additional certificate will be needed. Brand new 3PEKE protocol is proposed which possesses the advantages of LSH-3PEKE and LSSH-3PEKE. According to the analyses, it is obvious that the proposed protocol is secure, efficient, and practical. Moreover, it provides another solution to 3PEKE, especially in an environment where users communicate with other users frequently but cannot be expected to validate the server’s public keys correctly.

Fuw-Yi Yang et al (2009) proposed an Enhanced of Simple Three-Party Key Exchange Protocol. A simple three-party password authenticated key exchange (S-3PAKE) protocol assists users to complete business negotiations and agreements during a communication process without requiring a public server key, and the S-3PAKE protocol can resist all known attacks. Two parties execute the 2-PAKE protocol to obtain an agreement key first, so that, when they execute the 3PAKE protocol, they can authenticate the real identity with each other.
An effective three-party password authenticated key exchange (3PAKE) protocol was proposed for authenticating the real identity of both parties without requiring an execution of the 2-PAKE protocol to obtain the agreement key first, but, simply adding the real identity of the opposite party in the communication process, to achieve the same security effect. If two users want to share a session key, each user must share a set of passwords issued by an eligible organization whose server stores the user passwords, and provides the passwords for another user to authenticate their identities. Although this protocol allows users to authenticate the identities with each other, the protocol still face a high risk of being attacked by the password guessing attacks. However, users must confirm their identity with a server in advance when they use the server key mechanism, and thus such arrangement incur a higher consumption cost to users.

Pathak and Manju Sanghi (2011) proposed a Simple Three Party Key Exchange Protocols via Twin Diffie-Hellman Problem, a new variation of the computational DH assumption called chosen based computational Diffie Hellman (CCDH) and presented SPAKE-1 and SPAKE-2 simple password based authenticated key exchange protocols. Since then, several three party password authenticated key agreement protocols have been proposed based on CCDH assumption, but, most of them broken.

Two password based simple three party key exchange protocols via twin Diffie-Hellman problem were proposed and shown that the proposed protocol provide greater security and efficiency than the existing protocols. The protocol is also verified using Automated Validation of Internet Security Protocols and Applications (AVISPA) which is a push button tool for the automated validation of security protocols and the result shows that they do not have any security flaws. STPKE is vulnerable to undetectable online
password guessing attack. Also, found that modified STPKE protocol also cannot resist man in the middle attack.

David Jablon (1996) proposed Strong Password-Only Authenticated Key Exchange protocol. It belongs to an exclusive class of methods which provide authentication and key establishment over an insecure channel using only a small password, without risk of off-line dictionary attack. SPEKE and the closely-related Diffie-Hellman Encrypted Key Exchange (DHEKE) are examined in light of both known and new attacks, along with sufficient preventive constraints. Although SPEKE and DH-EKE are similar, the constraints are different.

The class of strong password-only methods is compared to other authentication schemes. These methods are important for several uses, including replacement of obsolete systems, and building hybrid two-factor systems, where independent password-only and key-based methods can survive a single event of either key theft or password compromise. Passwords must be memorized, and are thus small, while keys can be recorded, and can be much larger. The problem is that most methods need keys that are too large to be easily remembered.

Chun-Li Lin et al (1995) proposed Three-party Encrypted Key Exchange without Server Public-Keys. Three-party key-exchange protocols with password authentication clients share an easy-to-remember password with a trusted server only are very suitable for applications requiring secure communications between many light-weight clients (end users); it is simply impractical that every two clients share a common secret. Three-party EKE protocol (hereafter referred to as STW-3PEKE) is one in which all clients
share a password with a trusted server S and S mediates between two communication parties to allow their mutual authentication.

The three-party EKE protocol is particularly well-suited for applications that require secure communication between many light-weight and mobile clients (end users). On one hand, it is impractical that every two clients share a common secret. A heavyweight infrastructure, e.g., public keys and a public key infrastructure, is often not tolerable. The approach of using server public-keys in 3PEKE is suitable, when the number of message exchanged is of most concern. However, communication parties have to obtain and verify the public-key of the server, a task which puts a high burden on the user. In fact, key distribution services without public-keys are quite often superior in practice than PKIs or are at least widely deployed. Unfortunately, traditional three-party key distribution services such as Kerberos are all susceptible to dictionary attacks with weak passwords and do not immediately provide forward-security. 3PEKE protocol which is resistant to both offline and undetectable on-line password guessing attacks but, does not require server public keys. As such, it could serve as the basis for a key distribution service overcoming the deficiencies of Kerberos.

Hae-Soon Ahn and Eun-Jun Yoon (2010) proposed Cryptanalysis of Chang-Chang’s EC-Paka protocol which is security improvements on elliptic curve authentication key agreement protocol. An elliptic curve based password authenticated key agreement (EC-Paka) protocol is more efficient in terms of computation than other authentication key agreement protocols. But, EC-Paka protocol cannot resist the off-line password guessing attack. Password-based AKA protocols can be vulnerable to password guessing attacks because users usually choose easy-to-remember passwords.
Unlike typical private keys, the password has limited entropy, and is constrained by the memory of the user. The goal of the attacker, which is to obtain a legitimate communication party’s password, can be achieved within a reasonable time. Therefore, the password guessing attacks on the password-based AKA protocols should be considered a real possibility. EC-PAKA protocol cannot be used for practical application, especially in the resource-limited environments and real-time systems. Improving the Chang-Chang’s EC-PAKA protocol which can be able to provide greater security and to be more efficient than the existing EC-PAKA protocols by an accurate performance analysis.

Sung-Bae Choi et al (2011) proposed Two undetectable on-line dictionary attacks on Debiao et al’s S-3PAKE protocol. Debiao pointed out that S-3PAKE protocol proposed by Lu and Cao for password-authenticated key exchange in the three-party setting is vulnerable to an off-line dictionary attack. Then, some countermeasures to eliminate the security vulnerability of the S-3PAKE is been proposed. Nevertheless, this points out that the enhanced S-3PAKE protocol is still vulnerable to undetectable on-line dictionary attacks. Three-party password-based authenticated key exchange (3PAKE) protocols are extremely important security technologies to secure communications and are now extensively adopted in various network communications. These 3PAKE protocols allow users to communicate securely over public networks simply by using easy-to-remember passwords. In the 3PAKE protocols, each user can exchange session keys with other users securely via the remote server. The remote server authenticates users by encrypting sending messages with personal passwords; only valid users can decrypt the received messages with their own passwords and derive the correct common session keys for their subsequent communications.
It is claimed that the enhanced S-3PAKE protocol (Debiao-S-3PAKE) is secure to the off-line dictionary attack. Nevertheless, it points out that Debiao-S-3PAKE protocol is still vulnerable to undetectable on-line dictionary attacks in which an attacker exhaustively enumerates all possible passwords in an on-line manner to determine the correct one. Debiao-S-3PAKE protocol is not secure to undetectable on-line dictionary attacks by any other registered user. Password-based authentication protocols can be vulnerable to dictionary attacks because users usually choose easy-to-remember passwords. Unlike typical private keys, the password has limited entropy, and is constrained by the memory of the user.

Kazukuni Kobara and Hideki Imai (2002) proposed a pretty-simple password-authenticated key-exchange protocol, and is proven to be secure in the standard model under the following three assumptions. (1) DDH (Decision Diffie-Hellman) problem is hard. (2) The entropy of the password is large enough to avoid on-line exhaustive search (but not necessarily off-line exhaustive search). (3) MAC is selectively unforgeable against partially chosen message attacks, (which is weaker than being existentially unforgeable against chosen message attacks).

The on-line attack is a serial exhaustive search for a secret performed on-line using a server that verifies the secret, and the off-line attack is that performed off-line in parallel using recorded transcripts of a protocol. While the on-line attacks can be prevented by letting the server take appropriate intervals between invalid trials, the off-line attacks cannot be prevented by such measures since the attack is performed off-line and independently of the server. Thus, the off-line attacks are critical to most of the protocols using human-memorable passwords not having enough entropy to avoid off-line exhaustive search. While PKI (Public-Key Infrastructures)
can realize an authenticated key exchange or key-transport (being secure against off-line attacks) like SSH (Secure SHell), SSL/TLS (Secure Socket Layer/Transport Layer Security), Station to Station protocol and the protocols in do, that the receivers of public-keys must verify them using the fingerprints (digests) or the verification keys of digital signatures attached with it. This means, the entities must carry about something, which is hard to remember. On the other hand, PAKE (Password-Authenticated Key-Exchange) protocols do not require its entities to carry something hard to remember (except a password) to verify something. Unfortunately, the protocol proposed is too inefficient to use in practice, since it employs techniques from generic multi-party computations, such as non-malleable commitments, secure polynomial evaluations and zero-knowledge proofs. While is more efficient than, it still requires large communication costs and computation costs.

Whitfield Diffie and Martin Hellman (1976) proposed the New Directions in Cryptography. Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing gave rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. It suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing. Public key distribution systems offer a different approach to eliminating the need for a secure key distribution channel. In such a system, two users who wish to exchange a key communicate back and forth until they arrive at a key in common. A third party eavesdropping on this exchange must find it computationally infeasible to compute the key from the information over heard.
A second problem, amenable to cryptographic solution which stands in the way of replacing contemporary business communications by teleprocessing systems is authentication. In current business, the validity of contracts are guaranteed by signatures. The security of most cryptographic systems is in the computational difficulty to the cryptanalyst discovering the plaintext without knowledge of the key. This problem falls within the domains of computational complexity and analysis of algorithms, two recent disciples which study the difficulty of solving computational problems. Using the results of these theories, it may be possible to extend proofs of security to more useful classes systems in the foreseeable future.

Whitfield Diffie et al (1992) proposed Authentication and Authenticated Key Exchanges. Two-party mutual authentication protocols providing authenticated key exchange, focusing on using asymmetric techniques. A simple, efficient protocol referred to as the station-to-station (STS) protocol is introduced, examined in detail, and considered in relation to existing protocols. The definition of a secure protocol is considered, and desirable characteristics of secure protocols are discussed.

The goal of an authentication protocol is to provide the communicating parties with some assurance that they know each other’s true identities. In an authenticated key exchange, there is the additional goal that the two parties end up sharing a common key known only to them. This secret key can then be used for some time thereafter to provide privacy, data integrity, or both. The main problem here is that the challenged party has no influence over what is signed. (As a general rule, it is better if both parties have some influence over the quantity signed.) The challenger can abuse this protocol to get a signature on any quantity he chooses.
Jonathan Katz et al. (1994) proposed Efficient and Secure Authenticated Key Exchange Using Weak Passwords. A 3-round protocol was proposed for password-only authenticated key exchange, and provides a rigorous proof of security for protocol based on the decisional Diffie-Hellman assumption. The protocol assumes only public parameters i.e., a common reference string which can be hard-coded into an implementation of the protocol, in particular, and in contrast to some previous work, this protocol does not require either party to generate and share a public key in addition to sharing a password.

The protocol is also remarkably efficient, requiring computation only 4 times greater than classical Diffie-Hellman key exchange which provides no authentication. Protocols for mutual authentication of two parties and generation of a cryptographically strong shared key between them (authenticated key exchange) are fundamental primitives for achieving secure communication over public, insecure networks. Indeed, protocols for mutual authentication are necessary because one needs to know with that one is communicating, while key-exchange protocols are required because private-key encryption schemes and message authentication codes rely on shared cryptographic keys must be refreshed periodically.

Higher-level protocols are frequently developed and analyzed assuming the existence of secure channels between all parties, yet this assumption cannot be realized without a secure mechanism for implementing such channels using previously shared information. Unfortunately, it is much more common for users to share weak, low-entropy passwords which may be chosen from a known space of possibilities (say, a dictionary of English words). In this case, the problem becomes much more difficult as one must ensure that protocols are immune to off-line dictionary attacks in which, an
adversary exhaustively enumerates all possible passwords in an attempt to determine the correct one.

Taekyoung Kwon (2000) presents a new password authentication and key agreement protocol, AMP, based on the amplified password idea. The intrinsic problems with password authentication are, the password itself has low entropy and the password file is very hard to protect. The amplified password proof and the amplified password file for solving these problems are presented. A party commits the high entropy information and amplifies the password with that information in the amplified password proof. The party never shows any information except that it knows it. The amplified password proof idea is very similar to the zero-knowledge proof in that sense.

A server stores the amplified verifiers in the amplified password file that is secure against a server file compromise and a dictionary attack. AMP mainly provides the password-verifier based authentication and the Diffie-Hellman based key agreement, securely and efficiently. AMP is easy to generalize in any other cyclic groups. In spite of those plentiful properties, AMP is actually the most efficient protocol among the related protocols due to the simultaneous multiple exponentiation method. Among them, AMPn is actually the basic protocol describes the amplified password proof idea while AMP is the most complete protocol that adds the amplified password file. AMPi simply removes the amplified password file from AMP. In the end, a comparison to the related protocols in terms of efficiency is given. Password file protection is another problem that makes password authentication more unreliable. If a password file is compromised, it is vulnerable to dictionary attacks.
Duncan Wong et al (2003) proposed a RSA-based password authenticated key exchange scheme which supports short RSA public exponents. The scheme is the most efficient one among all the RSA-based schemes currently proposed when implemented on low-power asymmetric wireless networks. It is observed that its performance can further be improved by proposing two modifications. The first modification shortens the size of the message sent from the server to the client. The second modification dramatically reduces the size of the message sent from the client to the server and therefore, can be used to reduce the power consumption of the client for wireless communications in a significant way. Generalize the modified schemes and formalize the security requirements of all underlying primitives that the generic scheme is constituted.

Password authenticated key exchange scheme is the same as a conventional authenticated key exchange scheme, after two communicating parties successfully executing the scheme, each of them should have certain assurance that it knows each other’s true identity (authentication), and it shares a new and random session key only with each other, and the key is derived from contributions of both parties (key exchange). Unlike a cryptographic-key authenticated key exchange scheme, the two communicating parties do not have any pre-shared cryptographic symmetric key, certificate or support from a trusted third party. Instead, they only share a password. The major difficulty in designing a secure password-based protocol is due to the concern of implicated off-line dictionary attacks against a small password space. A password, a passphrase, or a PIN (Personal Identification Number) generally needs to be easy to remember. Usually, it has significantly less randomness than its length suggested or is simply very short in length. In this approach, the password space is considered to be so small that an
adversary can enumerate it efficiently. Later it was found to be insecure, and has to use a large prime for the public exponent. This defeats the purpose of using RSA for low-power clients in target applications because the computational complexity of performing a RSA encryption is no less than that of carrying out a modular exponentiation in a Diffie-Hellman key exchange based protocol.

Victor Boyko et al (1998) proposed Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. This approach presents a new protocol called PAK which is the first Diffie-Hellman-based password-authenticated key exchange protocol to provide a formal proof of security (in the random oracle model) against both passive and active adversaries. In addition to the PAK protocol that provides mutual explicit authentication, it shows a more efficient protocol called PPK that is provably secure in the implicit-authentication model. PAK is extended to a protocol called PAK-X, in which one side (the client) stores a plaintext version of the password, while the other side (the server) only stores a verifier for the password. It formally proves security of PAK-X, even when the server is compromised.

Two entities only share a password, and are communicating over an insecure network, want to authenticate each other and agree on a large session key to be used for protecting their subsequent communication. This is called the password-authenticated key exchange problem. If one of the entities is a user and the other is a server, then this can be seen as a problem in the area of the remote user access. Many solutions for remote user access rely on cryptographically secure keys, and consequently have to deal with issues like key management, public-key infrastructure, or secure hardware. Many solutions those are password-based, like telnet or Kerberos, have problems that range from being totally insecure (telnet sends passwords in the clear) to
being susceptible to certain types of attacks. This approach presents a new password-authenticated key exchange protocol called PAK (Password Authenticated Key exchange), which provides perfect forward secrecy and is proven to be as secure as Decision Diffie-Hellman in the random oracle model. PAK does not require an ideal block cipher assumption for security, but, has a more complicated proof.

In addition to PAK, a more efficient 2 round protocol called PPK (Password-Protected Key exchange) that is provably secure in the implicit authentication model is shown. Extending PAK to a protocol called PAK-X, in which one side (the client) stores a plaintext version of the password, while the other side (the server) only stores a verifier for the password. It is common knowledge that users cannot remember long random numbers, hence if the user is required to know a large secret key (either symmetric or private/public) then these keys will have to be stored on the user's system. Furthermore, keeping this secret requires an extra security assumption and introduces a new point of weakness. Even if a user is required to know some public but non-generic data, like the server's public key, this must be stored on the user's system and requires an extra assumption that the public key cannot be modified. In either case, (1) there is a significant increase in administration overhead because both secret and public keys have to be generated and securely distributed to the user's system and the server, and (2) this would not allow for users to walk up to a generic station that runs the authentication protocol and be able to perform secure remote authentication to a system that was previously unknown to that station.

Yung Cheng LEE et al (2011) proposed an Improvement on the Password Authenticated Three-Party Key Agreement Protocol. Key agreement mechanisms provide users to establish a session key for secure
communications over networks. One of the most important security services of key agreement is user authentication. Password authentication schemes are the simplest and most convenient authentication mechanisms in a network environment to protect unauthorized access. In this approach protocol cannot withstand guessing attack, that is, an adversary can obtain the password with the intercepted messages.

An improved password authenticated three-party key agreement protocol to overcome the flaw is also proposed. The proposed protocol not only can resist guessing attack and replay attack, but also has the merits of key secrecy such that any adversary, trust third party and other clients cannot know any information about session keys, and none of other clients can affect the computation result of the session key. Moreover, the protocol provides perfect forward secrecy. An adversary cannot obtain any previous session keys even though the current session key is compromised. A timestamp based password authentication scheme that allows users to freely update their passwords keeps the merit that the remote server does not require to store passwords or verification table. Hence, this scheme suffers from online guessing attacks.

Chi-Chao Chang and Tzonelih Hwang (2006) proposed Modular Design for Round-Oriented Password Authentication Protocol. It seeks to identify the functional modules in password authentication schemes and give a general procedure for generating protocols with these modules. It also gives a proof of security for the generalized protocol produced from the procedure. With modular and round-oriented design, it is shown that flexible infrastructure can be built to provide sound solutions to password authentication in a wide range of hardware/software implementations and computing capabilities. Password schemes are simple because they are easy to
implement, and no additional hardware or software components are required other than standard input devices and security algorithms. There is also no need for users to carry physical tokens such as smart cards to keep their secret information. Besides, since passwords are memorized, these schemes are safe from loss, theft and damage. However, password authentication protocols do have their shortcomings.

Memorable passwords are often easy to guess because they either have some relationships to personal information or are susceptible to dictionary attacks. A good password authentication scheme should be resistant to these attacks or at least detects them within the first few attempts. The analysis of protocols has also made a great deal of progress from analyzing against specific attacks to the formal proofs of their security. However, these attempts to provide the single most efficient protocol face challenges not from their security, but from their practicality and flexibility. Each environment is unique in its hardware/software supports, corporate policies, user preferences and authoritative hierarchies.

There is simply no single solution to satisfy all password authentication requirements. An alternative approach of identifying major components in password authentication schemes which can be reorganized into round-oriented protocols and give a general procedure to create secure protocols without sacrificing communicating efficiency is proposed. System designers, using the technique provided here should be able to create secure password authentication protocols meeting their specific hardware/software restrictions. Protocols generated according to the procedures are consistent in terms of their security and round-efficiency. This protocol is heuristically secure because, the relationships between modules have very little impact on
the integrity of individual modules, which is seen in a variety of existing protocols and are considered to be secure.

Hung-Min Sun and Her-Tyan Yeh (2006) proposed a Password-based authentication and key distribution protocols with perfect forward secrecy. This approach focuses on an environment in which the users can use easy-to-remember passwords. In addition to password guessing attacks, perfect forward secrecy (PFS in short) is another important security consideration when designing an authentication and key distribution protocol. Based on the capability of protecting the client’s password, the application server’s secret key, and the authentication server’s private key, seven classes of perfect forward secrecy and focus on protocols achieving class-1, class-3, and class-7 are defined due to their hierarchical relations. Then, three secure authentication and key distribution protocols to provide perfect forward secrecy of these three classes were proposed. All the protocols are efficient in protecting poorly-chosen passwords chosen by users from guessing attacks and replay attacks.

Password-based mechanism has been the most widely used method for user authentication since, it allows people to choose and remember their own passwords without any assistant device. However, human users usually choose easy-to-remember passwords so that they are vulnerable to password guessing attacks. On the contrary, the entities excluding human users, such as servers, can directly use strong cryptographic secrets for entity authentication and hence prevent password guessing attacks. Three protocols that can fit this environment, resist various attacks, and separately achieve these three classes of PFS are proposed.
Her-Tyan Yeh and Hung-Min Sun (2003) proposed Password-based user authentication and key distribution protocols for client–server applications. This protocol discusses an environment in which a user (client) requests service from an application server through an authentication server. Two secure and efficient authentication protocols (KTAP: key transfer authentication protocol and KAAP: key agreement authentication protocol) proposed fits to this environment. These two proposed protocols can be efficiently applied to various communication systems in distributed computing environments since they provide security, efficiency, and reliability. On the contrary, the entities excluding human users, such as servers, can directly use strong cryptographic secret for entity authentication and hence prevent password guessing attacks.

From the viewpoint of the session key creation, the protocols can be classified into two flavors: key transfer protocols and key agreement protocols. In a two-party (a server and a client) setting, key transfer protocols means that the session key is created by the server and securely transmitted to the client, and key agreement protocols means that the client and the server contribute information to derive the common session key. In a three-party (a server and two clients) setting, key transfer protocols mean that the session key is created by the server and securely transmitted to these two clients, and key agreement protocols mean that both clients contribute information to derive the common session key. Compared with key-transfer protocols, the latter (key agreement protocols) is fairer and more secure, while the former (key transfer protocols) is more suitable for those cases in which computational capability of the clients is weaker or the server wants to monitor the communication message. However, an easily memorized password is also vulnerable to password guessing attacks.
Mihir Bellare et al (2011) proposed an Identity-Based (Lossy) Trapdoor Functions and Applications. Identity based trapdoor functions provide an automatic way to realize, in the identity-based setting, many functionalities previously known only in the public-key setting. In particular the first deterministic and efficiently searchable IBE schemes and the first hedged IBE schemes, which achieve best possible security in the face of bad randomness. A trapdoor function $F$ specifies, for each public key, an injective, deterministic map that can be inverted gives an associated secret key (trapdoor). The most basic measure of security is one-wayness.

Trapdoor functions are the primitive that began public key cryptography. Public-key encryption was built from TDFs. (Via hardcore bits.) Lossy TDFs enabled the first DDH and lattice (LWE) based TDFs. It is striking that identity-based cryptography developed entirely differently. The first realizations of IBE directly used randomization and were neither underlain by, nor gave rise to, any IB-TDFs. This asymmetry between the public-key and identity-based worlds (TDFs in one but not the other) is inherent. The applications of lossiness in the public-key realm suggest that lossy IBTDFs will be such a tool also in the identity-based realm. The solutions obtained are not practical but, it shows that the security goals can be achieved in principle, which was not clear prior to the work.

Eike Kiltz, Payman Mohassel and Adam O'Neill (1996) proposed an Adaptive Trapdoor Functions and Chosen-Ciphertext Security. ATDFs remain one-way even when the adversary is given access to an inversion oracle. The main application is the black-box construction of chosen-ciphertext secure public-key encryption (CCA-secure PKE). Namely, a black-box construction of CCA-Secure PKE from ATDFs, as well as a construction of ATDFs from correlation-secure TDFs. The notion of one-way trapdoor
functions (OW-TDFs) has played a central role in the study of cryptographic protocols, in particular for semantically secure public-key encryption (PKE). However, it is well-known that semantic security alone is not sufficient in many applications, rather, encryption must be secure against active adversaries, say, one can inject packets into the network and observe decryption or actions taken based on them.

As a result, resistance to so called chosen ciphertext attacks (CCA) has become the gold standard for security of PKE. Surprisingly, it is found that adaptivity, a seemingly fundamental notion in the context of chosen ciphertext security, serves to weaken the assumptions on a TDF needed to imply black-box CCA-secure PKE, as well as to unify and clarify the schemes. It is observed that this result extends to rule out a black-box construction of the former from ATDFs as well, by using the same breaking oracle.

Chin-Chen Chang et al (2010) proposed Provably Secure High Entropy Three-Party Authenticated Key Exchange Scheme for Network Environments. To achieve requirements in peer-to-peer systems, the server must assist entities in establishing secure communications among them (3PAKE). The 3PAKE mechanisms seem to be more realistic since an entity must memorize only a single password that is shared with a trusted server rather than memorizing multiple passwords with other entities. Roughly, 3PAKE mechanisms can be categorized into two types, key transport protocols and key exchange protocols.

In key transport protocols, a trusted server controls both the generation and distribution of session keys for communication entities. However, this means that the privacy of communication with respect to a
server cannot be ensured, even if a server shares some secrets with the communication entities. Therefore, 3PAKE with key exchange is the better mechanism for use in network environments.

The 3PAKE mechanism is superior to 2PAKE in terms of reducing memorable secrets, but a potential problem still remains. The problem is that a legitimate, but malicious, client can impersonate other clients even if the system involves complete authentication to entities. Unfortunately, it is found that the scheme is still vulnerable to an impersonation-response attack. This is due to the fact that the identities of entities and the transcripts are not always guaranteed; a legitimate, but malicious, client can modify the transcripts to get the response from a trusted server and then access other clients’ secrets. Furthermore, by applying a symmetric key cryptosystem, this scheme always encrypts authentication messages to provide secrecy protection; however, a symmetrical cryptosystem is usually susceptible to its troublesome key distribution. Once the adversary has compromised the password of the user, it is still harmful to the user since the adversary is computationally infeasible to derive the items that are used to be authenticated.

Yuh-Min Tseng (2006) proposed an Efficient Two-Party Identity-Based Key Exchange Protocol. The key exchange (or agreement) protocol is designed to allow two entities establishing a session key to encrypt the communication data over an open network. This approach presents an efficient identity-based key exchange protocol based on the difficulty of computing a discrete logarithm problem. As compared with the previously proposed protocols, it has better performance in terms of the computational cost and the communication steps. The proposed key exchange protocol provides implicit key authentication as well as the desired security attributes of an authenticated key exchange protocol.
A two-party key exchange (or agreement) protocol is used to establish a common session key for two specified entities, in which both two entities contribute some information to derive the shared session key. If three or more participants want to communicate securely over an insecure network, they may employ a conference-key establishment protocol to compute a conference key. However, it does not allow two entities to authenticate each other, so the protocol requires an authentication channel to exchange the public keys. A public-key based key exchange protocol adopts public-key cryptographic techniques to achieve the purposes of user authentication and key exchange. On the way of key management, although the public-key-based key exchange protocol is better than password-based key exchange protocol.

However, on-line access to get and verify public keys from a public key system in a network system is time-consuming. Moreover, it needs to require extra efforts to maintain public-keys in a public key system. On the other hand, an identity-based key exchange protocol can be regard as a variation of the public-key based key exchange protocol. An identity-based key exchange protocol is a protocol that uses user’s identity or some other information combined with its identity as one’s public key to achieve user authentication and key exchange. Thus, a verifier does not verify the certificates of the public keys.

Ting-Yi Chang et al (2010) proposed a Communication-Efficient Three-Party Password Authenticated Key Exchange protocol. Three-party password authenticated key exchange (3PAKE) protocols allow two users (clients) to establish a session key through an authentication server over an insecure channel. Clients only share an easy-to-remember password with the trusted server. Most schemes employ the server public keys to ensure the identities of both the servers and symmetric cryptosystems to encrypt the
messages. The 3PAKE requires neither the server public keys nor symmetric cryptosystems such as DES.

The formal proof of security of the 3PAKE is based on the computational Diffie-Hellman assumption in the random oracle model along with a parallel version of the proposed 3PAKE. The comparison shows that the 3PAKE is more practical than other 3PAKEs. The user should obtain server's public key via some authentication service such as the X.509 directory authentication service. This results in problems associated with public key infrastructures (such as revocation, centralized trust, key management issues, etc.), and new servers are allowed to join the network at any time during the execution of the protocol without requiring access to an on-line, trusted authority. In other words, a public key infrastructure is often considered as a drawback.

In view of the above disadvantages, some people have attempted to propose a 3PAKE without server public keys. New scheme with only a symmetric cryptosystem such as DES, Rijndael, etc., a pseudo-random function, and two different one-way hash functions to remove the server public keys in 3PAKE was proposed. Although CC-3PAKE has fewer rounds than in LSSH-3PAKE, the server needs to secretly store the trapdoor of the one-way trapdoor function. In other words, the server stores the trapdoor of the one-way trapdoor function as well as the password table. If one of them is lost, the scheme is broken. If the server needs to store any more secrets, then it would have a higher probability of breaking the scheme.

Li Gong et al (1993) proposed an approach on Protecting Poorly Chosen Secrets from Guessing Attacks. In a security system that allows people to choose passwords, those people tend to choose passwords that can
be easily guessed. This weakness exists in practically all widely used systems. Instead of forcing users to choose well-chosen secrets, which are likely to be difficult to remember, a solution is proposed that maintain both user convenience and a high level of security at the same time.

The basic idea is to ensure that data available to the attacker is sufficiently unpredictable to prevent an off-line verification of whether a guess is successful or not. Common forms of guessing attacks are examined by developing examples of cryptographic protocols that are immune to such attacks, and suggest a systematic way to examine protocols to detect vulnerabilities to such attacks.

These poorly chosen passwords are vulnerable to attacks based upon copying information. For example, the result of applying a one-way hash functions to a password or of encrypting a message using the password as the encryption key and experimenting off-line. Such secret guessing attacks are often associated with stored password systems such as that in the UNIX operating system. Often overlooked is the possibility of applying such attacks to messages passed over networks, in particular such attacks are likely to succeed if, for reasons of user acceptability an encryption key is derived algorithmically from a user-chosen password.

The term poorly chosen is used to describe an encryption key derived from a user chosen password. The term well-chosen is used to describe an encryption key chosen at random from a large key space. The distinction is based on the presumably low probability of an attacker successfully guessing a randomly chosen key compared with the presumably higher probability of successfully guessing a user chosen password. A common, but unacceptable counter to this risk is to encourage users to choose
passwords that are obscure. Some systems choose passwords for their users, but because these passwords are likely to be unmemorable, this method is inconvenient to users. Likewise inconvenient to users are long passwords.

Cheng-Chi Lee and Ya-Fen Chang (2008) proposed a Practical Three-Party Key Exchange Protocol With Round Efficiency. It allows two parties A and B to share an easy-to-remember password with a trusted server S. S acts as a coordinator between two communication parties to complete mutual authentication. Once authentication is achieved, two parties can share a session key to encrypt and decrypt their communication. Password guessing attacks can be divided into three classes: (1) detectable on-line password guessing attacks, (2) undetectable on-line password guessing attacks, and (3) off-line password guessing attacks. Among the three classes, off-line password guessing attacks are the most critical ones. C-3PEKE not only meets all above requirements, but also withstands the three attacks.

As a result, C-3PEKE is a practical three-party key exchange protocol with round efficiency. It is superior to others, but, 3PEKE is not robust enough against off-line password guessing attacks from E. E can intercept transmitted messages from public channel and then break password by playing off-line guessing attacks. E can guess a password P0 until the guessing password P0 is equal to the correct password P. Otherwise, E repeatedly guesses a new P0 off-line. Suppose that E tends to get A’s password PA. Therefore, C-3PEKE suffers from offline password guessing attacks.
Bellare et al (1998) proposed Many-to-one Trapdoor Functions and their Relation to Public-key Cryptosystems. The heart of the task of building public key cryptosystems is viewed as that of making trapdoors infact, public key cryptosystems and trapdoor functions are often discussed as synonymous.

It shows that non-injective trapdoor functions (with super-polynomial pre-image size) can be constructed from any one-way function (and hence it is unlikely that they suffice for public key encryption). On the other hand, it shows that trapdoor functions with polynomial pre-image size are sufficient for public key encryption. Together, these two results indicate that the pre-image size is a fundamental parameter of trapdoor functions. Turning the attention to the converse, asking what kinds of trapdoor functions can be constructed from public key cryptosystems.

The first step is to show that in the random-oracle model one can construct injective trapdoor functions from any public key cryptosystem. Among non-injective trapdoor functions, further distinction based on the amount of non-injectivity, measured by pre-image size is made. A (trapdoor, one-way) function is said to have pre-image size $Q(k)$ (where $k$ is the security parameter) if the number of pre-images of any range point is at most $Q(k)$. It is shown that pre-image size is a crucial parameter with regard to building public-key cryptosystems out of a trapdoor function. Since unapproximable trapdoor predicates and semantically secure public key cryptosystems are equivalent. Infact whether or not semantically secure public key cryptosystems imply injective trapdoor functions seems a hard one.

Julien Cathalo and Christophe Petit proposed a One-Time Trapdoor One-Way Functions. Trapdoors are widely used in cryptography, in particular
for digital signatures and public key encryption. In these classical applications, it is highly desirable that trapdoors remain secret even after their use.

Positive applications of trapdoors that do not remain secret when they are used. One-time trapdoor one-way functions (OTTOWF) are introduced and formally defined, a primitive similar in spirit to classical trapdoor one-way functions, with the additional property that its trapdoor always becomes public after use.

Three constructions of OTTOWF are provided. Two of them are based on factoring assumptions and the third one on generic one-way functions. Then the potential applications of the primitive and in particular the fair exchange problem is considered.

Two fair exchange protocols were provided using OTTOWF, where the trapdoor is used to provide some advantage to one of the parties, whereas any (abusive) use of this trapdoor will make the advantage available to the other party as well. Comparing the protocols with well-established solutions for fair exchange and describe some scenarios where they have advantageous characteristics. If trapdoors provide their owners with some specific advantage, their leakage will, on the other hand, strongly limit this power. Fresh parameters are to be generated after each use, which is an important limitation in many applications.