Integrated Key Exchange Protocol Capable of Revealing Spoofing and Resisting Dictionary Attacks

David Lai and Zhongwei Zhang

Department of Mathematics and Computing, University of Southern Queensland,
Toowoomba, Queensland, 4350, Australia.
Email: (lai, zhongwei)@usq.edu.au

Abstract. In this paper we propose a new verifier-based password authentication protocol using Dynamic Passwords. The protocol features mutual authentication, integrated session key exchange and is resistant to both dictionary attacks and replay attacks. It also reveals any successful spoofing. The protocol achieves these features by using one-way hash functions, symmetric encryption and two-part Dynamic Passwords. Dynamic Passwords break user passwords into two parts: a dynamic part and a static part. The dynamic part is similar to a one-time password, which can reveal to a legitimate user if any spoofing has occurred and makes the protocol more resistant to social engineering. The static part adds entropy to the password and makes the password more resistant to dictionary attacks. Due to the fact that verifiers are not plain-text equivalent to passwords and from which no meaningful information about passwords can be extracted, verifiers in contrast to passwords are stored in authentication servers. The protocol is most suitable for mobile users because no persistent data is stored on the user side.

Keywords: Integrated Key Exchange, Dynamic Passwords, Kerberos, Dictionary Attack, Replay Attack, Revealing Spoofing.

1 Introduction

Among the many authentication protocols available, password systems are the most common and easiest forms of authentication. Other forms of authentication may make use of something that a user owns or is part of the user. In many cases, dedicated hardware may be required. An authentication protocol that requires dedicated hardware is less flexible, and may have compatibility and portability problems when implemented across different platforms. Using passwords, users only have to remember their own account name and password in order to access protected resources. Password authentication is easy to use and straight forward to implement. However it has some limitations.

Strong passwords are comparatively long. Memorizing a long password made up of unrelated characters and digits can be a daunting task. This encourages people to choose short passwords that are easy to guess such as birthdays, names of kin or common English words, and to use the passwords for as long as possible. These weak passwords are the targets of dictionary attack, which is a form of brute force attack. In a dictionary attack, possible passwords from lists of common dates, names and dictionary words are used. The results are validated using some publicly known information about the password and the authentication protocol used.

A second problem is the replay attack. Replay attack occurs when an adversary captures a legitimate login packet and replays it later to gain access to resources as a legal user.

Another problem is spoofing. If a password is stolen, an adversary can login as the owner of the password. The owner of the stolen password is not alerted to the illegal login.

With all these problems in mind, we propose the use of Dynamic Passwords. A Dynamic Password consists of two parts. The first part changes each time when the Dynamic Password is used and is called the dynamic part. The second part has a relatively longer life time and length than the first part and is called the static part. If a Dynamic Password is changed with every use, replay attack is not possible. At the same time, spoofing attack can easily be detected.

To deliver the new dynamic part of a Dynamic Password, we propose Integrated Key Exchange Using Dynamic Passwords (IKE) [1]. This protocol is portable for mobile users, resists replay and dictionary attacks, provides mutual authentication and integrates key exchange with authentication. IKE also offers a detection mechanism when spoofing happens

In this paper, we will introduce the Dynamic Password in Section 2 and a description of Integrated Key Exchange starts in Section 3. We will discuss the possible application of IKE in Section 5.

2 Dynamic Password

Length, composition and life time [7] are three important password design factors. A password can be made stronger by increasing its length, decreasing its lifetime and enlarging the character set from which characters are drawn to form the password. Off-line dictionary attacks are possible if the authentication messages contain publicly known information that can be used to validate a password guess. If no such information is available, adversaries can still launch on-line brute force attacks.

In [7], there is a guideline for determining the length and lifetime of passwords that are safe against on-line brute force attacks. Assuming a password guess consists of sending 200 bits and using a modem with speed of 230.4 kbps, a password of 5 characters drawn from a 62-character set can have a lifetime of one second only. A password of 8 characters long can have a lifetime of one day as shown in Table 1. Memorizing a long password consisting of unrelated characters is not easy, and is not practical when the password changes frequently. Users more readily accept longer passwords when the passwords have a longer lifetime.

Dynamic Password breaks a normal password into two parts. The Static Part (SP) has a longer lifetime and also a longer length. The Dynamic Part (DP) has a shorter lifetime and shorter length. SP and DP together satisfy the length and lifetime requirements of a strong password. A longer lifetime alleviates the effect of the longer length of SP. The short lifetime of DP is mitigated by its relatively shorter length.

A practical implementation is to use dictionary words as the character set of SP. It makes SP, and in turn Dynamic Password more user-friendly. If we only use words of 4, 5 or 6 letters, [[4-6]letter word], the character set for SP consists of approximately 23,300 characters. A SP of length 2 means the SP is made up of two 4, 5 or 6 letter words. Again assuming a password guess consists of sending 200 bits and using a modem with speed of 230.4 kbps, Table 1 lists

[1] As pointed out by a reviewer, the name Integrated Key Exchange has been used by IPV_6. We might select a different in the future.

the combination of lengths of SP and DP for different lifetimes. The last column gives the length of ordinary passwords with the same lifetime. The character set used in the table for DP is [A-Za-z0-9].

Table 1. Lengths of SP and DP for different lifetimes.

Lifetime	Length of SP	+	Length of DP	Normal Password
1 day	2	+	3	8
1 day	3	+	1	8
1 month	2	+	4	9
1 month	3	+	2	9
1 year	2	+	5	10
1 year	3	+	2	10

With every new DP, a new Dynamic Password is formed even though SP remains the same. If the DP is assigned a lifetime until next login, the Dynamic Password formed is effectively a one-time password [3].

If we assume users login every day, a one-time Dynamic Password will have a lifetime of 1 day. From Table 1 a normal password with a lifetime of 1 day is 8 characters long. In comparison, a Dynamic Password of lifetime 1 day has a DP of 1 character long and SP of three dictionary words. As the SP can remain the same for an extended period of time, users can remember the static SP after a few login sessions. The daily task of memorizing 8 unrelated digits and alphabets is reduced to memorizing 1 only. Both SP and DP can be lengthened to enhance the strength of Dynamic Passwords formed.

Replay attack is not possible with Dynamic Passwords. As the DP of a Dynamic Password will be changed with every use of the password, a replay of login request will definitely fail.

One-time Dynamic Passwords can also reveal spoofing. If an adversary impersonates a legitimate user and gains access to an account, DP will be changed for the next login. Legitimate users are alerted if spoofing has occurred since they cannot login using the correct but expired DP.

Both SP and DP when used alone are weak passwords. When SP and DP are used together as a Dynamic Password, they form a stronger password that is easy to remember.

In the discussion below, DP and SP have character sets of [A-Za-z0-9] and [[4-6]letter word] respectively. The transmission of the new DP and establishment of session keys will be discussed in Section 3.

3 Integrated Key Exchange Using Dynamic Passwords

When we use Dynamic Passwords we need a protocol that authenticates a user and transmits the new DP of a Dynamic Password after each login. We will take a look at the existing key exchange schemes in Sections 3.1 to 3.2 and present Integrated Key Exchange Using Dynamic Passwords (IKE) in Section 3.3.

3.1 Encryption Key Exchange

The classic protocol Encryption Key Exchange (*EKE*) by Bellovin and Merritt [1] and its successor Augmented-EKE (*A–EKE*) [2] initiated a collection of authentication protocols that apply encryption for key exchanges. *EKE* uses a shared secret indirectly to authenticate servers and users.

In *EKE* the shared secret is the password. When Alice authenticates herself to Bob, a session key R between Alice and Bob is also established. *EKE* is secure against dictionary attacks as the contents of the messages are generated randomly. It does not provide any clue for guessing the password. However *EKE* is susceptible to the Denning-Sacco attack [8] in which a stolen session key can be used to launch a replay attack on the password. A method to strengthen *EKE* against Denning-Sacco Attacks was proposed. A more efficient Minimal *EKE* (*M–EKE*) was also proposed [8].

During *EKE* message exchanges, passwords are not sent across the communication channel, but are used as a symmetric encryption key. Both parties must have access to the shared password. If the server is compromised and passwords are stolen, an adversary can impersonate a legitimate user. In view of this, Augmented Encryption Key Exchange (*A–EKE*) was proposed to fix this problem. In *A–EKE*, a one-way hash value of the password is used as the shared secret. The symmetric encryption key is the one-way hashed value of the password instead of the password itself. However, a stolen session key can be used to launch a dictionary attack on the password in *A–EKE*.

Other alternatives to *A–EKE* are proposed. Strong Password-Only Authenticated Key Exchange ($SPEKE$) [4] uses the hash of the password as the base for exponentiation instead of a fixed primitive base. Extended Password Key Exchange Protocols [5] are a family of protocols that extend the *A–EKE* and $SPEKE$ protocols to $B-EKE$ and $B-SPEKE$ by using a second Diffie-Hellman exchange instead of a digital signature to prove that a user has the knowledge of a certain password.

Open Key Exchange (OKE) [6] eliminates the use of encryption for the user's public key. Another feature of OKE is that the user's public key can be reused as long as the private key is kept secret.

3.2 Asymmetric Key Exchange

Based on the swapped-secret approach, Asymmetric Key Exchange (AKE) [9] is another class of protocol that exchanges keys without using encryption. Initially, each party computes a secret and generates a verifier with a one-way hash function. They swap their secrets and use them as long term swapped secrets. For each session, each party generates another session secret and swaps with the other party. The session key is generated from the swapped long term and session secrets. Authentication is completed when both parties confirm that they have the same session key.

AKE is similar to the Diffie-Hellman key exchange approach. Session keys are formed based on the swapped secrets. The difference is that AKE combines some shared secrets in forming the session key. The shared secrets make AKE free from Man-in-the-Middle Attacks. No encryption is used in AKE and it takes only four steps. It is not suitable for mobile users, as both parties must store the verifier of the other party.

3.3 Integrated Key Exchange Protocol

Existing key exchange schemes do not offer transmission of new passwords. The DP of a one-time Dynamic Password will be renewed with every login. Integrated Key Exchange Using Dynamic Passwords is a password authentication protocol which authenticates users, delivers new DP to users and establishes session keys.

Integrate Key Exchange Using Dynamic Password (IKE) consists of two phases. In the Registration phase, users get their account names, SP and initial DP. The login phase is a typical user login session. We will assume each user can only have a single login session. The server will reject multiple login sessions for one user. In the discussion that follows, a server running IKE is called $\mathbb{IKES}$. An application on the user side running IKE is called $\mathbb{IKEC}$. The user uses $\mathbb{IKEC}$ to login to $\mathbb{IKES}$. The abbreviations used are listed in Table 2.

Table 2. List of abbreviations for IKE.

Abbr.	Meaning
DP	Dynamic Part of a Dynamic Password
DP_n	New DP selected by Server
SP	Static Part of a Dynamic Password
$H(X)$	One way hash value of X
R_n	The n^{th} random number
M_n	The n^{th} message
SK	Session Key
$\{X\}_Y$	Symmetric encryption of X using key Y
$\{X\}^Y$	Symmetric decryption of X using key Y
$RandomGen()$	Random string generator
$SPPasswdMap(X)$	Maps X from random string space to SP password space with character set [[4-6]letter word]
$DPPasswdMap(X)$	Maps X from random string space to DP password space with character set [A-Za-z0-9]

Phase 1: Registration

The registration process is performed at $\mathbb{IKES}$. The user first selects a unique account name. $\mathbb{IKES}$ then generates two random numbers and maps them to SP and DP password spaces to form SP and DP respectively. The SP and DP are then transferred to the user via a secure channel. $\mathbb{IKES}$ uses some one-way function to form hash values of SP and DP. The $H(SP)$ and $H(DP)$ are stored in $\mathbb{IKES}$. SP and DP are erased from $\mathbb{IKES}$. The process of registration is shown in Algorithm 1.

Algorithm 1: Registration Phase of Integrated Key Exchange Using Dynamic Passwords

Begin (Server Side)
$AccountName \longleftarrow UserInput$
$SP \longleftarrow SPPasswdMap(RandomGen())$
$DP \longleftarrow DPPasswdMap(RandomGen())$
$SecureFileStorage \longleftarrow H(SP), H(DP)$
$User \longleftarrow DP, SP$
end

After registration, the server will have $H(SP)$ and $H(DP)$. The user will have to remember SP and DP. DP will be changed with every successful login.

Phase 2: Authentication

An honest execution of the protocol is shown in Algorithm 2.[2]

Algorithm 2: An honest execution of Integrated Key Exchange Using Dynamic Passwords

---

Begin (Process 1, Client Side)
$DP, SP \longleftarrow UserInput$
$R_1 \longleftarrow RandomGen()$
$M_1 \longleftarrow \{R_1\}_{H(SP)}$
end

$M_1(fromClient) \longrightarrow Server$

Begin (Process 2, Server Side)
$H(DP), H(SP) \longleftarrow SecureFileStorage$
$R_1 \longleftarrow \{M_1\}^{H(SP)}$
$R_2 \longleftarrow RandomGen()$
$DP_n \longleftarrow DPPasswdMap(RandomGen())$
$M_2 \longleftarrow \{R_1, DP_n\}_{R_1}$
end

$Client \longleftarrow M_2(fromServer)$

Begin (Process 3, Client Side)
$R_1, DP_n \longleftarrow \{M_2\}^{R_1}$
$R_3 \longleftarrow RandomGen()$
$M_3 \longleftarrow \{R_3, H(DP)\}_{R_2}$
$SK \longleftarrow H(R_1, R_2, R_3)$
$DP \longleftarrow DP_n$ Upon successful use of SK
end

$M_3(fromClient) \longrightarrow Server$

Begin (Process 4, Server Side)
$R_2 \longleftarrow StoredMemory$
$H(DP) \longleftarrow \{M_3\}^{R_2}$
Compare $H(DP)$ with stored value
$SK \longleftarrow H(R_1, R_2, R_3)$
$H(DP) \longleftarrow H(DP_n)$ Upon successful use of SK
end

---

[2] Algorithm 2 has been refined to address the vulnerability of dictionary attacks against the combined password which concerned the reviewer.

Process 1: To start a login session, the user inputs the account name, SP and DP to $IKEC$. $IKEC$ generates a random string R_1 and forms message M_1:

$$M_1 = \{R_1\}_{H(SP)} \tag{1}$$

$IKEC$ deletes SP, DP and $H(SP)$ because they are not needed by $IKEC$ any more. However $IKEC$ holds R_1 and $H(DP)$ until the session key is established. M_1 and account name are sent to $IKES$.

Process 2: When $IKES$ receives M_1 and the account name from $IKEC$, $IKES$ retrieves the corresponding $H(SP)$. $IKES$ decrypts M_1 and gets R_1. $IKES$ generates a new DP_n, a random string R_2 and forms message M_2 which is sent to $IKEC$.

$$M_2 = \{R_2, DP_n\}_{R_1} \tag{2}$$

Process 3: Upon receiving M_2, $IKEC$ decrypts M_2 using the stored value of R_1 and extracts the random string R_2. $IKEC$ generates a random string R_3 and forms message M_3. M_3 is sent to $IKES$.

$$M_3 = \{R_3, DP\}_{R_2} \tag{3}$$

$$SK = H(R_1, R_2, R_3) \tag{4}$$

Upon successful use of SK, $IKEC$ will display DP_n to the user. The user must use this new DP_n in place of the old DP If $IKEC$ does not receive a valid M_2 after a predefined period of time, $IKEC$ may assume that $IKES$ has not received the login message M_1. So $IKEC$ will delete all stored information about the user and prompt the user for a new login.

Process 4: After receiving M_3, $IKES$ decrypts M_3 and extracts DP and R_3. DP is hashed and compared with the corresponding stored hash values. If they match, $IKES$ will form SK as in equation (4).

Upon successful use of SK, $IKES$ will replace the stored $H(DP)$ with $H(DP_n)$. If $IKES$ does not receive a valid M_3 after a predefined period of time, $IKES$ may assume that $IKEC$ has not received the message M_2. So $IKEC$ will delete all stored information about the user and prompt the user for a new login. $IKEC$ cannot communicate with $IKES$ using SK and so will not display DP_n to the user as the new DP.

4 Analysis of Integrated Key Exchange

Mutual Authentication IKE provides explicit user authentication and implicit server authentication. If the user does not provide the proper DP in M_3, user authentication fails. If the user cannot communicate with the server using SK, it means that one or more of the random strings R_1, R_2 and R_3 do not have the same corresponding values in the server. It implies that the server has not use the proper $H(SP)$ to decrypt M_1. So server authentication fails.

Dictionary Attacks If the messages exchanged in the authentication process contain publicly known information that can be used to validate a password guess, dictionary attacks are possible. In IKE, the contents of messages M_1, M_2 and M_3 are listed in equations (1, 2, 3).

As the character set of DP and DP_n is [A-Za-z0-9], R_1, R_2 and R_3 are randomly generated, so the content of M_1, M_2 and M_3 do not provide any extra information for the adversary to validate their password guesses.

Replay Attacks After each successful login, $IKES$ will store a new $H(DP_n)$ in its secret files. The user will be prompted to remember the new DP_n displayed on the screen for next login. If an adversary replays any previous login messages, the DP used in M_3 cannot form the same hash value as the $H(DP)$ stored in $IKES$. User authentication will fail. This will prevent adversaries from launching replay attacks.

Brute Force Attack In IKE, the Dynamic Password used must satisfy the length and lifetime for resisting on-line brute force attacks. Increasing the length of SP and DP will increase the strength of Dynamic Passwords formed. This will reduce the chance of success for on-line brute force attacks.

Table 1 shows that changing the DP from 1 character to 2 characters will change the lifetime requirements of the Dynamic Password from 1 day to 1 year. In fact increasing the length of the DP to 3 will give a lifetime of more than 80 years under the same conditions.

Spoofing The easiest way to compromise passwords is by looking over the shoulder of users when they enter their passwords. IKE cannot stop adversaries from spying on user passwords. But IKE makes such an attack harder with DP in the Dynamic Password. An adversary has to look over the user's shoulder while the user enters the SP and also when the new DP is displayed by $IKEC$. The spying action is thus made more conspicuous and suspicious. The chance that an adversary can get both the SP and the new DP without alerting the user is much less than with simple password entry.

Revealing Spoofing If an adversary has knowledge of a valid account name and password, he can login to the account and just read information without changing anything. It is hard for a user to discover that an adversary has gained access to his account.

As the DP of the user changes for every successful login, the DP will be changed by the spoofed login. The legitimate user still assumes and uses the expired DP to initiate a login session after spoofing. Obviously the user cannot login. At this point, the user must contact the server administrator and re-initialize his SP and DP. He can then check with the server administrator as to the time of his last successful login. The last login time may reveal whether spoofing has occurred or not.

5 Application of IKE

IKE is highly portable compared to authentication schemes that use passwords. To apply IKE, all we have to do is to determine the lengths of the SP and the DP, the one-way hash algorithm and the symmetric encryption algorithm to be used in $IKE4$.

One common one-way hash algorithm is the MD5 algorithm which produces 128 bit hash values. SHA1 produces 160 bit hash values. Triple DES may be used as the symmetric encryption algorithm. SP can be 4 random words from a set of 4-6 letter dictionary words. DP can be 4 or more random characters from the character set [A-Za-z0-9].

Kerberos is a network authentication system that uses passwords. After authentication, a user has to get tickets for different services. In Section 5.2 we will describe how IKE can be used in Kerberos to perform user password authentication.

5.1 Distributed Computer Networks Systems

We can apply IKE in its simplest form without any adaptation. In a typical client-server situation, IKE can provide the necessary mutual authentication between client and server.

In a distributed computer networks system, a user can login to an authentication server and obtain session keys or service tokens to be used with other servers within the system. Using IKE, the traffic between the authentication server and the user are protected by symmetric encryptions that have different keys for every message. The random nature of the authentication message content makes the authentication process immune to dictionary attacks. The ability of IKE to reveal spoofing makes the user more confident about personal privacy and system security within the distributed computer networks system.

5.2 *IKE* ported to Kerberos

We can also port IKE to other authentication schemes using passwords. Kerberos is an authentication protocol for accessing distributed resources across the network. Kerberos authenticates and grants tickets to users. Tickets enable users to use different services within a certain period of time without any further need to authenticate themselves. The only major drawback is that it is vulnerable to dictionary attack. IKE is practically immune from dictionary attack and complements the password authentication part of Kerberos. Porting IKE to Kerberos involves some minimal changes to the message exchange steps in Kerberos.

The abbreviations used in the Kerberos message exchange are listed in Table 3. During

Table 3. List of abbreviations used in Kerberos message exchange.

Abbr.	Meaning
C	Account User Name
T	Ticket Granting Server name
KDC	Key Distribution Center
K_X	Secret key of entity X
K_{XY}	Session key between entities X and Y
$\{M\}_K$	Message M encrypted with key K

Kerberos message exchange, an adversary can easily pretend to be a legitimate user and sends KDC an initial plain request. KDC just returns message 2 to the adversary without any authentication. The session key between the TGS and the client is encrypted with the secret key of the client, which is derived from the user password using some known function. The adversary can then start dictionary attacks or brute-force attacks on the encrypted session key and validate his guesses against some known format data encrypted along with the session key.

IKE can be applied to Kerberos. The share secret changes from a secret symmetric key to hashes HSP and $HHDP$. Message M_1 and message M_2 need to be changed with an added step between message M_2 and message M_3. The rest remains the same.

$$M_1 = C, T \quad \text{Plain request from client to server changes to} \tag{5}$$

$$M_1 = \{R_3, H(DP)\}_{H(SP)} \tag{6}$$

$$M_2 = \{K_{CT}\}_{K_C} \quad \text{The } KDC \text{ encrypted reply message changes to} \tag{7}$$

$$M_2 = \{K_{CT}, DP_n\}_{R_1} \tag{8}$$

An added message M_{2a} is sent from the user to KDC:

$$M_{2a} = \{R_3, H(DP)\}_{H(SP)} \tag{9}$$

6 Conclusion

Integrated Key Exchange Using Dynamic Password is a password authentication scheme, which allows us to have the benefit of strong passwords with short lifetime, yet with easy to remember passwords. The authentication messages are immune to replay and dictionary attacks. IKE is also portable and reveals spoofing. IKE is also resistant to social engineering. The strength of the protocol relies on the strength of the encryption algorithm and the hashing algorithm used. Implementation of IKE is simple, easy and cheap. It can easily be adapted for use in Kerberos and is most suitable for Distributed Networks Systems. To improve and provide a formal security argument for IKE will be the focus of our future work.

7 Acknowledgments

We would like to thank Peter Cattell for proof reading this paper.

References

1. Steven M. Bellovin and Michael Merritt: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California. (May 1992) 72-84
2. Steven M. Bellovin and Michael Merritt: Augmented Encrypted Key Exchange: A Password-Based Protocol Secure against Dictionary Attacks and Password File Compromise. ACM Conference on Computer and Communications Security. (1993) 244-250
3. N. Haller and C. Metz and P. Nezzerr and M. Straw: A One-Time Password System. RFC2889 (Feb 1998)
4. David P. Jablon: Strong Password-Only Authenticated Key Exchange. Computer Communication Review. **26(5)** (1996) 5–26
5. David P. Jablon: Extended Password Key Exchange Protocols Immune to Dictionary Attack. Proceedings of the WETICE'97 Workshop on Enterprise Security, Cambridge, MA, USA. (1997)
6. Stefan Lucks: Open Key Exchange: How to Defeat Dictionary Attacks without Encrypting Public Keys. Ecole Normale Suprieure, Paris. (April 1997) 79–90
7. National Institute of Standards and Technology: Password Usage. Federal Information Processing Standards Publication. **112** (May 1985)
8. Michael Steiner and Gene Tsudik and Michael Waidner: Refinement and Extension of Encrypted Key Exchange. Operating Systems Review. **29(3)** 22-30
9. Thomas Wu: The Secure Remote Password Protocol. Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium, San Diego, CA. (1998) 97–111