1.

Design and Implement of Key Management System for Public Key Algorithms

تصميم وتنفيذ نظام ادارة مفاتيح لخوارزميات المفتاح العام

Mohammed Gheni Alwan
محمد غني علوان
Department of Computer Science
University of Technology
Mgaz_mgaz @yahoo.com

Diyar K. Mohammed Saed
ديار خيري محمد سعيد
Department of Computer
University of Duhok
Spiderdm@yahoo.com

Abstract

Design cryptographic system requires some properties like security, speed and other. One of the most important design criteria is the key management system it is considered as another part of the cryptographic system. Good cryptographic systems must have a strong key management systems responsible for generate the key, store and exchange it. Since the security of the system depends on the secrecy of the key not on the algorithm. It becomes important to study and design this part of cryptographic system. In this paper a new key management system was proposed. This system is capable of generating pair of private / public key, store, exchange and revoke it. The proposed system works between two parties and does not need the third parity.It can countermeasure man in the middle attack, replay attack and clogging attack. The proposed system is based on the PEM (Privacy Enhanced Email) since it's a simple electronic mail security service program but its key management process suffers from some difficulties which will be discussed in this paper. The proposed system can work with any security system requires key exchange process.

الخلاصة

تصميم انظمة التجفير يتطلب بعض الخصائص مثل الامنية، السرعة وغيرها. واحدة من اهم معايير التصميم هي نظام ادارة المفاتيح حيث انه يعتبر الجزء الاخر من نظام التجفير. نظام التجفير الجيد يجب ان يمتلك نظام ادارة مفاتيح قوي مسؤل عن توليد المفاتيح، خزنها وتبادلها. طالما ان امنية النظام تعتمد على امنية المفتاح وليس الخوارزمية . اصبح من المهم دراسة وتصميم هذا الجزء من نظام التجفير. في هذا البحث نظام ادارة مفاتيح قد تم اقتراحه . هذا النظام قادرعلى توليد زوج المفتاح السري/العام، خزنه، تبادله والغاءه. النظام المقترح يعمل بين طرفين ولا يحتاج الى طرف ثالث ويقاوم هجوم الرجل في المنتصف، هجوم التكرار، هجوم العرقلة. النظام المقترح صمم بالاعتماد على نظام سرية البريد الالكتروني طالما انه برنامج امني بسيط لحماية البريد الالكتروني ولكن له مشاكل في نظام ادارة المفاتيح سوف توضح في البحث . يمكن للنظام المقترح ان يعمل مع أي نظام امني لخدمة البريد الالكتروني.

1. Introduction

PEM (Privacy Enhanced Email) is a proposed standard that defines message encryption and authentication procedures in order to provide (PEM) services for electronic mail transfer on the Internet. It is most commonly used in conjunction with the Internet's Simple Mail Transfer Protocol (SMTP), but can be used with any electronic mail scheme (such as X.400) [1].

PEM is intended to be compatible with a wide range of key management approaches. It has mechanisms for using conventional (secret-key) cryptography or public-key cryptography for key management [1, 2] .However, all the readily available PEM implementations use public-key cryptography for key management.

Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties.

Key management encompasses techniques and procedures supporting initialization of system users within a domain, generation, distribution, and installation of keying material; Controlling the use of keying material; update, revocation, and destruction of keying material; and Storage, backup/recovery, and archival of keying material [3] .

The objective of key management is to maintain keying relationships and keying material in a manner which counters relevant threats, such as compromise of confidentiality of secret keys. Compromise of authenticity of secret or public keys. Authenticity requirements include knowledge or verifiability of the true identity of the party a key is shared or associated with it. Unauthorized use of secret or public keys.

Examples include using a Key which is no longer valid, or for other than an intended purpose. In practice, an additional objective is conformance to a relevant security Policy [3].

2. X.509 Public-Key Certificates

A public-key certificate is a specialized data structure used to securely bind a public key to a bunch of attributes. This identifies Alice as a person with a public key. Her certificate is her public key, her name, and some other information about her, all collected together and signed by a trusted person (a Certification Authority, or CA, in PEM lingo). Anybody else in the PEM universe can verify the CA's signature and be assured that the key belongs to Alice.

PEM uses certificates that conform to the X.509 standard [1,5]. These certificates bind a public key to a directory name (also known as a unique Internet address), and identify the issuer of the certificate. The following paragraphs explain the different parts of the certificate in detail [1, 5, 6].

- The "Version" field differentiates between successive versions of the certificate format. This field is always "0".
- The "Serial Number" field uniquely identifies this certificate among those issued by a particular issuer.
- The "Signature" field identifies the digital signature algorithm used to sign the certificate. Right now, two signature algorithms are defined: RSA with MD2, and RSA with MD5.
- The "Issuer" is the name of the person who signed the certificate. The "Subject" is the name of the person own the certificate . The "Validity" Field contains a pair of dates: the certificate is valid only after the first date and before the second.
- And finally, the subject's public key is in the "Subject Public Key Info" field.

The issuer signs this whole construct, using the signature algorithm specified in the "Signature" field and his own private key. Then he attaches his signature to the end of the construct. That's the certificate.

To validate the certificate, all a user has to do is to verify the issuer's signature, much in the same way that a receiver verifies the signature on an incoming message.

3. The PEM Public-Key Certification Framework

PEM's certification framework is a subset of the X.509 standard it is hierarchical, but with some modifications to deal with the real world.

PEM's certification framework accommodates a wide range of trust policies, but imposes constraints to facilitate uniformity across mail systems. In particular, be a small number of different PCAs, each with a substantially different policy.

The framework is based on the concept of a certification authority (CA).According to the X.509 documents, a CA is "an authority trusted by one or more users to create and assign certificates." In PEM, CA's are the people who sign certificates.

These CAs are organized in a single tree, at the top is the Internet PCA Registration Authority (IPRA); it will operate under the auspices of the Internet Society, a nonprofit, professional organization that promotes use of Internet technology around the world. The IPRA not only will provide that common reference point from which all certification chains stem, it also will set policy for PEM certifications [7].

The IPRA will issue certificates to a second tier of entities, called Policy Certification Authorities (PCAs). The IPRA will be responsible for registering the PCAs. Each PCA must file a document describing its policies with the IRPA. The IRPA will then sign and distribute that statement; any user will be able to get a signed copy of the policy statement of any PCA [1,7].

The IPRA will have other functions, as well. (Actually, this entity does not exist yet, but that all PCA and CA names are unique, and will manage all certificate revocation lists [7].

PEM has special mechanisms to handle certificate revocation. The PEM specification defines a Certificate Revocation List (CRL) to spread information about revoked certificates [1, 2]. A CRL is a special message, signed by its issuer. It consists of the issuer's name, the date the CRL is generated, and a sequence of pairs, each consisting of a certificate serial number and a date at which the certificate was revoked [2]. This CRL is spread far and wide across the net, so, when a receiver validates a certificate, he has to check that the certificate is not listed on a CRL. This is similar to a merchant checking a credit card against a "hot list" of bad numbers [1]

4. Proposed Key Management Subsystem

The proposed key management system is designed to work between two parties only and overcome the difficulties found in PEM X.509 Public-Key Certificates. Which can be summarized as follows:

1. Tree of certification authorities with only one IPRA as a root, this structure will cause a bottleneck problem on the IPRA and the person need to issue a certificate must follow all the path from the root to the leaves.

2. Its depends on the principle of certification authority which means there is a third party in certification.

3. Each party want to send a secure E- mail message to other party must first check CRL(certification revocation list) to ensure that the key will be used not revoked, this require that the sender contact the nearest certification authority [4]. .

In the proposed system no third party is required to manage the exchange process which is done in PEM or MIME by certification authority, rather than it's implemented by the proposed system using special message format. The proposed format is designed to be secure and countermeasure different types of attack in network like clogging attack , replay attack and man in the middle attack. The proposed system is designed to generate and manage the key exchange for RSA and MVECC public key algorithm. EL Gammal public key algorithm is selected to be used in key exchange process. Also SHA-1 is used to provide authentication and message integrity, radix64 encoding is used to encode the result message body. The proposed key management system consists of the following action:

1. Create public/private key pair Process.
2. Store key pair Process.
3. Public Key exchange Process.
4. Key revocations Process.

For services in 1 and 2 no message are required. For Service 3 and 4, messages are required to be used, where in case 3 the message is called KEYEXCHANGE and in case 4 the message is called REVOK.

Also a control message is called AKNWLEDGE introduced with these two messages.

This message carries different header fields according to message type (revoke, key exchange) but they are common in the text body.

Message text consists of two parts (code, acknowledge text), Code consists of three digit interpreted as follows:

- First digit: represents acknowledge type, possible value 1 and 2.
- Second digit: represents message type, possible value is (3 and 4).
- Third digit: represents algorithms identifier used, possible value

(1, 2 and 3). Table (1) gives a brief summary of these codes.

Table (1) Summary of acknowledge code

CODE	*FIRST DIGIT*	*SECOND DIGIT*	*THIRD DIGIT*
14-	First akn	REVOK	Depends on revoked key algorithm (1 for RSA and 2 for MVECC)
153	First akn	key exchange	EL Gamal alg
24-	Second akn	REVOK	Depends on revoked key algorithm(1 for RSA and 2 for MVECC)
253	second akn	key exchange	EL Gamal alg

Some symbol used:

R64: radix 64 encoding

H: SHA-1 to compute digest

E: encryption process

4-1 Create Public and Private Key Pair

This process is responsible for generating public/private key pair to system used (RSA and MVECC) algorithm. The user must submit public parameter needed to invoke each Process then generates public key submitted to the user.

4-2 Store Key Pair

This Process stores the received public key from the sender automatically after verifying the received message in the key Table and stores user generated public and private keys when the user chose to store them.

What's important to remember here is the mechanism proposed to protect the owner private key. An algorithm is proposed to encode private key on the key Table, so private key field in key Table will not contain a private key as plain text. When the private key is needed in any Process then the encoded private key is decoded to retrieve the original value.

Table (2) Private Key Encoding Scheme

Binary Value	Encode Value	Binary Value	Encode Value
000	0	0	8
001	1	1	9
010	2	00	88
011	3	01	89
100	4	10	98
101	5	11	99
110	6		
111	7		

The proposed encoding procedure works as follows:

1. Convert the private key to binary representation.
2. Encode each three bit into the corresponding value on left hand side of Table (2).
3. Use the right hand side of Table (2) to encode the remainder of the length of the binary representation of modular three.

4-3 Public Key Exchange Process

Exchanging public key is performed through a special proposed message type "KEYEXCHANGE" and its associated acknowledgements. This Process (i.e. key exchange) uses the one-to-one trust model or what's called direct trusting which mean any two parties can exchange public key at any time and at any place without the need for a third party . The proposed key exchange model is shown in Figure (1). In key exchange process three messages are used, the first one is used to establish the public key

parameter between two sides, the Remainder two messages are the acknowledge of this message which carries the public key.

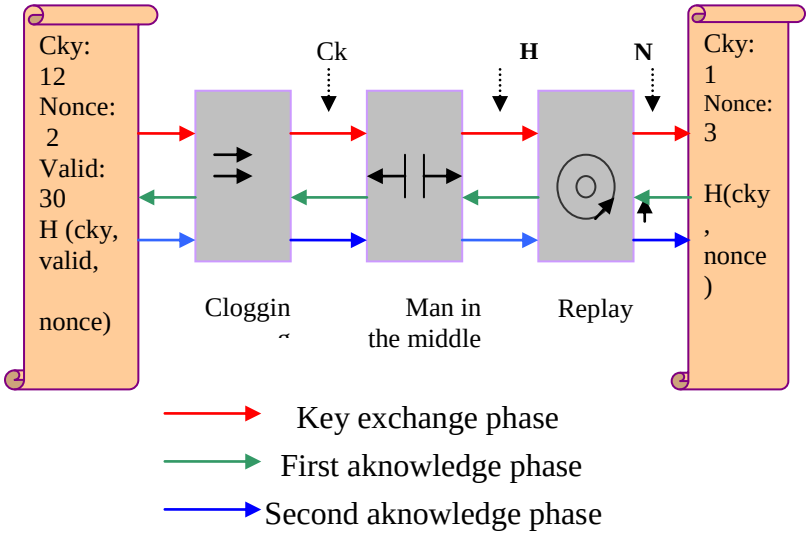

Figure (1) Proposed System key exchange model

4-3-1 Key Exchange Message format

Figure (2) shows the proposed key exchange message format:

- - - - - Begin Privacy Enhanced Message - - - - -
Proce_Type:1, KEYEXCHANGE
Cky:
Nonce:
Issuer:
Responder:
Activate Date:
Validity:
Key_Xchng_Info:
<Text>
- - - - - End Privacy Enhanced Message - - - - -

Figure (2) Key Exchange Message format

The purpose and format of each field are as follow:

1. **Proc-type:** refer to the type of message; the number "1" in these messages and all other messages refers to the first version of key management system.

2. Cky: This field carries an integer value; this value is secret number between both sides and changed dynamically with date. This value is intended to authenticate the claimed source and prevent clogging attack, in which the opponent requests a high number of keys; the victim spends considerable computing resource doing useless key generation Process rather than real work.

In proposed system Cky is the value of agreement date between two parties, for example if agreement date =03/09/2010 then Cky is,

Cky = 3+9+2010
Cky = 2022

When a new message wants to exchange, the system gets the Cky between the sender and the receiver from its database and adds it to current date, for example if *current date = 10/05/2010 then*
Cky= Cky + (day + month)
Cky=2022 + (10+5)
Cky=2037

3. Nonce: is an integer value like a sequence between sender and receiver to prevent replay attack.

4. Activate Date: is the date, in which the key is considered valid at this date, this date is computed according the following Process:

Cdate = 05/03/2010, Then
Activate date = 3× (Delay + Error) + cday
If delay =3 and Error =2 then
Activate date = 3 × (3 +2) +5
Activate date = 15 +5
So Activate date = 20/3/2010

This means that the system takes in account the time needed to compute a knowledge between both sides, where 3 (is the time to send first message "key exchange" + the time needed to send first acknowledge +the time needed to send second acknowledge).

5. Validity: is the time to live (key life) a system maintains three intervals (30 days, 90 days, 180 days).

6. Key _ Xchng_Info: This field carries an integer parameter which carry a Diffie Hellmann parameter, this parameter is used in the acknowledge message to encrypt public key, and it is computed over (a, p) global parameter between both sides. The sender fills this field with (a^{xi}), where xi is computed dynamically using current time as follows:

Xi = (H+M+MO) mod H.

7. Text: carries the value (alg ID + R64 [H (Cky || validity || nonce || a^{xi})]).

Hash is computed over (Cky, nonce, validity, a^{xi}) to prevent tampering.

4-3-2 First Acknowledge message format

Figure (3) shows the proposed message format for First Acknowledge.

- - - - - Begin Privacy Enhanced Message - - - - -
Proce_Type:1, AKNWLEDGE
Cky:
Nonce:
Issuer:
Responder:
Public_Key_Info:

<Text>

- - - - - End Privacy Enhanced Message - - - - -

Figure (3) First Acknowledge Message format

The Cky, Nonce, Issuer, and Responder are the same as in key exchange massage, the new field is

1. Public_Key_Info: carries two parameters, one is the public key intended to exchange, this key is encrypted using EL Gamal algorithm and a^{yr} for the responder, yr as mentioned previously is computed dynamically like a^{xi}. This field will look like: R64 [E (public key) || a^{yr})].

2. Text: carries the value (153 + R64 [S_{kr} (H (cky_i || cky_r || $nonce_i$ || $nonce_r$ || a^{xi} || a^{yr}|| E (public key))]).

S_{kr} means encryption using EL GAMAL algorithm with sender private key as a signature.

The Second Acknowledge message takes the same message format in the first acknowledge except that the sending value is now from the sender to receiver, what is different is:

1. Public_Key_Info: carries the value (R64 [E (public key) || a^{xi})].
2. Text: carry the value (253 + R64 [S_{kr} (H (cky_i || cky_r || $nonce_i$ || $nonce_r$ || a^{xi} || a^{yr}|| E (public key_i || E (public key_r))]).

5- Key Revocation Process

Public/private key pair needs to be revoked for two reasons, the first one is when its validity expires then there is a needed to revoke it, and in the second reason when the security of its private key is compromised, in both cases the system provide a mechanism to manipulate it, wherein the first case the revoke message is constructed automatically by the program when the validity of the key are expired, a special process is implemented by the system.

In the second case a revoke message can be user directed message.

The next sections shall displays the revoke message format and its associated acknowledge.

5-1 Revoke Message Format

Figure (4) shows the revoke message format:

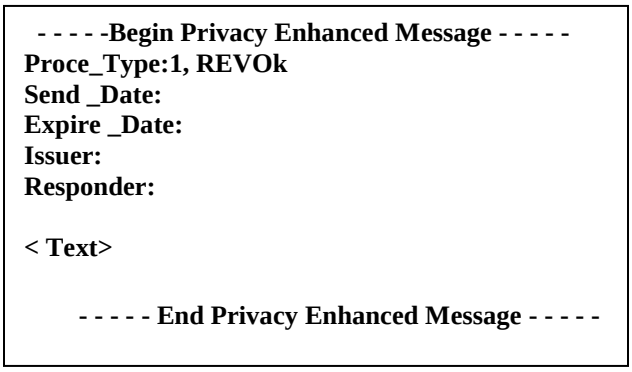

Figure (4) Revoke Message Format

The above header field is the same as in the acknowledge message, what's different is the proce_type and text.

Text = alg ID + R64 [H (sender public key || receiver public key)].

Alg ID: is the algorithm identifier that intended to revoke its key.

5-2 Revoke First and Second Acknowledge

Figure (5) shows the Revoke First and Second Acknowledge

```
- - - - - Begin Privacy Enhanced Message - - - - -
  Proce_Type:1, AKNWLEDGE
Send _Date:
Expire _Date:
Issuer:
Responder:

<Text>
- - - - - End Privacy Enhanced Message - - - - -
```

Figure (5) Revoke First and Second Acknowledge Message Format

In the first acknowledge the text will be

Text =14-- + R64 [H (public)]

If A sends it to B then it looks like text = 14-- + R64 [H (public B)] in the first acknowledge which means that A deletes B's public key from its database.

In the second acknowledge, text = 24-- + R64 [H (public A)] means B deletes A's public key from its database.

Send and expire date here are just encoded and we see that ,no encryption process is applied to the hash of the public key because no one else except the sender and the receiver knows the public key of each other and this is due the Process of key exchange Process proposed and developed by the system. Two types of revocation process provided by the system as follow:

5-2-1 Automatic Key Revocation Process

This process is implemented by the system, where public key information for both parities is stored in special database called **"KEY TABLE"** the parameter validity and

last modified date is important for this Process, where at each time user logs in the system the first step system implement is the check Process for key Table looking for bad keys (i.e. expired keys) this is done by computing the date from last modified date to the current date subtracting the value from time to live (validity) and then checks if the validity is less than or equal to zero. If it's then a revoke message is performed and a user notes about it.

5-2-2 User Directed Process

User can prepare revoke message by submit receiver name, algorithm ID, sender and receiver public key, this information will be written in the revoke message.

6- Experimental Result

This example demonstrates the output of key exchange message proposed by the system . The sender "Mgaz" wants to exchange his public key with the receiver "Alaa78" , so Mgaz will create the key exchange message with the suitable parameter. Figure (6) shows the key exchange form with specified information (issuer name and responder name, key validity and the intended public key algorithm). Table (3) shows the public parameter needed to form the message.

Table (3) key exchange example parameter

Parameter	Value
Cky	2003
Nonce	٣٨
a	5
P	97
Current date	01/02/2010
Delay	3
Error	2

This information is retrieved from certificate database and sequence table depending on issuer and responder name

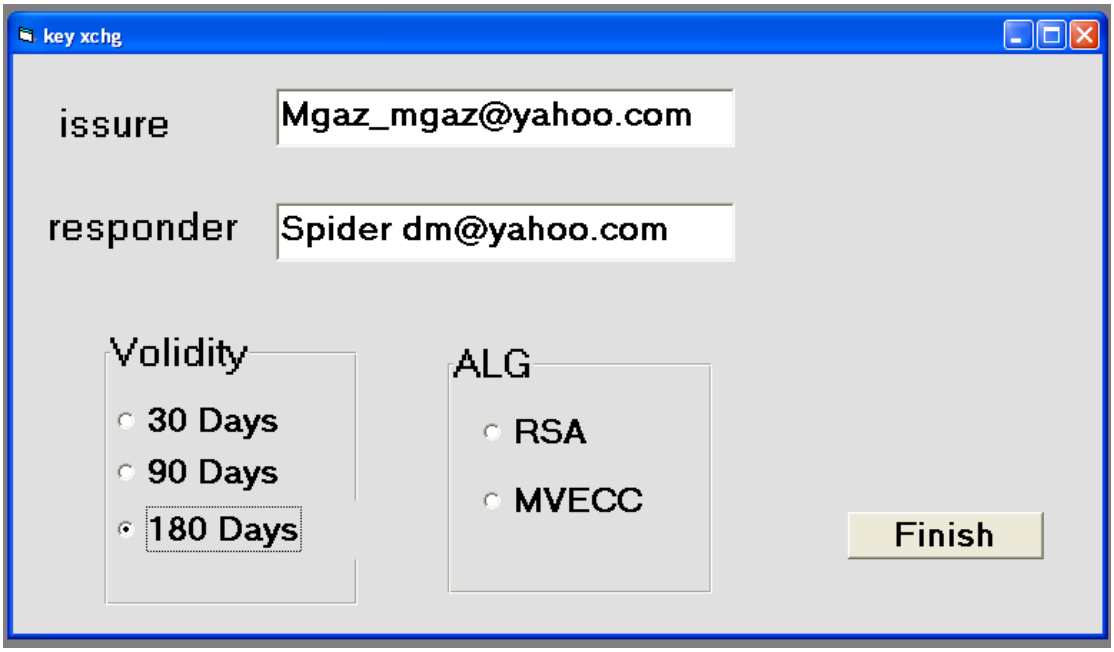

Figure (6) Key exchange information

Figure (7) shows the hash value to the component (Cky, nonce, validity, activate date, a^{xi}). Xi is dynamic value computed depending on current time and represents the secret key to the sender for EL Gamal algorithm, with the prepared key exchange message.

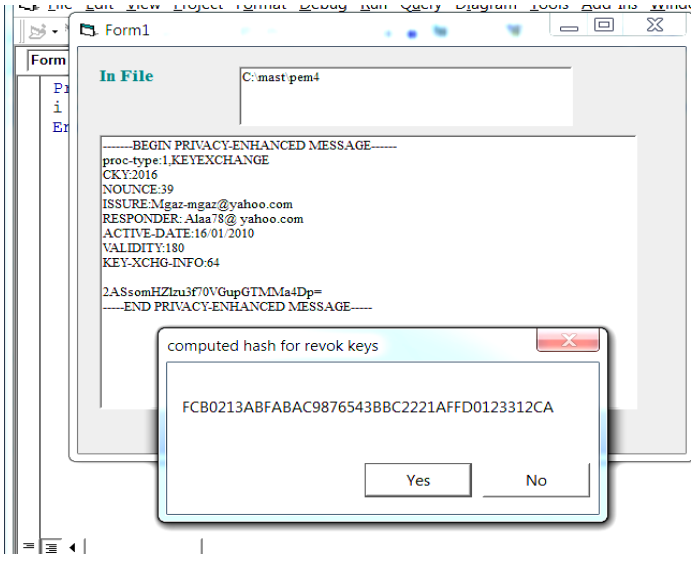

Figure (7) Prepared key exchange message

In short, The prepared key exchange message will be sent from Mgaz to Alaa78, if the message modified in the transmission by man in the middle attack this will be easily discovered by the integrity value computed using secure hash function. If the opponent intercept the message then replay it in another time this will be easily detected by the value of the nonce which will be repeated for two recipient message, finally cky ensure that the clogging attack will be prevented.

7- Revoke

This example explains the automated key revocation process and this message is prepared by the system and acknowledge about it. Figure (8) shows key table with specified record which contains pair of keys with expire time.

Current date = ***26/1 /2010,*** after performing the check process, it is found that this key pairs must be revoked. The intended receiver here is "alimaster@yahoo.com". With delay =2 and error =1. Figure (9) shows the prepared revoke message.

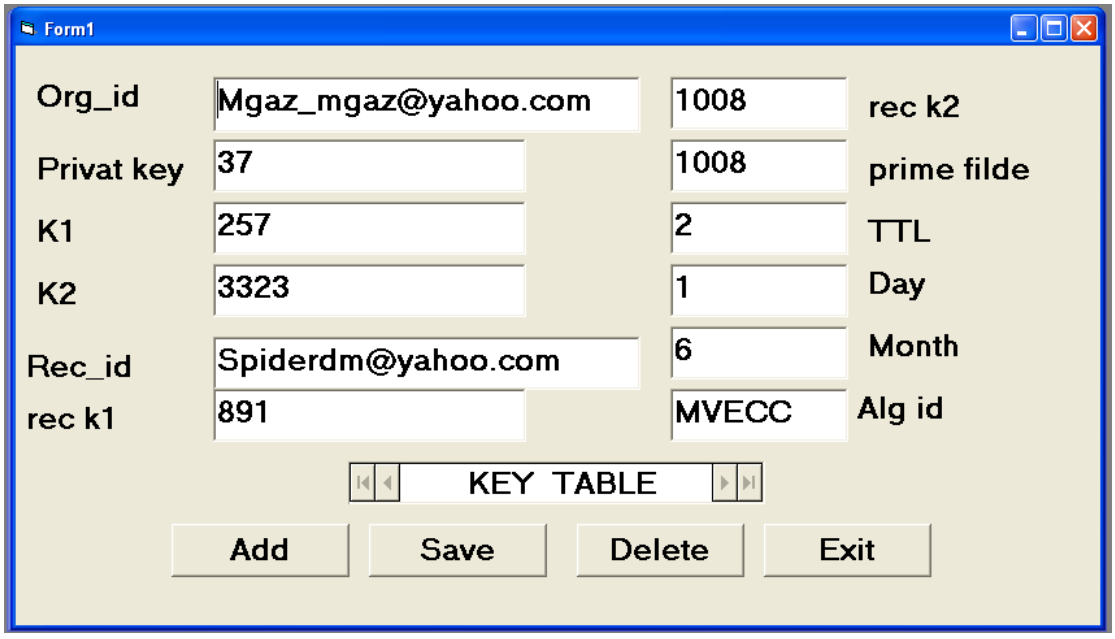

Figure (8) Information to be revoked from key table

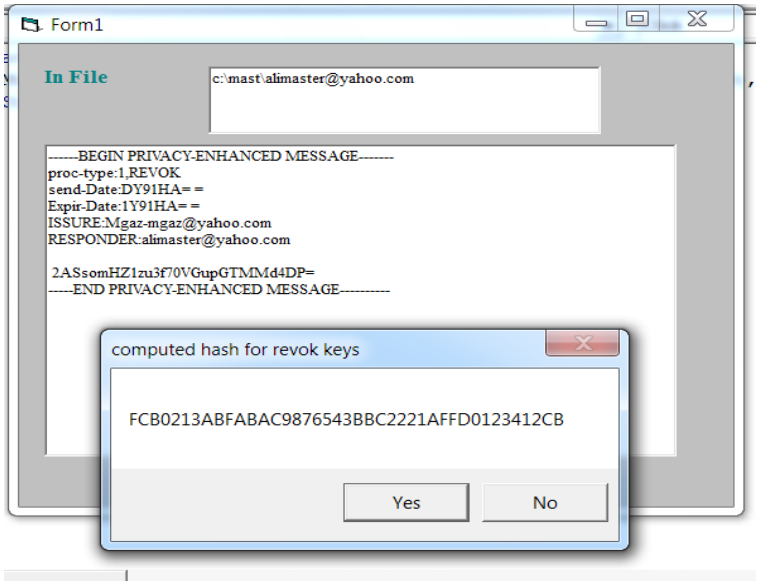

Figure (9) Prepared revoke key message

As shown in Figure (9) the specified path where this message is saved is "***c:\mast\Spiderdm@yahoo.com***" this means that the system can issue more than one revoke message at check time without any conflict in their path.

8- Conclusion

In this paper we implement a key management technique which is an enhanced technique for the PEM key management and can be used with any cryptographic system. The advantages of this key exchanging process can be summarized as follows:

1. It allows generated public key only when it's needed.
2. It protects transmitted public key by encryption using EL GAMAL public key algorithms with dynamically changed secret key.
3. It provides authentication, it easily allows discovering tampering with public key.
4. It prevents man- in-the-middle attack, replay attack, and clogging attack.
5. It can be initiated easily between any two parity anywhere without the need for a third party, what's needed is to agree upon some global parameter such us (Cky, a, p) before transmitting.

.

6. It is applicable to a personal environment and commercial environment.

References

1. Schneier B.,"E-mail Security: How to Keep Your Electronic message Private", John Wiley &Sons, Inc.1995

2. J. Linn,"RFC 1421: Privacy Enhancement for Internet Electronic Mail: part I: Message Encryption and Authentication Procedures ", standard track, networking group, 1993.

Site = [www.fags.org/rfcs/rfc1421.html].

3. William Stallings, "Network Security Essentials: Applications and standards", Prentice Hall, New Jersey, 2000.

4. Schneier B. "Applied Cryptography second edition protocols, Algorithms, and Source codes in C", John Wiley and Sons, Inc, 1997.

5. S. Kent " RFC 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management", Network Working Group, 1993.

Site = [www.fags.org/rfcs/rfc1422.html].

6. Behrouz A. "Cryptography and Network Security ",McGraw-Hill, 2008.

7. B. Kaliski " RFC 1424: Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services", Network Working Group, 1993.

Site = [www.fags.org/rfcs/rfc1424.html].