©gopalax -International Journal of Technology And Engineering System(IJTES): Jan –March 2011- Vol.2.No.2.

The Design and Implementation of Passwords Management System using Blowfish Algorithm

[1]Savita Devidas Patil and [2]Ashish T.Bhole

[1]Department of Computer Engineering
[1,2]SSBT's College of Engineering & Technology, Jalgaon, Maharashtra, India
e-mail: savpats@yahoo.com
[2]Asst. Prof., Department of Computer Engineering
SSBT's College of Engineering & Technology, Jalgaon Maharashtra, India
e-mail: ashishbhole@hotmail.com

***Abstract*- Now a days, especially on the Internet, you might have found one user having more and more usernames or IDs and passwords, which contains users private information. It is very difficult to remember all usernames and passwords and it is unsafe to write them down on you notebook. To solve this problem, this paper explained a design and implementation of Passwords Management System, by which you can manage your usernames and passwords efficiently. It was made based on Blowfish Algorithm, which is a symmetric block cipher provides high security and designed by Bruce Schneier. Because of the difficulties associated with remembering passwords, a group of software applications, password managers has emerged. These applications deal with everything from the simple storage of user IDs and passwords to the management of password access across many users. Blowfish algorithm will provide security to such application.**

***Keywords* - Password management; Blowfish algorithm; key; Cryptographic Algorithm component**

I. INTRODUCTION

Cryptography is the practice and study of hiding information. Modern cryptography intersects the disciplines of mathematics and computer science. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. Cipher is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a key. Cryptanalysis is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations Encryption the conversion of information from a readable state to nonsense. Decryption the conversion of nonsense data generated by encryption into original information

There are various basic algorithms used for security purpose such as Blowfish, Data Encryption Standard (DES), Advanced Encryption Standard (AES) Triple DES But out of these Blowfish is very fast and secure algorithm for providing security and managing passwords.

II. BLOWFISH ALGORITHM

Blowfish is a keyed, symmetric block cipher, included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. However, the Advanced Encryption Standard now receives more attention. Schneier designed Blowfish as a general-purpose algorithm, intended as an alternative to the DES and free of the problems and constraints associated with other algorithms.

Description of Blowfish is given as follows:

Blowfish is a 64-bit block cipher with a variable length key. Key length of implemented algorithm is 132 bit. Algorithm consist of two parts-

- Key expansion- converts a key of up to 448 bits into several subkey arrays totaling 4168 bytes.
- Data encryption –consist of simple function iterated 16 times. Each round consist of key dependent permutation and a key- and data dependent substitution.

All operations are additions and XORs on 32-bit word. Only additional operations are four index array data lookups per round. Blowfish uses large number of subkeys. These keys must be pre computed before any data encryption or decryption.

The P-array consists of 18 32-bit subkeys:

$$P_1, P_2, \ldots\ldots., P_{18}$$

Four 32- bit S-boxes have 256 entries each:

$$S_{1,0}, S_{1,1}, \ldots.., S_{1,255}$$
$$S_{2,0}, S_{2,1}, \ldots.., S_{2,255}$$
$$S_{3,0}, S_{3,1}, \ldots.., S_{3,255}$$
$$S_{4,0}, S_{4,1}, \ldots.., S_{4,255}$$

The original subkey *pbox* and *sbox* are fixed. They are initialized in order with a fixed string that consists of hexadecimal digits of *Pi*(less the initial 3)

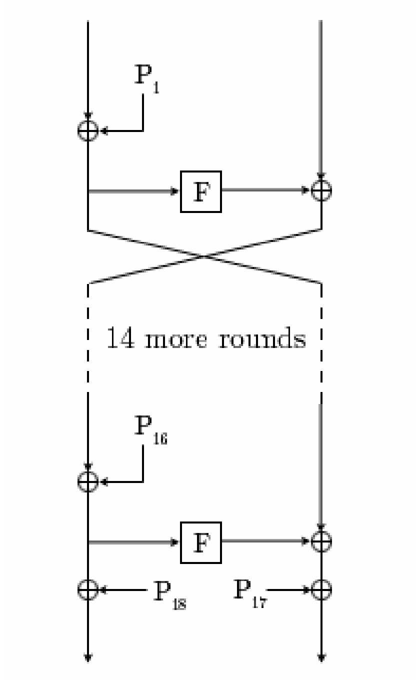

Figure 1 Data Encryption process of Blowfish.

Data encryption occurs via a 16-round Feistel network after key expansion. Each round consists of a key dependent permutation and a key-and-data-dependent substitution. The above figure 1 shows the action of Blowfish. Each line represents 32 bits. The algorithm keeps two subkey arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries.

The upper right shows Blowfish's F-function. The function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 2^{32} and XORed to produce the final 32-bit output. Function F fallows Divide xL into four eight-bit quarters: a, b, c, and d

$$F\ (xL)=F(\ a,b,c,d)=((S_{1a}+\ S_{2b})\oplus S_{3c})+\ S_{4d}$$

Herein, “+” is addition on 32-bit words, and “$\oplus$” represents XOR; S1,a represents key_sbox[1][a], and similar of others. The process of decryption is the same as encryption, except that key_pbox is used in the reverse order. Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order. This is not so obvious because xor is

The subkeys are calculated using the Blowfish algorithm. The exact method fallows:

1. Initialized the first P-array and then the four S-boxes, in order with a fixed string.
2. XOR P_1 with the first 32-bit of the key XOR P_2 with the second 32-bits of key and so on. Reaped cycle until P-array has been XORed with key bits.
3. Encrypt all the zero string with blowfish algorithm using subkeys generated as above.
4. Replace P_1 and P_2 with the output generated of step 3.
5. Encrypt the output of step 3 using blowfish with the modified subkeys.
6. Replace P_3 and P_4 with the output of step 5
7. Continue the process, replacing all elements of the P-array and then all four S-boxes in order, with the output of continuously changing Blowfish algorithm.

III. DESIGN OF PASSWORD MANAGEMENT SYSTEM

Everyone has the security problem of private information which can be solve by password protection and extraction. Password Management System (PMS) is due to manage a variety of username and password info on Internet, which is stored in the disk file (called *password database*) in the form of ciphertext. The user can handle the information in the database as long as he has succeeded in passing the identification. Usually, we set each operation on Internet started through the PMS (e.g.: adding the website address you browse frequently into PMS). When it’s time to enter username and password, we can get them from corresponding record. After the identification, the user can add, modify and get info from his database. The modified information will be restored to the database after encrypted automatically by PMS. Therefore, if the user remembers the identification password, then he remembers all. Whole structure of PMS is shown in following Figure 2.

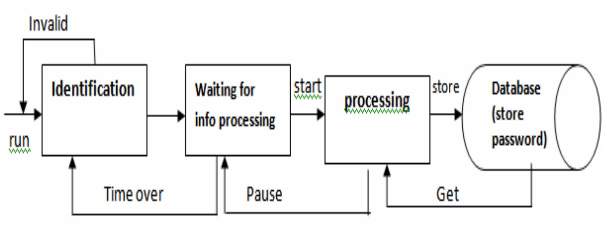

Figure 2.The Structure of PMS

"Waiting for info processing" is added in this system to improve the information security which allows only 3 attempt for enter into processing. After identification or when info processing has paused for the time, PMS will return to identification module, which means you have to be identified again if you want to reprocess the info. PMS will return to the "identification" module once the time exceeds than the prescribed time, which effectively prevent the info from extracted by others because of the users long leaving after open the database. According to the framework, the program structure of PMS can be divided into two parts: frontend and backend. The frontend offers user an interactive platform in the form of graphical interface and captures commands send by the user. The backend is the core of PMS, which has blowfish implementation, responsible for proper running, including storing after encrypted and getting info after decrypted via Blowfish.

The user can operate the fields through the graphic interface, including adding, deleting, modifying etc. Operation commands send by the user will be transferred to the backend to be responded after captured by the frontend. Program structure of PMS is as follows

A. PMS backend methods

1. Method blowfishDecrypt- The normal entry to the decryption process which performs decryption.
2. Method BF_key- This method is only called by the makeKey method to generate a session key from user data.
3. Method blowfishEncrypt- The normal entry to the encryption process. This method, outputs the result in a byte array form.
4. Method makeKey- Expands a userKey to a working Blowfish session key (P) and generates this session s-boxes data (sKey). The key bytes are fist extracted from the user-key and then used, repetitively if need be, to build the contents of this session key and S-boxes values.
5. Method engineUpdate- Performs the actual encryption or decryption process.

B. PMS frontend methods

It is user interaction platform consist of fallowing classes

1. Edit dialog class.
2. Add dialog class.
3. Password setting dialog class.
4. Password manager dialog class.
5. Delete record dialog class.

IV. IMPLEMENTATION OF PMS

The program structure was divided into frontend and backend. Data processing in the backend is the core of the system and the implementation of Blowfish is the key part.

lass called *CBlowfish*, which offers an encryption and a decryption interface:

void blowfishEncrypt(const block in, block out);

Input Plaintext;

Output Ciphertext,

void blowfishDecrypt(const block in, block out);

Input Ciphertext;

Output Plaintext

During this blowfish implementation key can be generated by method *makeKey* and session key can be generated by method *BF_key.*

Usernames and related passwords info are stored in the *database* in the form of records which consist of following classes

BF_accounts; //user account

BF_website; // website name

BF_User; //Username

BF u_Password; //Password

BF u_Notes; //Remark

These above all classes are used to manage the fields of a record. All fields of each record are stored in the database after encrypted by Blowfish.

The user can operate the fields through the graphic interface, including adding, deleting, modifying etc. Operation commands send by the user will be transferred to the backend to be responded after captured by the frontend.

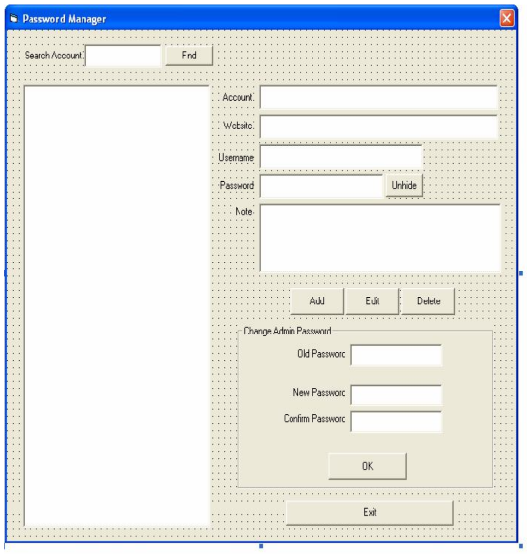

Figure 3.Records displaying of PMS

VI. CONCLUSION

Blowfish cipher is not only secure, but also fast, and suitable for different platforms, therefore, it has a high value of application in the field of info security. The information in the database of password management system is stored in the form of ciphertext that cannot be read by other viewers, while only by passing the identification can the user use the database, which means the system is secure and reliable. It is useful in web services. In future this system will be improved according to following areas

1) In data warehouse and data mining application.
2) In embedded system.
3) Used in Internet for web applications.

REFERENCES

[1] Mingyan Wang, Yanwen Que, "The Design and Implementation of Passwords Management System Based on Blowfish Cryptographic Algorithm," *vol. 2, pp.24-28, 2009 International Forum on Computer Science-Technology and Applications, 2009*

[2] Krishnamurthy G.N, Dr. V. Ramaswamy, Leela G.H, Ashalatha M.E, "Performance enhancement of Blowfish and CAST-128 algorithms and Security analysis of improved Blowfish algorithm using Avalanche effect" *VOL.8 No.3, March 2008 IJCSNS International Journal of Computer Science and Network Security*

[3] B.Schneier, The Blowfish Encryption Algorithm. July 22, 2009 from http://www.schneier.com/blowfish.html

[4] Bruice Schneier, "Applied Cryptography-Protocol, Algorithm, and Source Code in C" Second Edition 2008

[5] http://www.sans.org/reading_room/whitepapers/ sysadmin/options-secure-personal-password-management_1287

[6] http:// Cryptographic Algorithms and Protocols.htm

[7] B.Schneier, "Blowfish—One Year Later", available online at http://www.schneier.com/paper-blowfish-oneyear.html

[8] http://www.schneier.com/blog/archives/2010/05/ canada_spending.html