International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 1– No.9, April 2012 – www.ijais.org

Passwords Management System using Blowfish Cryptographic Algorithm with Cipher Block Chaining Mode

Ashish.T. Bhole
Assistant Professor
Department of Computer Engineering
SSBT's College of Engineering & Technology
Bambhori, Jalgaon, India

Savita D. Patil
Research Scholar
Department of Computer Engineering
SSBT's College of Engineering & Technology
Bambhori, Jalgaon, India

ABSTRACT
Everyone has the security problem of private information and increase in demand of Internet use, creates security problems which results in loss of internet users passwords. There are too many users on the Internet; you might have found one user having more usernames and passwords, which contains user's private information. It is difficult to remember all usernames and passwords and unsafe to write them on paper. Password security and data management is very difficult task. Security can be provided by cryptography and data management can be provided by password management system. To solve this problem, this paper explained a design and implementation of Passwords Management System, by which one can manage his usernames and passwords efficiently. Proposed Password Management System (PMS) is based on Blowfish Algorithm that operates on symmetric block cipher with 56-bytes key. The implementation of Password Management System (PMS) uses blowfish cryptographic algorithm in cipher block chaining mode (CBC) which provides high security. Implemented system deals with everything from the simple storage of user name and passwords to the management of password access across many users.

Keywords
Password management system, Blowfish algorithm, CBC, blowfish key, Cryptographic algorithm

1. INTRODUCTION
Cryptography has been long in use by governments, military sector and other areas for security purpose. Cryptography is used in the practice and study of hiding information. Modern cryptography intersects the disciplines of mathematics and computer science [5]. Cryptography has wide range of applications such as 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 [3]. 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 [1].

2. BLOWFISH ALGORITHM
Blowfish features a variable key length up to 448 bits. 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 [4]. Blowfish is a general-purpose algorithm, intended as an alternative to the DES and free of the problems and constraints associated with other algorithms. Blowfish is a 64-bit block cipher with a variable length key [8]. 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 consists 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. 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:

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

The original sub key *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) Fig 1 shows the action of blowfish. Each line represents 32 bits. The algorithm keeps two subkeys arrays such as the 18-entry P-array and four 256-entry S-boxes. 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 [5].

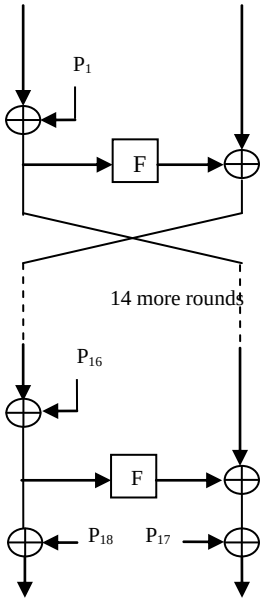

Fig 1 : Process of Blowfish algorithm.

Blowfish uses Feistel network after key expansion performs a 16-round in encryption. 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 [8].

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 [4]. Function F fallows Divide xL into four eight-bit quarters: a, b, c, and d

$$F\ (xL) = F(\ a, b, c, d) = ((S1a + \ S2b) \oplus S3c) + \ S4d$$

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, such that P1, P2, .., P18 are used in the reverse order. This is not so obvious because xor is commutative and associative. 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.

2.1 Operational Modes of Blowfish

Blowfish is a symmetric block cipher that can be used as replacement for DES or IDEA so that it can be used in four standard operation modes as DES and IDEA as follows:

2.1.1 Electronic Codebook Mode.

Electronic codebook (ECB) mode operates on block cipher. In ECB mode a block of plaintext encrypts into a block of ciphertext.

2.1.2 Cipher Block Chaining Mode.

Cipher block chaining (CBC) mode is more secure than ECB mode. In CBC mode an initial value is added modulo 2 (XORed) to the first plaintext block to form the Blowfish input block. This output is fed back and added modulo 2 to the next plaintext block forming the new Blowfish input block [5].

2.1.3 Cipher-feedback Mode.

Block ciphers can also be implemented as a self synchronizing stream cipher; this is called cipher-feedback (CFB) mode. In this mode, input is processed by j bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce the next unit of ciphertext.

2.1.4 Output-feedback Mode.

The output-feedback (OFB) mode is a method of running a block cipher as a synchronous stream cipher. It's similar to CFB mode, except that j bits of the previous output block are moved into the right-most positions of the quence.

2.2 Operational Mode Selection

Performance of blowfish algorithm is high in CBC mode. As expected, CBC requires more processing time than ECB because of its key-chaining nature. But CBC is much better than ECB in terms of protection [11]. Therefore implemented PMS uses cipher block chaining mode (CBC) to perform operation that provides more security.

3. DESIGN OF PASSWORD MANAGEMENT SYSTEM

The security problem of private information can be solving 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. Password manager are not only store passwords, but manage access to them among users and groups. The user can handle the information in the database as long as he has succeeded in passing the identification by master key password. When it's time to enter username and password, we can get them from corresponding record. After master key password, 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 Fig 2.

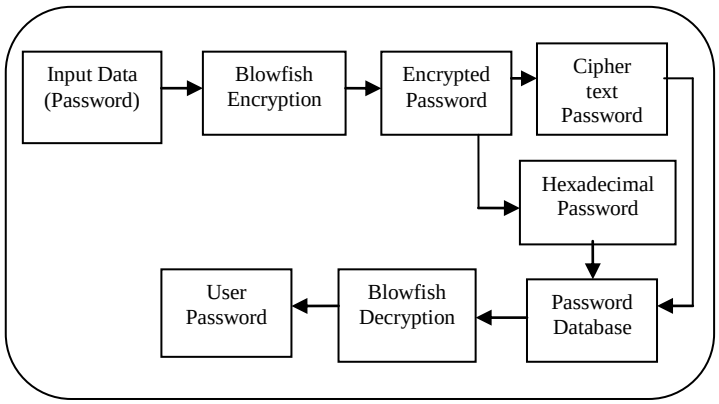

Fig 2: The Structure of PMS

3.1 General Steps of Password Management system

1. User search account from already stored account list if found then use it, otherwise user can add new account with its password.
2. Adding new account will add website name, username and password.
3. PMS uses blowfish key which encrypt users password using blowfish algorithm.
4. Blowfish algorithm will generate encrypted password in the form of hexadecimal and unreadable ciphertext.
5. These encrypted passwords are stored in one database which is locked with one master key which makes it more secure.
6. Deletion, edition operation can be performed on stored account.

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 of the user will be transferred to the backend to be responded after captured by the frontend.

4. IMPLEMENTATION OF PMS

Implemented PMS uses Blowfish algorithm which operates on 448-bits key using CBC mode operation with and without padding. 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.

4.1 PMS backend methods

Blowfish algorithm was implemented by means of class called basBlowfish, which offers an encryption and a decryption interface:

1. basBlfArrays: Initialises P-array and S-boxes
2. blf_EncipherBlock: Encrypts two words
3. blf_DecipherBlock: Decrypts two words
4. blf_Initialise: Initialise P&S arrays using key
5. blf_KeyInit: Initialise using byte-array key
6. blf_EncryptBytes: Encrypts an block
7. blf_DecryptBytes: Decrypts an block
8. blf_Key: Initialise key length

4.2 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.

4.3. Database Setup

All passwords are stored in one database, which is locked with one master key. Database consists of following entries:

1. User account
2. Website name
3. Username
4. Password
5. Pkey1; // Blowfish key
6. En_Pkey1; // encrypted blowfish key
7. Password; // encrypted password ciphertext
8. CipherHex_password; // encrypted password in hex

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.

4.4. Results of Proposed System

Performance of implemented system is describes by size. Database storage size comparison can be done. Implemented system compares with Linux encryption algorithms. During this comparison we took password as input having same size and same text in all compared systems. Ciphertext size comparison can be shown in Table 1.

Table 1 : Ciphertext size comparison

Algorithm	Input/ Password	Input Size in Bytes	Ciphertext	Cipher Size in Bytes
PMS Blowfish	manesh 555	9	3CA96D194BE003D86145C95ECB09F9F0	32
AES	manesh 555	9	4eace63ced447f06f90d4722775958cab3f9cced075e7fop1	48
Twofish	manesh 555	9	4eace63ced447f06f90d4722775958cab3f9cced07	42
Linux-Blowfish	manesh 555	9	05$PFePcBqiv.hdc0I7PK0E7.fk9zjNVPUlFL1DW2kdKqTsvNkwNy3Mq	56
Linux-3DES	manesh 555	9	A8ZI2SA3FxvVo.$.Fpiklom2z9IN2WKlhskin8iq	40
Linux-MD5	manesh 555	9	wcd2rOx1.$.OpTjPcJsKj0NIdGlevUz1	32

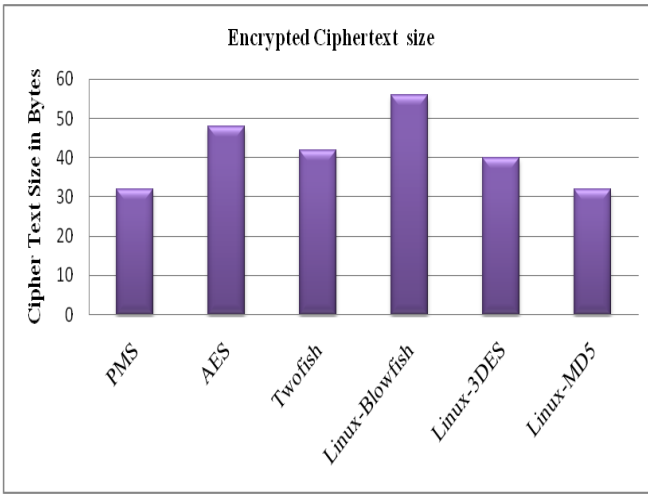

Fig3 : Performance Results of Ciphertext size

Fig.3 shows the bar series that denotes ciphertext size in bytes, which shows how much space required to stored ciphertext in database. Fig. 3 shows implemented PMS requires less space compare with other systems.

Performance evaluation of implemented PMS can also be done by taking results of different size input as passwords and how much long and secure it generates output as ciphertext can be taken. Table 2 shows results of different size passwords and ciphertext.

Table 2 : Results of different size passwords

Sr No	Password Text	Password Size In Bytes	Ciphertext	Cipher Size In Bytes
1	swami	5	5299A2EE0CEB142E	16
2	abcdefgh	8	314C515FE6F7A9ED A50CE1FDD0E6886 A	32
3	Manesh555	9	39822F777D652A416 4E13928366E4409	34
4	swamiaama rth123s	16	B2D51D626BD39A2F 1A2342D614575D035 12AE69CB8E58EF2	48
5	abcdefghijk lmnopqrstu vwxyz1234 s9	25	FD427095EE4092C10 DD450E5F3F2559997 A3CDC84E1FF0457C F4716A7A9EC483855 75A595D901229	64

By this observation we conclude that if we use large password length it produces large size ciphertext which provides high security. Fig4 shows performance results of strength of encrypted ciphertext size and password size taken as input.

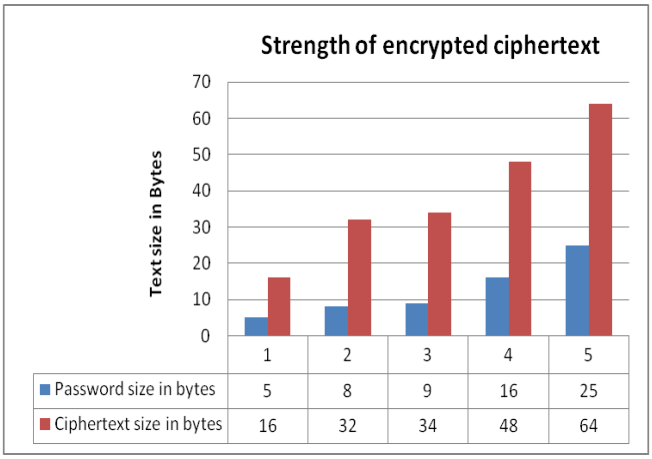

Fig4: Performance results of strength of encrypted ciphertext

5. CONCLUSION

The objective of the implementation of PMS is to provide security to user's passwords in order to give protection from hacking. PMS uses blowfish algorithm in CBC mode with 448 bits key size. 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 information security. The information in the database of password management system is stored in the ciphertext and hexadecimal format that cannot be read by others, while only by entering master key password and passing the identification, the user can use the database, which means the system is secure and reliable. It provides more security by blowfish algorithm. In future PMS can be improved in data warehouse applications and embedded systems.

6. REFERENCES

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

[2] Savita D. Patil, Ashish T. Bhole, "The Design and Implementation of Passwords Management System using Blowfish Cryptographic Algorithm", *International Journal of Technology And Engineering System* (IJTES), Vol. 2, No. 2, pp. 193-196, Jan–March 2011.

[3] 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" IJCSNS *International Journal of Computer Science and Network Security*, Vol. 8, No. 3, March 2008.

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

[5] Bruice Schneier, "Applied Cryptography - Protocol, Algorithm and Source Code in C", Second Edition, John Wiley & Sons, 2008.

[6] Afaf M. Ali Al-Neaimi, Rehab F. Hassan, "New Approach for Modifying Blowfish Algorithm by Using Multiple Keys", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 11, No. 3, March 2011.

[7] Diaa Salama Abd Elminaam, Hatem Mohamed Abdual Kader and Mohiy Mohamed Hadhoud, "Evaluating The Performance of Symmetric Encryption Algorithms", *International Journal of Network Security*, Vol.10, No.3, pp. 216–222, May 2010.

[8] B.Schneier, *"*The Blowfish Encryption Algorithm*", In Dr Dobb's Journa*l, pp. 38-40, April 1994.

[9] Simar Preet Singh and Raman Maini, "Comparison of Data Encryption Algorithms", *International Journal of Computer Science and Communication*, Vol. 2, No. 1, pp. 125-127, January-June 2011.

[10] A. H. Al-Hamami, M. A. Al-Hamami and S. H. Hashem, "A Proposed Modifications to Improve the Performance of Blowfish Cryptography Algorithm", *First National Information Technology Symposium (NITS 2006)*, 5-7 Feb. 2006.

[11] Gurjeevan Singh, Ashwani Kumar Singla, K.S. Sandha, "Throughput Analysis of Various Encryption Algorithms", *IJCST*, Vol. 2, Issue 3, September 2011.

[12] Ashwaq T. Hashim , Dr. Saleh M. Al-Qarrawy, Janan A. Mahdi, " Design and Implementation of an Improvement of Blowfish Encryption Algorithm" *IJCCCE*, Vol. 9, No. 1, 2009.

7. AUTHORS PROFILE

Ashish T. Bhole is working as Assistant Professor in the Department of Computer Engineering, SSBT's COE & T, Bambhori, Jalgaon, India. He received B.E. Degree in Computer Engineering in 1999 from SSBT's COE & T, Bambhori, Jalgaon affiliated to North Maharashtra University, India, M. Tech. in Computer Science & Engineering in 2008 from S. A. T. I., Vidisha affiliated to Rajiv Gandhi Technological University, India and currently pursuing Ph.D. in Computer Science & Engineering. He has 12 years of teaching & research experience. He has published 15 papers in various International & National Journals, Conference Proceedings. He is a Member of Institution of Engineers (India) & Life Member of Indian Society for Technical Education. His research area includes Computer and Wireless Networks, Network Security, Routing Protocols and Traffic Engineering.

Savita D. Patil is a Research Scholar in the Department of Computer Engineering, SSBT's COE & T, Bambhori, Jalgaon, India. She received B.E. Degree in Information Technology in 2005 from R.C. Patel Institute of Technology, Shirpur affiliated to North Maharashtra University, India. She has 7 years of teaching experience. She has published 6 papers in various International and National Journals, Conference Proceedings. She is a Life Member of Indian Society for Technical Education. Her research area includes Information Security, Web Application and Database Security.