New Approach for Modifying Blowfish Algorithm Using 4-States keys

Afaf M. A. Al-Neaimi
Department of Software Engineering, AL-Zaytoonah University of Jordan,
Amman,Jordan
drafafn@alZaytoonah.edu.jo (Phone: o796576167)

Rehab F. Hassan
Department of Computer Science, University of Technology,
Baghdad, Iraq
rfhassan68@yahoo.com (phone: 009647700481784)

New Approach for Modifying Blowfish Algorithm Using 4-States keys

Abstract

Within the last decade, there has been a vast increase in the accumulation and communication of digital computer data in both the private and public sectors. Much of this information has a significant value, either directly or indirectly, which requires protection. One of the most important protection methods is inventing and developing different encryption algorithms. The algorithms uniquely define the mathematical steps required to transform data into a cryptographic cipher and also to transform the cipher back to the original form.

This paper introduces a new method to enhance the performance of the Blowfish algorithm. This is done by replacing the predefined XOR operation applied during the 16 round of the standard algorithm by a new operation depends on using two keys, each key consists of a combination of 4 states (0, 1, 2, 3) instead of the ordinary 2 state key (0, 1). This replacement adds a new level of protection strength and more robustness to breaking methods.

Key words: computer security, data encryption standard, triple data encryption algorithm.

1. Introduction

Blowfish is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneier and 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 aging DES and free of the problems and constraints associated with other algorithms. At the time Blowfish was released, many other designs were proprietary, encumbered by patents or were commercial/government secrets. Schneier has stated that, "Blowfish is unpatented, and will remain so in all countries. The algorithm is hereby placed in the public domain, and can be freely used by anyone."

2. Related work

Cryptography has a long and fascinating history. Beginning with the work of Feistel at IBM in the early 1970s and culminating in 1977 with the adoption as a U.S. Federal Information Processing Standard for encrypting unclassified information, DES, the Data Encryption Standard, is the most well-known cryptographic mechanism in history. It remains the standard means for securing electronic commerce for many financial institutions around the world [6, 7].

The most striking development in the history of cryptography came in 1976 when Diffie and Hellman published New Directions in Cryptography. This paper introduced the revolutionary concept of public-key cryptography and also provided a new and ingenious method for key exchange, the security of which is based on the intractability of the discrete logarithm problem. Although the authors had no practical realization of a public-key encryption scheme at the time, the idea was clear and it generated extensive interest and activity in the cryptographic community [1, 2].

In 1978 Rivest, Shamir, and Adleman discovered the first practical public-key encryption and signature scheme, now referred to as RSA. The RSA scheme is based on another hard mathematical problem, the intractability of factoring large integers. This application of a hard mathematical problem to cryptography revitalized efforts to find more efficient methods to factor. The 1980s saw major advances in this area but none which rendered the RSA system insecure. Another class of powerful and practical public-key schemes was found by ElGamal in 1985. These are also based on the discrete logarithm problem [1,3].

The search for new public-key schemes, improvements to existing cryptographic mechanisms, and proofs of security continues at a rapid pace. Various standards and infrastructures involving cryptography are being put in place. Security products are being developed to address the security needs of an information intensive society [9]. The research is an attempt to improve most cryptographic algorithms that depend on using logical OR operation, then the DES algorithm is explained briefly in section 3, where the proposed improvement will be applied on, section

4 contains a full description to the operation that will be exchanged with the ordinary OR operation. The rest of the paper explains the new improved DES algorithm, with the conclusion and the suggested future work that can be applied.

2. Blowfish Algorithm

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm [4].

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two
parts: a key-expansion part and a data-encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key dependent Permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round. Diagram shown in Figure (1) shows the action of Blowfish. where Blowfish has 16 rounds.
The input is a 64-bit data element, x.
Divide x into two 32-bit halves: xL, xR.
Then, for i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
After the sixteenth round, swap xL and xR again to undo the last swap.
Then, xR = xR XOR P17 and xL = xL XOR P18.
Finally, recombine xL and xR to get the ciphertext [4, 5].
F-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 232 and XORed to produce the final 32-bit output. Since Blowfish is a Feistel network, it can be inverted simply by XORing P17 and P18 to the ciphertext block, then using the P-entries in reverse order

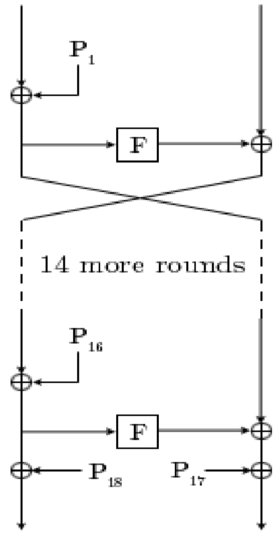

Figure 1 : BolwFish each round action

3. Improved 4-States operation

To increase the security and key space, that makes the encryption algorithms more robustness to the intruders, a new manipulation bits process has been added in [8] by using different truth table for manipulation bits process work on 4-states (0,1,2,3) , while the traditional binary process (XOR) work on (0, 1) bits only. The symbol # has been used to refer to the operator that execute this process use truth tables that shown in figure (2) [8].

#0	0	1	2	3
0	3	2	1	0
1	2	3	0	1
2	1	0	3	2
3	0	1	2	3

#1	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

#2	0	1	2	3
0	2	3	0	1
1	3	2	1	0
2	0	1	2	3
3	1	0	3	2

#3	0	1	2	3
0	1	0	3	2
1	0	1	2	3
2	3	2	1	0
3	2	3	0	1

Figure 2 The truth tables for the # operation

The new operation needs 3 inputs, the first one specify the table number that should be used to calculate the result among the 4 tables, the other 2 inputs define the row and column number in the specified table where the cross point of them gives the result. An example of applying the operation is shown below:

```
    Input 1: 0 1 3 0 1 2 2 3 1
   Input 2: 3 2 2 1 0 1 2 1 1
   Input 3: 1 0 0 2 1 3 2 1 2
 ----------------------------------
  Result  : 1 2 0 0 1 2 2 3 2
```

4. The proposed algorithm to modify Blowfish using 4-states

This research proposed a new improvement to the Blowfish algorithm. The proposed improvement makes use of the new operation defined in the previous section, operation (#) applied during each round in the original Blowfish algorithm, where another key is needed to apply this operation at both sides, this key may come in binary form and convert to a 4-states key, or it may already come in a 4-states as that can be done with quantum channel.

Consequently, two keys will be used in each round of the original Blowfish, the first key K1will be used with the xL and Pi to produce the next left part. The second key K2 will be used with F(xL) and xR to produce the right part.

These three inputs to the # operation should be firstly converted from 32 bits to a 16 digits each may be one of four states (0, 1, 2, 3), i.e., each two bits converted to its equivalent decimal digits see figure(3).

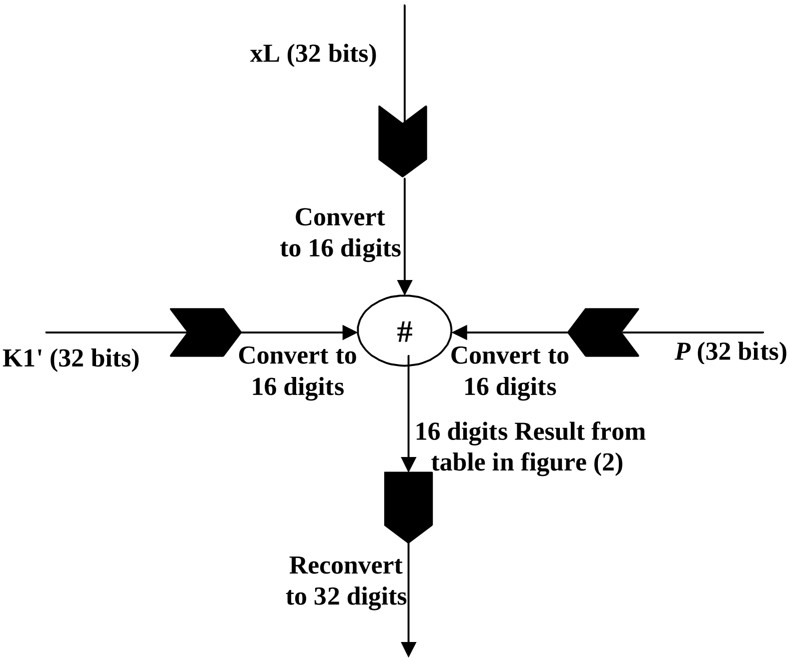

Figure 3 Inputs and Output of the # Operation in DES algorithm

For example, the binary number:
10010111010100101010011111010001001
will be converted to the number:
2 1 1 3 1 1 0 2 2 2 1 3 2 2 0 2 1

Then the # operation will be applied to generate a new 16 digits that should be reconverted to 32 bits , see Figure (4). Full details of the proposed improved Blowfish are given in Algorithm (2) [8].

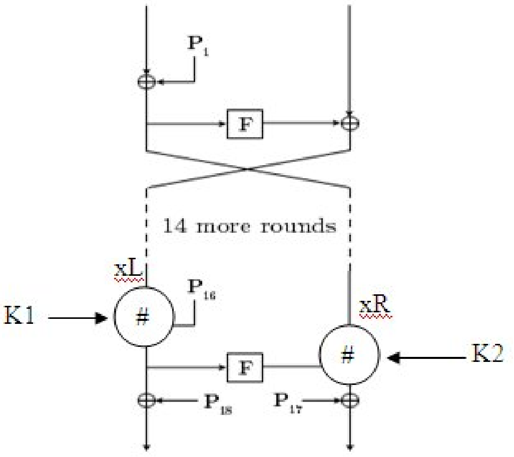

Figure 4 The new structure of each round

5. Implementation

The following example shows how the encryption and decryption operations results will be according to apply the # operation, so in any stage we can expect that the result of the previous left part of the data could be the binary number:

xL = **00101010010101011101010010111110111 0 0**

and the value of Pi-1 which represent here the first key could be the binary number:

Pi = **10010100011101010101001001110100 1101**

and the entered key, which represent the second key in the applied # operation, the binary number:

K1 = **11101010010101011010111010101010010111**

Firstly, the three entered 32-bits binary numbers should by converted to a 4-states 16-digits numbers.

Pi ' = **2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1**

xL ' = **0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0**

K1' = **3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3**

Then the # operation applied according to tables in figure (2) the result of encryption will be:

NewxL = **3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0**

If we reverse the whole operation we will get the initial, which is the result of the decryption operation that equal to the original data:

K1' = **3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3**

Pi' = **2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1**

NewxL = **3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0**

xL = **0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0**

Acknowledgments

The authors would like to thank the Al-Zaytoonah Private University of Jordan in Amman, Jordan for Supporting this publication.

5. Conclusions

Blowfish is now considered to be insecure for many applications. So it becomes very important to augment this algorithm by adding new levels of security to make it applicable and can be depending on in any common communication channel. Adding additional key and replacing the old XOR by a new operation as proposed by this paper to give more robustness to Blowfish algorithm and make it stronger against any kind of intruding. The ciphering process stills simple and can be implemented by hardware in this new proposed improvement, as well as the time complexity of the new algorithm stays the same since only one operation is replaced by another operation, and the conversion operations is very simple and straightforward.

References

[1] Alan G. Konheim"*COMPUTER SECURITY AND CRYPTOGRAPHY* ",2007 , by John Wiley & Sons, Inc.

[2] Alfred J.M., Paul V. C. and Scott A.V., "*Handbook of Applied Cryptography*", Fifth Addition, 2001.

[3} Bruce Schneier, "*Appli Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C*", 1996, Wiley Computer Publishing, John Wiley & Sons, Inc.

[4] B. Schneier, "*Applied Cryptography"*, John Wiley & Sons, New York, 1994.

[5] B. Schneier, "*Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) Fast Software Encryption"*, Cambridge Security Workshop Proceedings (December 1993)Springer-Verlag,1994pp.191-204.

[6] Coppersmith, Don. (1994). "*The data encryption standard (DES) and its strength against attacks*". IBM Journal of Research and Development, 38(3), 243–250.

[7] National Institute of Standards and Technology, (1979). "*FIPS-46: Data Encryption Standard (DES*)" Revised as FIPS 461:1988, FIPS 46-2:1993, FIPS 46 3:1999, available at http://csrc.nist7aznml;'.gov/publications/fips/fips46-3/fips46-3.pdf

[8] Hala Bahjat AbdulWahab1 , Abdul Monem S. Rahma, '*Proposed New Quantum Cryptography System Using Quantum Description techniques for Generated Curves*", The 2009 International conference on security and management, SAM2009, July 13-16 2009, Las Vegas, USA, SAM 2009.

[9] Henk C.A. van Tilborg, Eindhoven , "*ENCYCLOPEDIA OF CRYPTOGRAPHY AND SECURITY*", 2005, Springer Science+Business Media, Inc.