Perspectives in Science (2016) 8, 57—60

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/pisc

Random multiple key streams for encryption with added CBC mode of operation☆

P. Penchalaiah*, K. Ramesh Reddy

Vikrama Simhapuri University, Nellore, Andhra Pradesh, India

Received 22 December 2015; accepted 26 March 2016
Available online 1 April 2016

KEYWORDS
Cryptography;
Security;
Random bits;
Random keys;
BBS;
Cryptanalysis;
Encryption

Summary There are many cryptographic systems that use complex operations involving substitutions and permutations to produce resistant ciphertext, even if the level of the security of these cryptosystems are good, there should be trade-off between security level and operation cost, and the ever increasing virtual infrastructure and mobile, cloud computing technologies creating much more complexities and demanding cost effective and secure cryptographic algorithms. A cryptographic system is said to be secure if the ciphertext does not contain adequate details to find out the matching plaintext. In fact, one can produce unbreakable ciphertext by supplying randomly generated key on each bit of data that is mathematically infeasible to break. Since different random bits or keys would not lead to any repeating patterns.

For the first time, in this paper, we present a construction method to generate multiple random keys from a core-key with highest possible immunity to crack. We are with a particular emphasis on novel technique to secure user data, we have designed a secure and cost effective new cryptosystem called Rbits (Random bits) cypher. In different directions we identify that Rbits having highest immunity to crack and presenting various analysis tests in support from this viewpoint and the analyzed results are reported.

Introduction

The basic principle that all cryptosystem designers must kept in mind is that a cryptosystem should be secure even if the whole thing about the system is public knowledge except the encryption key which is secret. A cryptographic system is said to be secure if the ciphertext does not contain adequate details to find out the plaintext. The general

☆ This article belongs to the special issue on Engineering and Material Sciences.

* Corresponding author. Tel.: +91 9160642505.
E-mail address: penchal.caliber@gmail.com (P. Penchalaiah).

http://dx.doi.org/10.1016/j.pisc.2016.03.007

assumption is that it is impractical to decipher a message by simply knowing the ciphertext plus knowledge of the algorithm. This quality of encryption is what makes algorithms feasible for widespread use. The thing that is available to the unauthorized party is the ciphertext only. The objective is to use the Rbits for secure communication and protecting personal data with a trade-off between security and cost. The basic rule of Rbits cypher is that the ciphertext should not contain adequate details to find out the corresponding plaintext and time required to break the secret code should take more than the useful lifetime of the content.

Methodology

To develop secured and cost effective cryptosystem the services of OTP (Patil and Kumar, 2010) and BBS (Sidorenko and Schoenmakers, 2005) are used. OTP is called as unconditionally secure algorithm (Shannon, 1949) and the key with no repetition patterns that is equal in length to the message to be encrypted is used. Each character of the message is bitwise XOR with the key, to produce ciphertext (Patil and Kumar, 2010). Even though OTP well known for its perfect secrecy, its key size limits its practicality by causing tremendous overheads. Rbits uses OTP behaviour in different methodology and frees up from overheads. The security of the BBS generator depends on the difficulty of factoring '*n*'. Blum, Blum and Shub proved that the 'x^2 *mod n*' generator is unpredictable in polynomial time (for proofs Sidorenko and Schoenmakers, 2005). The BBS generator is unpredictable to left and unpredictable to right. In the applications where multiple keys are required in this way, it is not compulsory to store them all. All key can be effectively computed and recovered from previous values and the initial *x* and *N* (Junod, 1999; Blum et al., 1986).

About Rbits

Rbits (Penchalaiah and Ramesh Reddy, 2013; Penchalaiah and Ramesh Reddy, 2014) is a security mechanism which is symmetric key block steam cypher. In brief Rbits generates random bits at the both ends and these random bits are used to encrypt and decrypt. Random bits are generated by Blum Blum Shub (BBS) algorithm; both the sender and receiver uses BBS algorithm and common shared key say core-key (CK), parameter 'P' which is input to BBS for random bit generation. The key principal of this algorithm is that generating randomly changing multiple keys (Ki) (Bellare et al., 2011), for encryption using a single key CK, P which are common to both sender and receiver (Gu et al., 2011). The proposed Rbits features are: (i) Suitable for both short and long message communication. (ii) Faster in both Encryption and Decryption. (iii) Simple and required no heavy computations. (iv) Complex to cryptanalysis. (v) Added CBC mode gives unpredictable randomness and satisfies avalanche effect. (vi) Rbits is a block stream cypher. (vii) Variable length of block size as input. (viii) For each block of stream new key is used and all keys are distinct. (ix) Total Key length is as long as the length of the plaintext. Rbits can be used in: (i) Secure communication services. (ii) Client/Server communication service. (iii) Web applications. (iv) Secure data store in cloud. (v) Secure backup and restore.

Key selection and generation

The core key CK is derived from the initial steps of BBS (Blum et al., 1986): (i) Select two large prime numbers, x and y, such that $x \equiv y \equiv r \pmod m$, where r = 3 and m = 4. (ii) Compute n = x * y. (iii) Choose a random number 's', relatively prime to 'n'. (iv) Compute $Z_0 = s^2$ mod n and CK = Z_0, Here 'Z_0' is the common core key 'CK' and 'n' is parameter 'P'. Multiple keys are generated by using BBS algorithm. Enough multiple keys Ki of size key_size(key length) are generated to encrypt a message with msg_len(message length) in stream or binary mode. To make the algorithm simple and faster, the proposed operation is XOR. XORing has computational complexity of ''order b'' which is written O(b) where b is no of bits. XOR operation is very simple leads to cost effective.

```
Key Generation Pseudo Code
loop i = 1 to msg_len
  CKi = (CKi-1)^2 mod n
  bi = CKi mod 2
  kj = kj || bi
  if (i % key_size) = 0 then
  j = j + 1
end loop
```

Adding CBC

Even if this mechanism over comes the problem of producing the same ciphertext for a plaintext that appear more than once in the input (the corresponding ciphertext block will also appear more than once in the output). To make this mechanism hardest to cryptanalysis we still adding Cypher Block Chaining (CBC). In CBC, the output of the encryption of the previous block streams is feedback into the encryption of the present block stream. That is, each resultant is used to transform the encryption of the next block stream. Therefore every block of cypher text is reliant on the subsequent current input plaintext block, as well as all the previous plaintext blocks. As we stated earlier in introduction ''The thing that is available to the unauthorized entities is the ciphertext only'', many techniques of cryptanalysis use statistical properties of the available ciphertext. So with the added CBC operation the statistical characteristics of the plaintext are masked to such an extent that any type of cryptanalysis is infeasible (Bellare et al., 2011).

Algorithm

Prior to encryption and decryption process both the sender and receiver must satisfy the pre-requisites. (i) Sender and receiver must have pre-established a security binding (SB). SB defines core-key exchanging, parameter agreement (parameter here means block size). (ii) Both sender and receiver must use same stream block size. (iii) Core-key, Parameter, block size must be confidential and should be infeasible to predict.

Encryption and decryption

Sender follows the sequence of steps for Rbits encryption (Penchalaiah and Ramesh Reddy, 2013). (i) Sender converts message into stream of bits. (ii) The binary stream is divided into specific size of block of bits called block stream (for