A Generalized Key Scheme in a Block Cipher Algorithm and its Cryptanalysis

Soosaimicheal Aruljothi
(Corresponding author)
Research Scholar, Department of Computer Applications, Kalasalingam University, Krishnankoil, Tamilnadu, India.

Dr. Mandadi Venkatesulu
Phd,Senior Professor & Head, Department of SHIP, Kalasalingam University, Krishnankoil, Srivilliputtur (via), Tamil Nadu, India.

ABSTRACT
The prominent block cipher algorithms use two key schemes that characterize different use of keys in the encryption/decryption process. In the family of DES algorithms the keys are used as position pointers for input matrix; in the family of AES algorithm the keys are used as co operand with input matrix. The proposed crypt algorithm combines the idea of keys as position pointers and input matrix as co operand in the encryption/decryption process. The novel use of keys characterizes the nature of the proposed algorithm. This new algorithm found to outperform the existing cryptographic strategies in the common cryptanalysis criteria. This paper discusses and presents a crypt analysis of a new block cipher algorithm based on generalized key scheme.

General Terms
Encryption, Security, Histogram, Randomness, Differential attack, Key Sensitivity.

Keywords
Block cipher, Cryptanalysis, Key Schemes, AES, DES.

1. INTRODUCTION
Internet and network applications are growing very fast and so the need to protect the exchanged information. Encryption algorithms[1] play a important role in securing the exchanged information. Encryption is the process of transforming data into an imperceptible code while the decryption is the reverse process. An important ingredient of encryption/decryption process is the idea of key. To decipher an encrypted file, a key is required that was used to encrypt it. There are two types of cryptosystems namely, symmetric and asymmetric algorithms. A prominent symmetric algorithm, commonly in use is the block cipher algorithm. The Block cipher algorithms use two different key schemes viz., DES and AES[2,3].

DES is based on keys containing the position of elements to be manipulated in the input matrix. Though this method was recommended by NIST, the weakness of this method was sufficiently demonstrated in the literature. The use of key schemes in the variants of DES namely 3DES, RC2 and BlowFish, is similar. These schemes were able to overcome some of the shortcomings of the base DES algorithm but not all, namely the vulnerability at attacks and slower in timings. On the other hand, the AES uses binary keys with which a binary operator is defined over the binary input and key matrices. Here the key is a co operand with the input matrix. The permutation possibility of binary key matrix provides this strategy a greater flexibility and computational simplicity. Though a full attack on AES[4] is not found practical the algebraic simplicity makes it vulnerable for attack as the crypt process can be expressed as a series of simple equations that could be solved. The proposed crypt algorithm combines the idea of keys as position pointers and input matrix as co operand in the encryption/decryption process.

The rest of this paper is organized as follows. Related work is described in Section II. A view of the proposed algorithm in section III. This is followed by the experimental results and security analysis in section IV. Finally the concluding notes are introduced in section V.

2. RELATED WORK
In recent years, a number of different image encryption schemes have been proposed in order to overcome image encryption problems. A few image encryption techniques suggested recently are discussed in brief. A new efficient chaotic image stream cipher was suggested by Ismail et al [5] where they used two chaotic logistic maps to confuse the relationship between the cipher image and the plain image with an external secret key of 104-bits size. They also modified the secret key after encryption of each pixel in the plain image. Nayak et al [6] proposed a chaotic image cipher on the basis of index position of the chaotic sequence using logistic map in which permutation are made on the basis of their sorted real value of the sequence.

Wang et al [7] presented an image encryption algorithm based on simple Perceptron and using a high-dimensional chaotic system in order to produce three sets of pseudorandom sequence. Then to generate weight of each neuron of perceptron as well as a set of input signal, a nonlinear strategy is adopted. Sathishkumar and Bagan [8] developed a new algorithm in which pixels are transformed by simple diffusion processes. A logistic map was used to generate a pseudo random bit sequence for each process. Further, chaotic image cipher using two chaotic logistic maps and a secret key of 80-bits was suggested by Chen and Chang [9].

The block cryptographic scheme proposed by Xiang et al [10] use random binary sequences generated from the real-valued chaoticmap. The plaintext block is permuted by a key-dependent shift approach and then encrypted by the classical chaotic masking technique. In this algorithm the binary sequence used for substitution leaks the trajectory of the chaotic map for easy cryptanalysis, and the encryption speed is still slow compared to conventional cryptosystems. Lian et al [11] proposed a block cipher based on the chaotic standard map which is composed of three parts: a confusion process based on chaotic standard map, a diffusion function and a key. Wong et al[12] developed a chaos-based image encryption schemes, which performs permutation and the diffusion stages alternatively. It uses simple sequential add-and-shift operations to introduce diffusion effect in the confusion stage.

The diffusion effect is injected by adding the current pixel value with the previous permuted pixel.

3. METHODOLOGY

In this paper a new encryption algorithm called GKSBC (Generalized Key Scheme Block Cipher) is proposed. In this algorithm, block size of the plain text and key size are flexible. The proposed encryption scheme undergoes encryption, substitution and decryption process. In both encryption and decryption, there is a common process to generate a permutation matrix π, which is nonlinearly generated with the help of random generator. A permutation matrix is an identity matrix with the rows and columns interchanged. It has a single 1 in each row and column; all the other elements are 0. The inverse of a permutation matrix is the same as its transpose, $P^{-1} = P^T$. So, no extra calculation is needed to compute the reciprocal matrix for decryption. This is a valuable property for cryptographic purposes that increases algorithm speed and decreases memory usage. We summarize the process of the proposed encryption algorithm as follows:

1. Generate the permutation key:

If the block size is n × n then choose n^2 values in a random manner. Generate a pair of values (i,j) randomly, 1<=i<=n, 1<=j<=n, such that no pair (i,j) is repeated more than once. Let π denote the collection of values as per the order of generation.

2. Encryption:

Let $p_1,p_2,---,p_m$ be the plain text blocks of size n × n. Firstly, we shift each row to the left by a distance equal to the number of 1's present in the row followed by shifting each column upwards by a distance equal to the number of 1's present in the column of the row shifted matrix. We define the resulting matrix as $E_\pi(p)$. Let E(i,j) be the (i,j)th element of $E_\pi(p)$. Secondly, Change E(i,j) as per the permutation key [4,6] as follows,

For (i,j) ≠ (1,1), (1,n) ,(n,1), (n,n)
Define
E(i,j) = {(i,j),(i-1,j-1), (i-2,j-2) ,...,(i-k,j-k)}
Until (i-k) < 0 or (j-k) < 0
E(i,j) = {(i+1,j+1), (i+2,j+2),…, (i+l,j+l)}
Until (j+1) > n or (i+l) > n
E(i,j) = {(i+l-1,j+l-1), (i+l-2,j+l-2),..., (i+l-r,j+l-r)}
Until (i+l-r) > n or (j +l –r) < 0
E(i,j) = {(i+l-r-1,j+l-r-1),..., (i+l-r-p,j+l-r-p)}
Until (j+l-r-p) < 0 or (i+l-r-p) > n
E(i,j) = {(i-k-1,j-k-1),…,(i-k-q,j-k-q)}
Until (j-k-q) < 0 or (i-k-q) > n
For i = 1 or n
E(i,i) = {(l,l) where l varies from 1 to n}
For j = 1 to n
E(j,j) = {(k,k) where k varies from 1 to n}
If E(i,j)(mod 2) = 0, set E(i,j)= 1 else set E(i,j) = 0.

Let c_1 be the n × n matrix with the modified E(i,j). We define the resulting matrix c_1 as cipher text, corresponding to the plaintext p_1. Similarly we define $c_2,c_3, ---,c_m$

3. Decryption:

Let $c_1, c_2, ---, c_m$ be the cipher text blocks of size n × n bytes. Let E(i,j) be the (i,j) element of the first block c_1. Firstly, Change E(i,j) as per the permutation matrix key as follows in the reverse order. If E(i,j)(mod 2) = 0, set E(i,j)= 1 else set E(i,j) = 0. Let $E_\pi(p)$ be the n × n matrix with the modified E(i,j). Secondly, we shift each column downwards by a distance equal to the number of 1's present in the column followed by right shifting each row by a distance equal to the number of 1's present in the row of the column shifted matrix. We define the resulting matrix p1 as plain text, corresponding to the cipher text c_1. similarly we define $p_2, p_3, ---, p_m$.

Reversibility or the existence of unique decryption for every encryption

For a square of any order the algorithm defines an addition modulo for every element. The process yields an encrypted square matrix of same order. The decrypted square matrix is recovered with the same addition modulo function. If we can prove the addition modulo function helps to recover back the elements, then we may conclude that the process is reliable. This algorithm encrypts/decrypts the elements in an unique order generated at random. This order is reversed in an identical fashion as was determined in that random process, called randomized key. Every encryption stage is retraced in the decryption process. To prove this addition modulo function has an unique inversion it is sufficient to say that every possible case recovers the original elements in the addition modulo function.

In a matrix of zeros and ones any function defined on the elements of that matrix would yield either an odd number or even number. In this algorithm we define a sum of row and column elements for every element in the matrix. This may be defined as two components one the element itself as a component, the other is the remaining elements. Again the sum of the other elements other than the element itself, that belong to the row and column of that element, would either be even or odd. The element itself can either be 0 or 1, as the matrix is a zero and one matrix. Therefore the addition modulo function defined in the algorithm has four possible cases as defined below. This exercise proposes to prove that for all the possible cases the addition modulo function has an unique inverse. This may help to prove the algorithm has an unique inversion process. From the algorithm the addition modulo for different possibilities may be presented as below in the diagram

⊕	Even	Odd
0	1	0
1	0	1

Case (i) element = 0; sum of other elements = even.

The first case presumes a possibility where the sum of all the row and column elements for an element except the element is even and the element itself is 0. The addition modulo defined on the sum of this two components gives 1 as the output. In the inversion process (decryption) we could confront two components in that the element would be 1 and the other component would be even. Now the addition modulo of these two components would give 0. Therefore 0 is encrypted as 1 and again decrypted to 0.

Case (ii) element = 1; sum of other elements = even.

The second case presumes a possibility where the sum of all the row and column elements for an element except the element is even and the element itself is 1. the addition modulo defined on the sum of this two components gives 0 as the output. In the inversion process (decryption) we could confront two components in that the element would be 0 and

the other component would be even. Now the addition modulo of these two components would give 1. Therefore 1 is encrypted as 0 and again decrypted to 1.

Case (iii) element = 0; sum of other elements = odd.

The third case presumes a possibility where the sum of all the row and column elements for an element except the element is odd and the element itself is 0. the addition modulo defined on the sum of this two components gives 0 as the output. In the inversion process (decryption) we could confront two components in that the element would be 0 and the other component would be odd. Now the addition modulo of these two components would give 0. Therefore the encrypted and decrypted value remains 0 in this case.

Case (iv) element = 1; sum of other elements = odd.

The fourth case presumes a possibility where the sum of all the row and column elements for an element except the element is odd and the element itself is 1. The addition modulo defined on the sum of this two components gives 1 as the output. In the inversion process (decryption) we could confront two components in that the element would be 1 and the other component would be odd. Now the addition modulo of these two components would give 1. Therefore the encrypted and decrypted value remains 1 in this case.

4. CRYPTANALYSIS

In this section we performed a series of test to justify the efficiency of the proposed image encryption scheme. The evaluation consist of theoretical derivations and practical experimentation. A good encryption scheme should resist all kinds of known attacks, such as the statistical attack and the differential attack. The security of the proposed cryptosystem is investigated for digital images under the statistical and differential attacks. It will be shown that the proposed cryptosystem is secure from the cryptographic viewpoint. Here, some security analysis results, including the Image Comparison using Histogram, key space analysis, execution time, statistical analysis and differential analysis are presented. Tests are made on the image of Lena, girl and baboon as shown in Fig. 1.

4.1 Image Comparison using Histogram

In statistics, a histogram is a graphical representation showing a visual impression of the distribution of data. An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.

The quality of the algorithm could be assessed with the help of the histogram created from the cipher image[13]. A good algorithm creates a relatively uniform distribution in histogram. Since uniform distribution indicates a thorough diffusion of image and destruction of pattern in image. This ensures the dissimilarity in original and ciphered images. The histogram analysis compares those images in terms of their respective histograms. Figure 1 shows histogram analysis on several images having different contents and sizes using proposed algorithm. The histogram of original images contains large sharp rises followed by sharp declines as shown in figure 1(c). And the histogram of the encrypted images as shown in figure 1(d) has uniform distribution which is significantly different from original image and has no statistical similarity in appearance. Therefore, the proposed algorithm does not provide any clue for statistical attack. Plain-images and cipher-images are shown in figures 1(a) and 1(b), respectively.

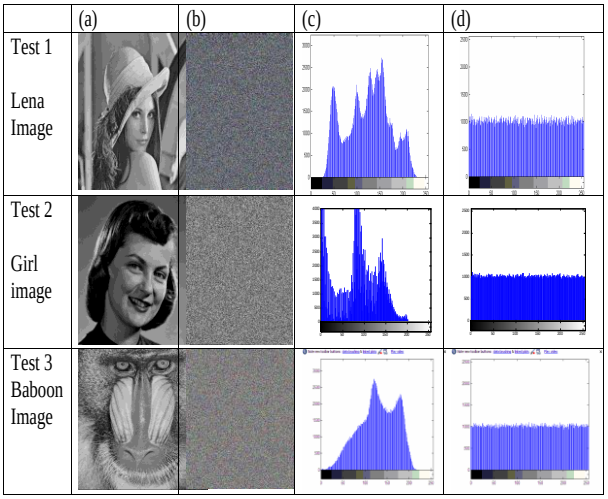

Figure 1. Histogram for three test images : (a) plain-image (b) cipher-image (c) plain-image histogram (d) cipher-image histogram

4.2 Key Sensitivity Test

Key sensitivity test is done to assess the security aspects of the algorithm. A large change in the encrypted images for a slight change in key would indicate high security characteristic of the algorithm [14]. Assume that a 8 × 8 ciphering key is used. This means that the key consists of 64 bits. For testing the key sensitivity of the proposed cryptosystem, we perform the following steps:

1. An image is encrypted using the secret key, which is a pair of values (i,j) generated randomly, 1<=i<=8, 1<=j<=8, such that no pair (i,j) is repeated more than once and the resultant image is referred to as encrypted image A [see Fig. 2(b)].
2. The same image is encrypted by making a slight modification in the secret key (i.e.,) by interchanging the last two pair of values. The resultant image is referred to as encrypted image B [see Fig. 2(c)].
3. Again, the same image is encrypted by making another slight modification in the secret key (i.e.,) by interchanging the first two pair of values. The resultant image is referred to as encrypted image C [see Fig. 2(d)].
4. Finally, the three encrypted images A, B, and C are compared.

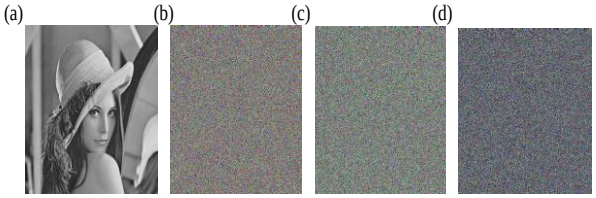

Figure 2. Key sensitivity test of the cryptosystem with (a) Lena image (b) Encrypted image A (c) Encrypted image B and (d) Encrypted image C

It is not easy to compare the encrypted images by simply observing them. Comparing two images for

similarity/dissimilarity would involve a subjective judgment. To make it formal and systematic correlation analysis is used. The correlation coefficient is calculated using the array of pixel wise grey scale measures for original and crypt images. Higher correlation value would indicate high similarity and lesser value would indicate high dissimilarity. This factor demonstrates to what extent the proposed encryption algorithm strongly resists statistical attacks. The correlation coefficients for the three encrypted images A, B, and C are presented in Table 1. It is clear that no correlation exists among the encrypted images even though they have been produced using slightly different secret keys.

Table 1: Results of the key sensitivity test

Image	Correlation coefficient
Encrypted image A	0.017019
Encrypted image B	0.02018
Encrypted image C	0.016248

4.3 Encryption time

The time taken by the algorithm to produce the cipher text from the plain text is called the encryption time. This time measures the speed of encryption. The lower encryption time indicates the efficiency of the algorithm. The results of this test for different files is shown in Table 2 as follows.

Table 2. Comparison of Encryption and Decryption Time for Different File Formats

File Type & Size	Time for Encryption	Time for Decryption
Txt,4kb	82 ms	73 ms
Doc,551kb	1 secs 233 ms	1 secs 123 ms
Bmp,2.25mb	2 secs 859 ms	2secs 703ms
Jpeg,1.69mb	2 secs 156 ms	2 secs 125 ms
Pdf,1.20mb	1 secs 390 ms	1 secs 360 ms
Xls,44kb	94 ms	90 ms
Mp3,7.42mb	8 secs 109 ms	7 secs 953 ms
Vob,56.6mb	58 secs	53 secs

4.4 Randomness Analysis

To verify the randomness of pseudorandom bit sequences, we apply the standard randomness test FIPS 140-2[15]. Any PRBS can be claimed as a good PRBS if passed all the tests specified. For analysis purposes, the randomness test has been performed which consist of four tests. A single bit sequence of 20,000 consecutive bits of output from the generator is subjected to each of the following tests:

4.3.1 The Monobit Test:

In this test, the number of ones is counted and denotes this quantity by x. Considering a bit stream of 20,000 bits in which all bit occurrences can be considered as Bernoulli trials with success probability of ½. Let x denotes the number of ones occurred in n bits, then the number of distinct patterns are

$$ {}_nC_x = \frac{n!}{[(n-x)!x!]} $$

The probability distribution function f(x) describes the probability of x number of ones in n bits. The test is considered pass if 9,654 < x < 10,346. In Table 3, we calculate the number of ones in the 20,000 bit stream generated by our algorithm for monobit test.

4.3.2 The Poker Test:

We divide 20,000 bit sequence into 5,000 contiguous 4 bit segments. The number of occurrences of each of the 16 possible 4 bit values is counted and stored[19]. Denote f(i) as the number of each 4 bit value i where 0 < i < 15. Then, we evaluate the following:

$$ x = \left(\frac{16}{5000}\right) \times \sum_{i=0}^{15} g_i^2 - 5000 \qquad (1) $$

The test is passed if 1.03 < x < 57.4.

4.3.3 The Runs Test:

A run is defined as a maximal sequence of consecutive bits of either all ones for all zeros, which is part of the 20,000 bit stream. The incidences of runs of all lengths should be counted and stored.

4.3.4 The Long Run Test:

A long run is defined to be a run of length 34 or more (of either zeros or ones). On the sample of 20,000 bits, the test is passed if there are no long runs.

Table 3. Randomness analysis for poker and long run test

Comparison	Poker test	Long run test
Theoretical Value	2.16–46.17	>34
Statistic for Lena Image	11.4624	None
Statistic for Girl Image	24.1152	None
Statistic for Baboon Image	11.366	None
Result	Pass	Pass

Table 4: Randomness analysis for Monobit and Runs test

Comparison	Test item	Monobit test	Run test Length of the run					
			1	2	3	4	5	≥6
Theoretical Value		9725–10725	2315–2685	1114–1386	527–723	240–384	103–209	103–209
Statistic for Lena Image	0 bit 1 bit	10019 9981	2531 2491	1158 1242	657 619	312 296	156 146	176 128
Statistic for Girl Image	0 bit 1 bit	9973 10027	2512 2541	1242 1311	618 701	309 287	164 198	159 156
Statistic for Baboon Image	0 bit 1 bit	10095 9005	2456 2523	1234 1267	684 711	289 345	201 158	187 182
Result		Pass	Pass	Pass	Pass	Pass	Pass	Pass

4.5 Differential Analysis

Another test that assesses the security aspect of the proposed algorithm is the differential analysis. This analysis compares the original and ciphered images using the measure over grey scale value of respective objects [16]. The higher values would indicate higher security.

1) Number of Pixels Change Rate (NPCR)

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (2)$$

2) Unified Average Changing Intensity (UACI)

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{C_1(i,j) - C_2(i,j)}{255}\right] \times 100\% \quad (3)$$

where W and H are the width and height of C_1 or C_2 . C_1 and C_2 are two ciphered images, whose corresponding original images have only one pixel difference and also have the same size. The $C_1(i , j)$ and $C_2(i , j)$ are grey-scale values of the pixels at grid (i,j). Then D(i,j) is determined by $C_1 (i , j)$ and $C_2 (i , j)$. If $C_1 (i , j) = C_2 (i , j)$, then, D(i, j) = 1; otherwise, D(i, j) = 0.

Table 5. NPCR and UACI Estimation

Image	NPCR	UACI
Lena	99.76%	33.56%
Girl	99.63%	32.79%
Baboon	99.68%	33.75%

Tests have been performed on the proposed scheme with three different images of size 512 × 512. Results obtained from NPCR show that the encryption scheme is not sensitive to small changes in the input image. In table 5, the UACI estimation result shows that the rate influence due to one pixel change is very low. The results demonstrate that a swiftly change in the original image will result in a negligible change in the ciphered image.

5. CONCLUSION

This paper proposed and analyzed the algorithm based on GKSBC (Generalized Key Scheme Block Cipher), in terms of common crypt analytic criteria namely image comparison using histogram, key sensitiveness, execution time, randomness and differential analysis. We observed that the histogram of the ciphered image is fairly uniform and is significantly different from that of the original image. The security of the cipher image of this scheme is evaluated by the key space analysis. According to FIPS 140-2 randomness tests for the image sequence encrypted by the proposed algorithm have no defect and pass all the four tests. The proposed cryptosystem by differential approach shows the estimated expectations and variance of NPCR and UACI are very close to the theoretical values. The efficiency and security of the proposed encryption scheme makes it an ideal choice in secure media applications where a large amount of multimedia data has to be encrypted/decrypted in real time.

6. REFERENCES

[1] Mitra A et al., "A new image encryption approach using combinational permutation techniques. International Journal of Computer Science", vol. 1, no. 2 , pp. 1306-4428, 2006.

[2] Stallings W, "Cryptography and network security", 4th Ed., prentice hall, PP. 58-309, 2005.

[3] Coppersmith D, "The data encryption standard(DES) and its strength against attacks", IBM Journal of Research and development, pp. 243-250, may 1994.

[4] Ferguson N et al., "Practical Cryptography", John Wiley and sons, 2003.

[5] Ismail et al., "A digital image encryption algorithm based a composition of two chaotic logistic map", International Journal of Network Security, vol. 11, no. 1 , pp. 1-10, 2010.

[6] Nayak et al., "Image encryption using an enhanced block based transformation algorithm", International Journal of Research and Review in Computer Science, vol. 2, no. 2 , pp. 275-279, 2011.

[7] Wang X Y et al., "A chaotic image encryption algorithm based on perceptron model", Nonlinear Dynamics, vol. 62, no. 3, pp. 615–621, 2010.

[8] Sathishkumar G.A et al., "A novel image encryption algorithm using pixel shuffling Base 64 encoding based chaotic block cipher", WSEAS Transactions on computers, vol. 10, no. 6, pp. 169–178, 2011.

[9] Chen et al., "A novel image encryption algorithm based on logistic maps", Advances in Information Science and Service Sciences, vol. 3, no. 7, pp. 364–372, 2011.

[10] Xiang T et al., "A Novel Block Cryptosystem Based on Iterating a Chaotic Map," Phys. Lett. A, vol. 349, pp. 109-115, 2006.

[11] Lian S.G et al., "A Block Cipher Based on a Suitable Use of Chaotic Standard Map," Chaos, Solitons and Fractals, vol. 26, no. 1, pp. 117-129, 2005.

[12] Wong K.W et al., "A Fast Image Encryption Scheme Based on Chaotic Standard Map," Phys. Lett. A, vol. 372, no. 15, pp. 2645-2652, 2008.

[13] Y. B. Mao, G. Chen, and S. G. Lian, "A novel fast image encryption scheme based on the 3D chaotic baker map", Int. J. Bifurcation Chaos Appl. Sci. Eng. PP. 3613–3624 , 2004.

[14] Shatheesh Sam I et al., "Block Cipher Scheme for Image Cryptosystem Using Alternative Chaotic Maps ", European Journal of Scientific Research. PP. 232- – 240, vol. 51, No.2, 2011.

[15] FIPS 140-2, "Security requirements for cryptographic modules", Federal Information Processing Standards publication. May 2001.

[16] E. Biham and A. Shamir, "Differential Cryptanalysis of the Full 16-Round DES," in Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology: Springer-Verlag, 1993.