In terms of basis restriction mean-square-errors[2] for $\rho = 0.95$, MDCTs have almost the same performance as DCT as shown in Table 2.

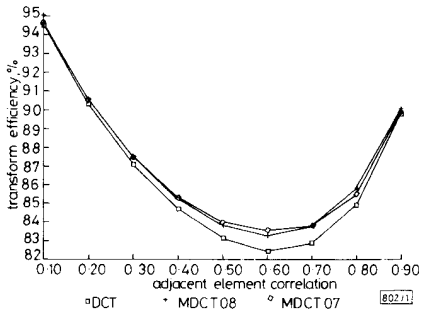

Fig. 1 *Comparisons of transform efficiencies of order-8 MDCTs and DCT for different values of ρ*

Table 2 BASIS RESTRICTION MEAN-SQUARE-ERRORS FOR ADJACENT ELEMENT CORRELATION ρ EQUAL TO 0·95

No. of coefficients retained	KLT	DCT	MDCT08	MDCT07
2	0·1372	0·1381	0·1382	0·1383
6	0·0567	0·0572	0·0573	0·0575
10	0·0406	0·0409	0·0409	0·0409
18	0·0263	0·0264	0·0264	0·0265
26	0·0189	0·0189	0·0189	0·0189
34	0·0136	0·0136	0·0136	0·0136

Concluding remarks: A new algorithm based on the principle of dyadic symmetry and computer search has been proposed for the development of new transforms. These transforms are optimised in terms of transform efficiency for image coding applications. The new transforms obtained have been shown to be better than the DCT in terms of transform efficiency.

S. J. HUANG
S. N. KOH
H. K. TANG

School of Electrical and Electronic Engineering,
Nanyang Technological Institute,
Singapore 2263

19th February 1990

References

1 CHAM, W. K., and CLARKE, R. J.: 'Application of the principle of dyadic symmetry to the generation of orthogonal transforms', *IEE Proc. F*, 1986, **133**, (3), pp. 264–270
2 CHAM, W. K.: 'Development of integer cosine transforms by the principle of dyadic symmetry', *IEE Proc. I, 1989*, **136**, (4), pp. 276–282

RSA BLOCKING AND MULTISIGNATURE SCHEMES WITH NO BIT EXPANSION

Indexing terms: RSA, Blocking, Multisignature

Implementations of the RSA scheme have inherent difficulties related to bit expansion, with consequent difficulties for blocking and multisignatures. Solutions to these latter two problems, which flow naturally from a solution to the root problem of bit expansion are presented.

Rivest, Shamir and Adleman[1] presented the first public key encryption scheme, thereby implementing the idea presented in the seminal paper of Diffie and Hellman.[2] This scheme has sustained twelve years of cryptanalytic attack and is currently the most used, publicly known, public key encryption scheme in the world. It requires that each user obtain as secret key two large primes, p and q, and a value d, relatively prime to $\Phi(n)$, for $n = p * q$, and make public n and the inverse of d, e, modulo $\Phi(n)$. For n of size k bits, user A secretly sends a message unit M, encoded as an integer less than n, to user B by encrypting M to ciphertext $C = M^{e_B} \bmod n_B$, using the public key of user B.

Three practical limitations for implementing this scheme are the following:

(i) *Bit expansion for message blocking:* Since $n < 2^k$, message unit M, $n \leq M \leq 2^k$ cannot be encrypted using this scheme since they will encrypt to the same value as does message $M' = M \bmod n$. As a result, for n with k bits, long messages in practice need to be blocked into message units (blocks) of maximum length, $(k-1)$ bits. Since the corresponding ciphertexts sometimes will be of length k bits, the encryption algorithm needs to be thought of as expanding message units by one bit per block. A recent solution to this problem[3] adapts the RSA scheme to take messages of k bits per block, requiring an average of $n/2^{k-1}$ exponentiations, and leaking information in the process.[4,5]

(ii) *Moduli size clashes for multisignature:* The RSA scheme also allows for applying digital signatures to messages After hashing the message to a message unit, M, of $k-1$ bits, user A applies secret key, d_A, to M, thereby producing signature $S_A = M^{d_A} \bmod n_A$. This encryption requires again an expansion of one bit. This digital signature idea has been expanded to that of applying any number of digital signatures to a document (the multi-signature problem). Since each user who wishes to sign the document has a distinct public key, one cannot assume that the signature, S_A, produced by the first signer A will be less than the modulus value, n_B, of the second signer B if $n_A > n_B$. The problem grows in difficulty as the number of signatories grows. The authors[6] presented a solution which required that the signers sign in the same order as the order of their moduli. Okamoto[7] proposed a scheme which requires decisions to be made at each step, and then that extra bits (of intermediate signatures) be appended to the message in order to identify those decisions so that the final signature can be verified.

(iii) *Moduli clashes for digital signature and secrecy:* The special case in problem (ii) for two signers is equivalent to the problem mentioned in the original RSA paper[1] when a user A wishes to both apply a digital signature to a message unit and send that signed message unit secretly to a user B. To do this requires that both the secret key of user A and the public key of user B be used, specifically

$$C = (M^{d_A} \bmod n_A)^{e_B} \bmod n_B,$$

or alternatively

$$C' = (M^{e_B} \bmod n_B)^{d_A} \bmod n_A$$

Two well-known solutions to this last problem have been available for some time.[1,8] The first uses a threshold scheme, which requires that each user choose two sets of public and secret keys, one with modulus less than a system threshold value and one with modulus greater than that threshold value. The second chooses the one of the two formulas for C given in (iii) above that has the first modulus applied less than the second.

A third solution to this last problem was presented in the original RSA paper,[1] based on a technique of Levine and Brawley[9]:

> 'Each user has a single (e, n) pair where n is between h and $2h$, where h is a threshold (value) . . . A message is encoded as a number less than h and enciphered (as in the ordinary RSA scheme), except that if the ciphertext is greater than h, it is repeatedly re-enciphered until it is less than h. Similarly for decryption the ciphertext is repeatedly deciphered to obtain a value less than h.'

We shall refer to this encryption scheme as 'repeated exponentiation', utilised in a communication system with a system defined threshold value. We now show that this scheme, in such a communication environment, can be used to solve the other two problems listed above as well, requiring no bit expansion, and providing relative efficiency.

We solve the bit expansion problem for blocking messages as follows:

(*a*) Since intermediate values to be 're-encrypted' during repeated exponentiation have k bits, we might as well think of the input and output to the exponentiation function always as k bit numbers, with the original input, and the final output, being the only ones with high-order bit 0. The bit expansion of the algorithm itself, is 1-1 since this encryption scheme maps $(k-1)$bit message units to $(k-1)$bit ciphertexts.

(*b*) Though this encryption scheme is non-deterministic in the number of exponentiations required, that number has a geometric distribution with success parameter, p, of value $2^{k-1}/n$, which ranges in value over the interval (1/2, 1). Hence the expected number of exponentiations required in one encryption ranges over the interval (1, 2). We can take 1·5 as an average value for this number of exponentiations if users randomly choose their moduli values (n) over the interval $(2^{k-1}, 2^k)$.

(*c*) The result is that messages can now be blocked into message units of $k-1$ bits, and each message unit will be encryypted into a $(k-1)$ bit ciphertext. Hence we have removed the bit expansion problem associated with the original RSA scheme.

To solve the multisignature problem, all signers must choose moduli of the same number of bits—say k bits. Both message blocks and signatures will be restricted to being of maximum size $(k-1)$ bits. The order of signature is irrelevant. If there are m signers, the first signer applies the repeated exponentiation technique to the original $(k-1)$ bit message, M, producing a signature, S_1, of $(k-1)$ bits, using this signer's secret key. Then, in succession, signer i $(2 \leq i \leq m)$ receives a $(k-1)$ bit signature, S_{i-1}, from signer $(i-1)$, and applies the repeated exponentiation technique to S_{i-1} to produce $(k-1)$ bit signature S_i, using this signer's secret key. The final $(k-1)$ bit signature S_m constitutes the multisignature for message M. The bit-ratio is exactly $1-1$. The price to achieve this is in the number of exponentiations. Instead of the ideal of m exponentiations being required for applying m signatures, a reasonable approximation to an average number of exponentiations required using this scheme is $1{\cdot}5m$ (assuming each signer's expected number of exponentiations is 1·5).

Two final comments:

(*a*) We present here a tradeoff technique which solves the basic problem of bit expansion, and other problems based on it, while paying a price in time that is constantly being reduced as more and more efficient hardware implementations of exponentiation are produced.

(*b*) This approach is equally applicable to any encryption scheme which uses the modular operation for a product of two primes—in particular, for the authors' efficient Rabin scheme.[10]

T. KIESLER
L. HARN

Computer Science Telecommunications Program,
University of Missouri, Jansas City,
5100 Rockhill Road, Kansas City, MO 64110-2499, USA

26th June 1990

References

1 RIVEST, R. L., SHAMIR, A., and ADELMAN, L.: 'A method for obtaining digital signatures and public-key cryptosystem', *Commun. ACM*, 1978, **21**, (2), pp. 120–126
2 DIFFIE, W., and HELLMAN, M. E.: 'New directions in cryptography', *IEEE Trans.*, 1976, **IT-22**, pp. 644–654
3 SHIMADA, M., and TANAKA, K.: 'Blocking method for RSA cryptosystem without expanding cipher length', *Electron. Lett.*, 1989, **25**, (12), pp. 773–774
4 MITCHELL, C.: 'Comment: Blocking method for RSA cryptosystem without expanding cipher length', *Electron. Lett.*, 1989, **25**, (22), p. 1527
5 SHIMADA, M., and TANAKA, K.: 'Reply: Blocking method for RSA cryptosystem without expanding cipher length', *Electron. Lett.*, 1989, **25**, (12), pp. 1527–1528
6 HARN, L., and KIESLER, T.: 'New scheme for digital multisignatures', *Electron. Lett.*, 1989, **25**, (15), pp. 1002–1003
7 OKAMOTO, T.: 'A digital multisignature scheme using bijective public-key cryptosystems', *ACM Trans. Computer Systems*, 1988, **6**, (8), pp. 432–441
8 KOHNFELDER, L. M.: 'On the signature reblocking problem in public-key cryptography', *Commun. ACM*, 1978, **21**, (2), p. 179
9 LEVINE, J., and BRAWLEY, J. V.: 'Some cryptographic applications of permutuation polynomials', *Cryptologia*, 1977, **1**, pp. 76–92
10 HARN, L., and KIESLER, T.: 'Improved Rabin's scheme with high efficiency', *Electron Lett.*, 1989, **25**, (11), p. 726–728

MINIMUM AREA ANALOGUE-DIGITAL CALIBRATION NETWORK FOR HIGH-RESOLUTION DATA CONVERTORS

Indexing terms: Calibration, Data processing

A novel calibration network where the capacitors are determined with reference to the minimum geometry capacitor unit allowed by the technology and follow a non-binary weighting rule to accommodate the large inaccuracies inherent to such small capacitance values is described. This, together with the rather simple digital cell which is associated with the calibration capacitors, and an efficient interconnect strategy, makes it possible to achieve a minimum area calibration network suitable for high-resolution data convertors employing capacitor arrays.

Introduction: Two alternative types of convertors have become dominant for high-resolution data conversion, namely the sigma-delta and the successive approximation with self-calibration. Sigma-delta convertors attract widespread interest for high performance digital audio, since they minimise the number of critical analogue components and make extensive use of digital signal processing techniques.[1] Self-calibrated successive approximation convertors are still rather competitive for a large number of high resolution data acquisition applications.[2–4] To keep up with the competitiveness of this type of convertor it is necessary to develop new circuit techniques aiming at the reduction of the silicon area required for integration while maintaining the conversion resolution. This is particularly important in the case of the self-calibration network which typically occupies a significant portion of the overall area of the convertor.

In a high-resolution successive approximation analogue-to-digital convertor (ADC), the binary capacitor array is usually segmented into a main-array, for the M most significant bits (MSBs), and a sub-array, for the L least significant bits (LSBs), to significantly reduce the overall capacitance spread and, consequently, the input capacitive load and also the silicon area required for integration.[4] In this type of architecture, schematically illustrated in Fig. 1, the capacitors of the main array usually need to be calibrated to guarantee the required resolution and linearity specifications. This is achieved by means of self-calibrating capacitor arrays whose weights must be as small as $\pm 1/4$ of the LSB of the convertor. In the novel calibration network described in this letter, the capacitor vaues are determined with reference to the minimum geometry capacitor unit allowed by the technology and obey a non-binary weighting rule to accommodate the large inaccuracies inherent to such small capacitance values.[5–7] This, together