by impulse (salt and pepper) noise with occurrence rate ranging from 10 to 90% were also tested, and the PSNR performance is provided in Table 2.

Table 1: PSNR obtained by different filters for corrupted image 'Lena'

Noise percentage	Median (3×3)	Median (5×5)	Fuzzy [2]	ROM [3]	Our MMEM
	PSNR	PSNR	PSNR	PSNR	PSNR
10	34.25	31.23	37.88	38.98	38.60
20	29.58	30.60	34.19	36.55	36.76
30	23.85	29.72	31.19	33.43	35.41
40	19.18	28.21	28.00	29.88	34.32
50	15.28	24.44	24.97	26.04	32.97
60	12.31	19.09	21.66	21.97	31.76
70	9.95	14.16	18.27	18.12	30.29
80	8.07	10.34	14.72	14.00	28.50
90	6.54	7.42	10.30	9.29	26.09

Table 2: PSNR obtained by different filters for corrupted image 'Bridge'

Noise percentage	Median (3×3)	Median (5×5)	Fuzzy [2]	ROM [3]	Our MMEM
	PSNR	PSNR	PSNR	PSNR	PSNR
10	32.85	28.84	36.78	36.46	37.03
20	28.97	28.34	33.82	34.62	35.54
30	23.34	27.57	30.74	31.87	34.20
40	19.00	26.37	27.65	28.71	33.03
50	15.13	23.29	24.24	24.81	31.70
60	12.25	18.63	20.89	21.04	30.24
70	9.87	13.97	17.61	17.41	28.72
80	7.99	10.25	14.37	13.52	26.77
90	6.48	7.33	10.06	9.14	23.97

Conclusions: In this Letter we propose a minimum-maximum exclusive mean (MMEM) filter which is robust for removing impulse noise. Experimental results show that even if the noise is heavy (70%), the proposed filter can still work properly and the restored image is acceptable.

Acknowledgment: This work was supported by the National Science Council, Republic of China, under grant NSC86-2213-E009-108.

© IEE 1997 *12 November 1996*
Electronics Letters Online No: 19970106

Wei-Yu Han and Ja-Chen Lin (*Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China*)

Wei-Yu Han: Corresponding author

E-mail: gis81576@cis.nctu.edu.tw

References

1 GONZALES, R.C., and WOODS, R.E.: 'Digital image processing' (Addison-Wesley, 1992)

2 RUSSO, F., and RAMPONI, G.: 'A fuzzy filter for images corrupted by impulse noise', *IEEE Signal Process. Lett.*, 1996, **3**, (6), pp. 168–170

3 ABREU, E., LIGHTSTONE, M., MITRA, S.K., and ARAKAWA, K.: 'A new efficient approach for the removal of impulse noise from highly corrupted images', *IEEE Trans.*, 1996, **IP-5**, (6), pp. 1012–1025

Digital signature for Diffie-Hellman public keys without using a one-way function

L. Harn

Indexing term: Public key cryptography

The author proposes digital signature schemes without using a one-way function to sign Diffie-Hellman public keys. The advantage of this approach is, instead of relying overall security on either the security of the signature scheme or the security of the one-way function, the security of this proposed scheme is based on the discrete logarithm problem.

Introduction: A one-way function is needed in any digital signature scheme. Without using a secure one-way function, a digital signature can be easily forged [1, 2]. There are some well-known one-way hash functions, such as the MD4, MD5, SHA, etc. There exists a major difference of security assumptions between digital signature schemes and one-way functions. The security assumptions of most signature schemes are based on some well-known computational problems, such as the discrete logarithm problem, the factoring problem, etc. However, the security of most one-way hash functions is based on the complexity of analysing an iterated simple function. Since most computational problems are well-known and easy to understand, the security of most signature schemes can withstand quite a long period of time. However, a one-way function may seem very difficult to analyse at the beginning; but it may turn out to be vulnerable to some special attacks later. Thus, in general, the lifetime of one-way functions is shorter than that of signature schemes. For example, recent advancement of cryptanalysis research has found that MD5 is 'at the edge' of risking successful cryptanalytic attack [3]. There are two motivations of proposing signature schemes without using a one-way function. First, instead of relying overall security on the weaker assumption between the signature scheme and the one-way function, the security of our proposed schemes is based on the discrete logarithm problem. Secondly, the overall security can be easily understood and analysed.

Diffie and Hellman [4] proposed the well-known public-key distribution scheme based on the discrete logarithm problem in 1976 to enable two parties to establish a common secret session key based on their exchanged public keys. But their original scheme can only share one common secret key and did not provide authentication for the exchanged public keys. Since them, several key exchange protocols [5, 6] to allow two parties to share multiple secret session keys have been proposed based on the Diffie-Hellman public-key technique. In general, these protocols utilise a digital signature for each distributed public key to provide authentication. Since Diffie-Hellman's public key is obtained by computing an exponential function over GF(p) and the exponential function itself is a well-known one-way function, we propose signature schemes without using any additional one-way function for signing Diffie-Hellman public keys. In addition, since the Diffie-Hellman public key is a random number, our proposed schemes are not suitable for signing any given message.

Digital signature schemes for Diffie-Hellman public keys: Let p be a large prime and α be a primitive number in GF(p). Each user selects a fixed secret key $x \in [1, p–1]$ and computes a fixed public key $y = \alpha^x \bmod p$, where y is signed by one authority. $\{p, \alpha, y\}$ are the user public information.

A signature scheme uses a fixed secret key to sign a message and a verifier uses a signer's fixed public key to verify the signature of a message. In this proposed signature scheme, the message itself is a random Diffie-Hellman public key $r = \alpha^k \bmod p \in [1, p–1]$ computed by the signer, where k is a secret random integer $k \in [1, p–2]$ privately selected by the signer.

Now, we use the following model to describe the signing process. The signer uses his secret keys, x and k, to compute the signature s which satisfies

$$ax = bk + c \bmod \emptyset(p)$$

where (a, b, c) are parameters selected from values (r, s). The verification equation is determined accordingly as

$$y^a = r^b \alpha^c \bmod p$$

In the following, we will discuss the general form of the above signature equation to satisfy security considerations.
(i) Since x and k are two secret numbers and the verifier does not know these two values, x and k should be treated as in different terms in the above equation. Otherwise, if we combine these two secret parameters together (i.e. for example, if $xk = r+s \bmod \varnothing(p)$, then $y^k = \alpha^{r+s} \bmod p$ or $r^x = \alpha^{r+s} \bmod p$), the verifier cannot verify the signature.
(ii) To claim that s is a signature for the random public key r, the random public key r should be included in the signature equation and can be included in any parameter of (a, b, c).
(iii) To provide a digital signature, s should also be included in any parameter of (a, b, c). Thus, there are four parameters, (x, k, r, s), in the equation.
(iv) For security reasons, c cannot be zero. For example, if $rx = sk \bmod \varnothing(p)$, it is easy to forge a signature for a random public key to satisfy the verification $y^r = r^s \bmod p$. This can be shown by randomly selecting a $u \in [1, p-2]$ and computing $r' = y^u \bmod p$. The forged signature for the random r' is $s' = y^u u^{-1} \bmod p-1$.
(v) For security reasons, r cannot be combined with s. For example, if $x = k + rs \bmod \varnothing(p)$, it is easy to forge a signature for a random public key to satisfy the verification $y = r\alpha^{rs} \bmod p$. This can be shown by randomly selecting an $r' \in [1, p-2]$ and computing $r'' = y\alpha^{-r'} \bmod p$. The forged signature for the random r'' is $s'' = r'r''^{-1} \bmod p-1$.
(vi) The signature equation contains four parameters. Two parameters, (r, s), are public information. But, x is the fixed secret key of the signer and k is a random secret value for each random public key. Since the number of secret parameters is always one larger than the number of linear equations available to the attacker, the signature scheme is secure based on the discussion in the original ElGamal paper. We list all possible signature variations in Table 1.

Table 1: All possible signature variations

	Signature	
	Equation	Verification
(i)	$rx = k+s \bmod \varnothing(p)$	$y^r = r\alpha^s \bmod p$
(ii)	$sx = k+r \bmod \varnothing(p)$	$y^s = r\alpha^r \bmod p$
(iii)	$x = rk+s \bmod \varnothing(p)$	$y = r^r\alpha^s \bmod p$
(iv)	$x = sk+r \bmod \varnothing(p)$	$y = r^s\alpha^r \bmod p$

Discussion:
(i) Among all signature schemes we have listed in Table 1, the signature generation only requires us to solve a linear equation. The signature verification requires two modular exponentiations. In schemes (i) and (iii), the signature s can be solved without computing the inverse. More important than the efficiency is that these signature schemes are not relied on any one-way hash function.
(ii) The techniques used in the DSA [7] and the Schnorr scheme [8] can also be applied to all schemes in the table to shorten the signature and to speed up computation.

Conclusion: We have proposed signature schemes which are especially suitable for signing the Diffie-Hellman public keys. Using these schemes to sign Diffie-Hellman public keys, they do not require any one-way hash function and are very efficient in signature generation and signature verification.

© IEE 1997 *8 October 1996*
Electronics Letters Online No: 19970071

L. Harn (*Department of Computer Networking, University of Missouri, Kansas City, MO 64110, USA*)

References

1 BOYD, C.: 'Comment: New digital signature scheme based on discrete logarithm', *Electron. Lett.*, 1994, **30**, (6), pp. 480–481

2 NYBERG, K.: 'Comment: New digital signature scheme based on discrete logarithm', *Electron. Lett.*, 1994, **30**, (6), pp. 481

3 DOBBERTIN, H.: 'The status of MD5 after a recent attack', *CryptoBytes*, 1996, **2**, (2), pp. 1–6

4 DIFFIE, W., and HELLMAN, M.E.: 'New directions in cryptography', *IEE Trans.*, 1976, **IT-22**, (6), pp. 644–654

5 ARAZI, A.: 'Integrating a key cryptosystem into the digital signature standard', *Electron. Lett.*, 1993, **29**, (11), pp. 966–967

6 NYBERG, K., and RUEPPEL, R.A.: 'Message recovery for signature schemes based on the discrete logarithm problem'. Proc. Eurocrypt '94, May 1994, pp. 175–190

7 'The digital signature standard', *Comm. ACM*, 1992, **35**, (7), pp. 36–40

8 SCHNORR, C.P.: 'Efficient identification and signatures for smart cards'. Advance in Cryptology - CRYPTO'89, Santa Barbara, 20–24 Aug. 1989, (Springer-Verlag), pp. 239–252

Optimum source-to-channel assignment

A.K. Khandani

Indexing terms: Channel coding, Source coding

The problem of the optimum assignment of a set of source symbols to a set of channel symbols is expressed in terms of a quadratic assignment problem (QAP). Numerical examples are presented for the assignment of a scalar quantiser to a binary channel.

Consider a communication system aimed at transmitting a source **S** through a channel **C**. The source **S** has T symbols, $\mathbf{s}_i$, $i = 0, ..., T-1$, where $\mathbf{s}_i$ occurs with probability $P_s(i)$. The distortion between $\mathbf{s}_i, \mathbf{s}_j \in \mathbf{S}$ is equal to $D_s(i,j)$. The channel **C** has T symbols $\mathbf{c}_i$, $i = 0, ..., T-1$. The probability of receiving a channel symbol j conditioned on transmitting a channel symbol i is equal to: $P_c(j|i)$. The objective is to select a one-to-one mapping ξ between the set of the source symbols and the set of the channel symbols to minimise the end-to-end average distortion, namely

$$D_{ave} = \sum_{i=0}^{T-1}\sum_{j=0}^{T-1} P_s(i)P_c[j|\xi(i)]D_s[i,\xi^{-1}(j)]$$

We assign a T dimensional binary vector to each symbol of the source at the channel input. The vector corresponding to the ith symbol is composed of the elements: $[x_{ij}, j = 0, ..., T-1]$. If the ith source symbol is assigned to the lth channel symbol, we set $x_{ij} = 1$ for $j = l$ and $x_{ij} = 0$ for $j \neq l$. Using these notations, the assignmemt problem is formulated as

$$\begin{aligned} &\text{minimise} \sum_{i=0}^{T-1}\sum_{j=0}^{T-1}\sum_{k=0}^{T-1}\sum_{l=0}^{T-1} P_s(i)P_c(l|j)D_s(i,j)x_{ij}x_{kl} \\ &\text{subject to: } x_{ij} \in \{0,1\} \quad i,j = 0,...,T-1 \qquad (1) \\ &\sum_{j=0}^{T-1} x_{ij} = 1 \quad i = 0,...,T-1 \\ &\sum_{i=0}^{T-1} x_{ij} = 1 \quad j = 0,...,T-1 \end{aligned}$$

The optimisation scheme in eqn. 1 is equivalent to a standard problem of discrete optimisation known as a quadratic assignment problem (QAP) [3, 4]. This problem arises in discrete locational problems with mutual interaction between facilities. QAPs are known to be NP-hard and are generally very difficult to solve. The exact solution methods are mainly based on either finding an integer programming formulation for the problem or using the method of the branch and bound. There are also numerous works discussing different heuristic approaches to approximate the optimum solution [3, 4].

Tables 1 and 2 contain numerical results for the optimum assignment of the levels of a scalar Max quantiser [5] to the symbols of a binary channel. The distortion measure is the mean square distance. The corresponding QAP is solved using the branch and bound algorithm. A supplementary technique (known as reduction) is used which allows us to decompose the objective function of the QAP as the sum of a linear term and a quadratic term. The main strategy in computing lower bounds for a QAP (as required in the branch and bound method) is based on minimising the linear term and replacing the quadratic term by a lower bound