The complexity of the proposed algorithm is $O(N_B^2 \times N_P^2)$ since the complexity of the DAG shortest path algorithm for V nodes and E arcs is $O(V \times E)$ [2]. If the distance between p_k and b_l is $> D_{max}$, the arc $v(i, j) \rightarrow v(k, l)$ with $i < k$ can be removed, and as a result the computational amount is greatly reduced.

Experimental results: The proposed scheme was compared to the intra-frame vertex encoding algorithm [1] and CAE algorithm in MPEG4 VM 8.0 for 30 frames of QCIF 'news' sequences. The MPEG4 test segmentation mask was used as the original shape information. The distortion bound for vertex encoding was $\sqrt{2}$. The alpha threshold for CAE was 128, which is the equivalent distortion bound to that of the vertex encoder. In the proposed scheme, the first frame of the sequence was encoded by the intra-frame encoding scheme, and the other frames were encoded by the inter-frame encoding scheme. The total number of bits for the shapes with the proposed scheme was much less than that with the intra-frame vertex encoding scheme and slightly less than that with the CAE algorithm, as shown in Fig. 3. Note that the vertex encoder has several advantages over the CAE for natural looking video approximation, while the improvement is less dramatic.

Conclusion: We have proposed an efficient inter-frame shape encoding scheme. Temporal redundancy of shapes in video sequences is greatly reduced by considering the vertex points of the previous frame to select the vertex points of the current frame. The proposed inter-frame scheme performs better than the conventional intra-frame scheme and CAE algorithm in coding efficiency.

17 April 1998
Electronics Letters Online No: 19980918

Kyeong Joong Kim, Jong-Yeul Suh, Moon Gi Kang and Kyu Tae Park (*Department of Electronic Engineering, Yonsei University, 134 Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea*)

E-mail: mkang@bubble.yonsei.ac.kr

References

1 SCHUSTER, G., and KATSAGGELOS, A.: 'An optimal lossy segmentation encoding scheme'. Proc. Conf. Visual Communications and Image Processing, March 1996, (SPIE), pp. 1050–1061
2 CORMEN, T., and LEISERSON, C.: 'Introduction to algorithms' (McGraw-Hill Book Company, 1991)
3 CHUNG, J., KIM, J., and MOON, J.: 'Shape information reduction based on contour prediction and shape coding type'. ISO/IEC JTCL/SC29/WG11 Document MPEG95/0461, November 1995

Anonymous fingerprinting of electronic information with automatic identification of redistributors

J. Domingo-Ferrer

Fingerprinting is a technique for protecting intellectual ownership of electronic information. Anonymous fingerprinting schemes were recently proposed to allow a seller to fingerprint information sold to a buyer without knowing the identity of the buyer and without the seller seeing the fingerprinted copy. Finding a (redistributed) fingerprinted copy enables the seller to find out and prove to third parties whose copy it was. The authors present the first anonymous fingerprinting scheme in which the help of a registration authority is not required in order to identify a redistributor.

Fingerprinting is a technique for protecting intellectual ownership of electronic information. Given an original item of information, a tuple of l *marks* is probabilistically selected. A mark is a piece of the information item of which two slightly different versions exist. At the moment of selling a copy of the item, the seller selects one of the two versions for each mark; in other words, an l bit word is hidden in the information, where the ith bit indicates which version of the data is being used for the ith mark. Usually, it is assumed that two or more dishonest buyers can only locate and delete marks by comparing their copies (marking assumption, [1]).

Classical fingerprinting schemes [1, 2] are symmetrical in the sense that both the seller and the buyer know the fingerprinted copy. Even if the seller succeeds in identifying a dishonest buyer, the seller's previous knowledge of the fingerprinted copies means that they cannot be used as proof of redistribution in front of third parties. In [3], asymmetric fingerprinting was proposed, whereby only the buyer knows the fingerprinted copy; the drawback of this solution is that the seller knows the buyer's identity even if the buyer is honest. Recently ([4]) the concept of anonymous fingerprinting was introduced; the principle is that the seller does not know the fingerprinted copy or the buyer's identity. On finding a fingerprinted copy, the merchant needs the help of a registration authority to identify the redistributor.

In this Letter, we describe a construction for anonymous fingerprinting in which, on finding a fingerprinted copy, the seller needs no help to identify the dishonest buyer. That is, the role of the authority is limited to buyer registration. In addition, the redistribution fraud can be proven to third parties.

Let p be a large prime such that $q = (p - 1)/2$ is also prime. Let G be a group of order p, and let g be a generator of G such that computing discrete logarithms to the base g is difficult. Assume that both the buyer B and the registration authority R have ElGamal-like public-key pairs ([5]). The buyer's secret key is x_B and his public key is $y_B = g^{x_B}$. The registration authority R uses its secret key to issue certificates which can be verified using R's public key. All public keys are assumed to be known and certified.

Protocol 1 (Registration) :

(i) R chooses a random secret nonce $x_r \in \mathbf{Z}_p$ and sends $y_r = g^{x_r}$ to B.

(ii) B chooses secret random s_1 and s_2 in $\mathbf{Z}_p$ such that $s_1 + s_2 = x_B \pmod p$ and sends $S_1 = y_r^{s_1}$ and $S_2 = y_r^{s_2}$ to R. B convinces R of zero-knowledge of possession of s_1 and s_2. The proof given in [6] for showing possession of discrete logarithms may be used here. The buyer B computes an ElGamal public key $y_2 = g^{s_2} \pmod p$ and sends it to R. In fingerprinting, S_1 will act as a pseudonym.

(iii) R checks that $S_1 S_2 = y_B^{x_r}$ and $y_2^{x_r} = S_2$. R returns to B certificates $Cert(y_r||S_1)$, $Cert(y_r||S_2)$ and the value x_r. The certificates are linked by y_r and state the correctness of S_1 and S_2.

By going through the above registration procedure several times, a buyer can obtain several different pseudonym pairs (S_1, S_2).

Protocol 2 (Fingerprinting) :

(i) B sends y_r, y_2, $[S_1, Cert(y_r||S_1)]$ and *text* to M, where *text* is a string identifying the purchase. B computes an ElGamal signature *sig* on *text* with the secret key s_2; *sig* is not sent to M.

(ii) M verifies the certificate on S_1.

(iii) B and M enter a secure two-party computation ([7]). M's inputs are y_r, *text*, y_2 and *item*, where *item* is the original information to be fingerprinted. B's inputs are x_r, *sig*, S_2 and $Cert(y_r||S_2)$. The computations performed are:

(a) $ver_1 = Verify_1$ (*text*, *sig*, y_2). The signature *sig* on *text* is verified using the public key y_2. The output ver_1 is a Boolean variable only seen by M which is true if and only if the signature verification succeeds.

(b) $ver_2 = Verify_2$ (S_2, $Cert(y_r||S_2)$, x_r, y_r, y_2). First, the certificate on S_2 is verified. Secondly, it is checked that $g^{x_r} = y_r$ and $y_2^{x_r} = S_2$. Thirdly, it is verified that the value of y_r in the certificate on S_2 is the same as the value y_r in the certificate on S_1 previously verified by M. The output ver_2 is a Boolean variable only seen by M which is true if and only if the three aforementioned checks succeed.

(c) $item^* = Fing$ (*item*, *emb*). A classical fingerprinting algorithm is used to embed *emb* into the original information *item*, where

$$emb := text||sig||y_2||x_r||y_r||S_2||Cert(y_r||S_2) \qquad (1)$$

The fingerprinted information $item^*$ is obtained as output and is only seen by B. See the following for requirements on the fingerprinting algorithm.

In the above two-party computation, M obtains outputs first and, unless ver_1 and ver_2 are both true, B does not obtain the output $item^*$.

When redistribution of $item^*$ is detected, *identification* is based on the information extracted from $item^*$ and the purchase record

held by M. Under the marking assumption [1], a collusion-tolerant fingerprinting algorithm is needed; furthermore, it should be possible to extract the proof of redistribution without input from the buyer. Note that if the buyer's secret were needed for identification, this would mean that M does not know who to accuse when a redistribution is detected; asking all buyers to disclose their secrets is not compatible with anonymity of honest buyers. The scheme given in Section 5 of [4] allows the embedding and extraction of relatively large amounts of information and fulfills the above requirements.

On finding a redistributed copy, M extracts *emb*. The extracted information contains the values specified by eqn. 1 and is combined by M with the purchase record $[S_1, Cert(y_r\|S_1)]$ to construct a redistribution proof:

(i) The signature *sig* on the text *text* is verified using y_2.

(ii) The value y_r links the certificates on S_1 and S_2; moreover y_r cannot be altered, since it is part of the certificates.

(iii) The value x_r proves that the owner of the key y_2 is the same as the owner of S_2. This is so because, according to the registration protocol, R only reveals $\log_g y_r = x_r$ to B after B has provided y_2 such that $y_2^{x_r} = S_2$. Now, if the Diffie-Hellman key exchange is secure, B cannot produce a correct y_2 without knowing $\log_g y_2 = s_2 = \log_{y_r} S_2$.

(iv) Finally, to identify a buyer, M raises the public keys of buyers to x_r until a public key y_B is found such that $S_1 S_2 = y_B^{x_r}$. The dishonest buyer B has been identified. Note that, since y_r is certified, x_r cannot be forged by M to unjustly accuse a buyer.

The following two results analyse the security of the proposed construction.

Proposition 1 (Registration security) : Protocol 1 provides buyer authentication without compromising the private key x_B of the buyer.

Proof (sketch): In registration, R sees S_1, S_2, y_2 and two zero-knowledge proofs. The latter leak no information. Without considering the zero-knowledge proofs, R needs no knowledge of x_B to find the values S_1', S_2' and y_2' which are related in the same way as S_1, S_2 and y_1. Take a random s_2', then compute $y_2' = g^{s_2'}$ and $S_2' = y_r^{s_2'}$. Finally, $S_1' = y_B^{x_r}/S_2'$. Now consider the zero-knowledge proofs; imagine that an impersonator not knowing x_B can compute S_1, S_2 such that he can demonstrate possession of $\log_{y_r} S_1$ and $\log_{y_r} S_2$, and $S_1 S_2 = y_r^{x_r}$ holds. Then the impersonator can compute the discrete logarithm x_B. In general, if impersonation is feasible, so is computing discrete logarithms.

Proposition 2 (Buyer anonymity) : An honest buyer who follows Protocol 2 will not be identified if computing discrete logarithms is hard and secure two-party computation is feasible.

Proof (sketch) : In Protocol 2, M sees y_r, y_2, $[S_1, Cert(y_r\|S_1)]$ and his view of a secure two-party computation. Finding y_B would require knowledge of x_r. However, if secure two-party computation is feasible, the only way for M to find x_r is to compute $\log_g y_r$.

Acknowledgment: This work is partly supported by the Spanish CICYT under grant no. TIC95-0903-C02-02.

15 May 1998
Electronics Letters Online No: 19980961

J. Domingo-Ferrer (*Dept. d'Enginyeria Informàtica, Universitat Rovira i Virgili, ETSE-Autovia de Salou, s/n, 43006 Tarragona, Spain*)

E-mail: jdomingo@etse.urv.es

References

1 BONEH, D., and SHAW, J.: 'Collusion-secure fingerprinting for digital data'. Advances in Cryptology- CRYPTO'95, LNCS 963, (Springer-Verlag, Berlin, 1995), pp. 452–465

2 BLAKLEY, G.R., MEADOWS, C., and PURDY, G.B.: 'Fingerprinting long forgiving messages'. Advances in Cryptology- CRYPTO '85, LNCS 218, (Springer-Verlag, Berlin, 1986), pp. 180–189

3 PFITZMANN, B., and SCHUNTER, M.: 'Asymmetric fingerprinting'. Advances in Cryptology-EUROCRYPT'96, LNCS 1070, (Springer-Verlag, Berlin, 1996), pp. 84–95

4 PFITZMANN, B., and WAIDNER, M.: 'Anonymous fingerprinting'. Advances in Cryptology-EUROCRYPT'97, LNCS 1233, (Springer-Verlag, Berlin, 1997), pp. 88–102

5 ELGAMAL, T.: 'A public-key cryptosystem and a signature scheme based on discrete logarithms', *IEEE Trans. Inf. Theory*, 1985, **IT-31**, pp. 469–472

6 CHAUM, D., EVERTSE, J.-H., and VAN DE GRAAF, J.: 'An improved protocol for demostrating possession of discrete logarithms and some generalizations'. Advances in Cryptology-EUROCRYPT'87, LNCS 304, (Springer-Verlag, Berlin, 1988), pp. 127–141

7 CHAUM, D., DAMGAARD, I.B., and VAN DE GRAAF, J.: 'Multiparty computations ensuring privacy of each party's input and correctness of the result'. Advances in Cryptology-CRYPTO'87, LNCS 293, (Springer-Verlag, Berlin, 1988), pp. 87–119

Gray codes and 1D quadratic maps

G. Álvarez, M. Romera, G. Pastor and F. Montoya

The authors show how the symbolic sequences of same period superstable orbits in 1D quadratic maps are ordered according to Gray codes. Next, the Gray ordering number is introduced, in the interval (0, 1), allowing the simultaneous ordering of symbolic sequences of different period superstable orbits. Likewise, it is shown that Gray ordering number manipulation can determine whether or not a given symbolic sequence exists.

Introduction: Gray codes (see Table 1) were proposed by Gray [1] to be used for shaft encoders in 1953. Since then, Gray codes have seen a great variety of uses, ranging from communications [2] to combinational algebra [3] and from genetic algorithms [4] to electronics [5].

In this Letter, we show how the kneading theory of 1D quadratic maps [6] can be alternatively formulated using Gray codes, in a numerical fashion rather than symbolically as has been previously done.

Table 1: 4bit binary reflected Gray code

Order	Binary code	Gray code
01	0000	0000
02	0001	0001
03	0010	0011
04	0011	0010
05	0100	0110
06	0101	0111
07	0110	0101
08	0111	0100
09	1000	1100
10	1001	1101
11	1010	1111
12	1011	1110
13	1100	1010
14	1101	1011
15	1110	1001
16	1111	1000

Table 2: Real Mandelbrot map symbolic sequences of superstable orbits ordered according to c-parameter values

n	Sequence	c-value
1	C	0
2	LC	–1
3	LRC	–1.754877666...
4	LRLC	–1.310702641...
4	LRRC	–1.940799806...
5	LRLLC	–1.625413725...
5	LRRLC	–1.860782522...
5	LRRRC	–1.985424253...