$\leq A_l(t_1, t_2)+K_l(t_1)-K_l(t_2)$. Finally, by writing explicitly the expression for $A_l(t_1, t_2)$, the proof of the theorem becomes straightforward.

End-to-end delay bound guarantee: Denote by d_l^n the end-to-end delay experienced by the nth packet from stream l. Let I be the number of nodes on the path of flow l, and C_i, $i = 1, ..., I$ be the server capacity at node i. Define $\mathbf{N}_i$ as the set of streams served by node i. Let MAX_l be the maximum packet size from stream l, and MAX_l^n be the maximum packet size from stream l up to packet n (i.e. $MAX_l^n = \max_{k \in \{1, ..., n\}} L_l^k$, where L_l^k is the length of the kth packet from stream l).

Theorem 2: For stream l which is constrained by a (σ_l, ρ_l)-leaky bucket, if the scheduling algorithm at each server on the path of the flow belongs to CBFQ, then the end-to-end delay bound for packet n of stream l is given by

$$d_l^n \leq \frac{\sigma_l + (I-1)MAX_l^n}{\rho_l} + \sum_{i=1}^{I} \sum_{m \in \mathbf{N}_i \backslash \{l\}} \frac{MAX_m}{C_i} + \sum_{i=1}^{I-1} \tau_i$$

where τ_i is the propagation delay between nodes i and i+1.

Proof of theorem 2: By using the hints given in the proof of theorem 1 with the framework provided in [3], this delay bound can be proved easily.

Complexity of CBFQ: It is easy to prove that after serving a packet, the order of the terms in step 4 of the algorithm above does not change except for the queue that has just been served. In this case the sorting is not a full sort but just an insertion of an element (either the queue that has just been served, or a queue that just became active) in an already sorted list. Besides, since the counters are independent, the update operation in step (v) can take place in parallel making the complexity of step (v) equal to $O(1)$. With these remarks and all the other operations being $O(1)$, the complexity of the algorithm becomes the same as the complexity of a search algorithm: $O(\log(J))$. Compared to the alternative approaches, this constitutes a major advancement towards practical implementations. In addition, since the values of the counters are bounded, numerical overflow problems would not occur in our algorithm.

Conclusion: A new scheduling algorithm CBFQ for packet-switched networks has been proposed. Based on a set of counters that keep track of the credits earned by each traffic stream, CBFQ decides which stream is to be served next. The CBFQ algorithm achieves the same fairness and delay bounds as the alternative algorithms such as SCFQ while having a lower computational overhead and requiring hardware for practical implementation.

16 January 1997

Electronics Letters Online No: 19970409

K.T. Chan, B. Bensaou and D.H.K. Tsang (*Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*)

E-mail: eeking@ee.ust.hk

References

1 GOLESTANI, S.J.: 'A self-clocked fair queueing scheme for broadband applications'. Proc. IEEE INFOCOM, 1994, pp. 636–646

2 TSANG, D.H.K., BENSAOU, B., and CHAN, K.T.: 'Credit-based fair queueing for ATM networks', *Electron. Lett.*, 1996, **32**, (25), pp. 2306–2307

3 GOYAL, P., LAM, S.S., and VIN, H.M.: 'Determining end-to-end delay bounds in heterogeneous networks'. Proc. 5th Int. Workshop on Network and Operating System Support For Digital Audio and Video, 1995, pp. 287–298

Cryptanalysis of 'nonlinear-parity circuits' proposed at Crypto '90

A.M. Youssef and S.E. Tavares

Indexing term: Cryptography

In [1] Koyama and Terada proposed a family of cryptographic functions for application to symmetric block ciphers. The authors show that this family of circuits is affine over $GF(2)$. More explicitly, for any specific key K, the ciphertext Y is related to the plaintext X by the simple affine relation $Y = M_K X \oplus d_K$ where M_K is an $n \times n$ non singular binary matrix and d_K is an $n \times 1$ binary vector where n is the block length of the cipher. This renders this family of ciphers completely insecure as it can be broken with only $n + 1$ linearly independent plaintext blocks and their corresponding ciphertext blocks.

Definitions and cipher description:
Definition 1: A function f: $\{0, 1\}^n \rightarrow \{0, 1\}$ is defined to be an affine function if $f(x_1, x_2, ..., x_n) = a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n \oplus b$ for constants a_i, $b \in GF(2)$.

The cipher proposed in [1] is described by the authors as follows:

Definition 2 [1]: A parity circuit layer of length n, or simply an $L(n)$ circuit layer, is a Boolean device with an n-bit input and an n-bit output, characterised by a key that is a sequence of n symbols from $\{0, 1, -, +\}$.

Definition 3 [1]: Function $B = f(K, A)$, as computed by an $L(n)$ circuit layer with key $K = k_1k_2 ... k_n \in \{0, 1, -, +\}^n$ is the relation from an n-bit input sequence $A = a_1a_2 ... a_n \in \{0, 1\}^n$ to an n-bit output sequence $B = b_1b_2 ... b_n \in \{0, 1\}^n$ defined below. An $L(n)$ circuit layer first computes a variable $T \in \{0, 1\}$ such that:

$$T = \bigoplus_{j=1}^{n} t_j \tag{1}$$

where

$$t_j = \begin{cases} 1 & \text{if } (k_j = 0 \text{ and } a_j = 0) \text{ or } (k_j = 1 \text{ and } a_j = 1) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Output $B = b_1b_2 ... b_n$ of the circuit layer is then

$$b_j = \begin{cases} \overline{a_j} & \text{if } \begin{cases} k_j = - \text{ and } T = 1 \\ \text{or } k_j = + \text{ and } T = 0 \\ \text{or } k_j = 1 \end{cases} \\ a_j & \text{otherwise} \end{cases} \tag{3}$$

The proposed cipher is obtained by composing the parity circuit layers as follows:

Definition 4 [1]: A parity circuit of width n and depth d, or simply a $C(n, d)$ circuit, is a matrix of d $L(n)$ circuit layers with keys denoted by $K = K_1||K_2||...||K_d$ for which the n output bits of the $(i-1)$-th circuit layer are the n input bits for the i-th circuit layer for $2 \leq i \leq d$. The key for the $C(n, d)$ circuit is a $d \times n$ matrix whose d lines contain the circuit layer keys.

An example (given in [1]) of $C(n,d)$ with n = 10, and d = 3 is shown in Table 1.

Table 1: $C(n,d)$ with n = 10, and d = 3

Input	1	0	1	1	0	0	1	0	0	1
K1	–	0	1	–	+	+	1	1	–	+
	0	0	0	0	0	0	0	1	1	1
K2	+	1	0	1	1	+	0	–	+	–
	1	2	0	1	1	1	0	1	0	1
K3	–	0	1	+	+	0	–	+	+	–
Output	1	1	1	0	0	1	0	0	1	1

Main result: The Lemma below follows from the fact that the set of affine functions is closed under the XOR operation.

Lemma 1: Any combinational Boolean function that can be realised using only XOR gates is an affine function over $GF(2)$.

In [1], the authors presented many cryptographic properties of these $C(n, d)$ circuits. They also claimed, without proof, that the order of the Boolean canonical form of $C(n, d)$, defined to be the maximum of the order of its product terms, increases exponentially as n or d increases, and hence it is practically infeasible to cryptanalyse $C(n, d)$ if $n \geq 64$ and $d \geq 8$. The following theorem shows that this is not true.

Theorem 1: The output ciphertext, Y, of the parity check circuit $C(n, d)$ is related to the plaintext input, X, by the following affine relation:

$$Y = M_K X \oplus d_K \qquad (4)$$

where M_K is an $n \times n$ non-singular binary matrix and d_K is an $n \times 1$ binary vector.

Proof: Eqns. 2 and 3 can be expressed as

$$t_j = \begin{cases} a_j \oplus k_j \oplus 1 & k_j = 0 \\ a_j \oplus k_j \oplus 1 & k_j = 1 \\ 0 & k_j = + \\ 0 & k_j = - \end{cases} \qquad (5)$$

$$b_j = \begin{cases} a_j & k_j = 0 \\ a_j \oplus 1 & k_j = 1 \\ a_j \oplus T \oplus 1 & k_j = + \\ a_j \oplus T & k_j = - \end{cases} \qquad (6)$$

Hence for any fixed key, $K = K_1 \| K_2 \| \ldots \| K_d$, eqns. 1, 2 and 3, and consequently the parity circuit, $C(n, d)$, can be realised using only XOR gates. Hence every output component of the parity circuit is an affine function of the input variables over $GF(2)$. Since $C(n, d)$ is a bijective mapping [1], then the matrix M_K is non-singular. □

Example: For the cipher described by Table 1, the ciphertext, Y, in relation to the plaintext input, X, is given by the following equation:

$$Y = \begin{bmatrix} 1&1&1&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&1&0&1&1&0&0&0 \\ 0&0&0&1&1&0&0&1&0&0 \\ 0&1&1&1&1&1&1&1&0&0 \\ 0&0&0&0&0&1&1&0&0&0 \\ 0&1&1&0&0&0&0&0&1&0 \\ 0&1&1&0&0&0&0&0&0&1 \end{bmatrix} X \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \qquad (7)$$

By noting that affine functions cannot satisfy the strict avalanche criterion (SAC) [2] then the $C(n, d)$ circuits can never satisfy the SAC as was falsely claimed by Lemma 10 in [1].

Conclusion: The private key cryptosystem proposed in [1] is affine over $GF(2)$ and hence it is completely insecure.

21 February 1997

Electronics Letters Online No: 19970387

A.M. Youssef and S.E. Tavares (*Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, K7L 3N6, Canada*)

E-mail: tavares@ee.queensu.ca

References

1 KOYAMA, K., and TERADA, R.: 'Nonlinear parity circuits and their cryptographic applications'. Advances in Cryptology: Proc. CRYPTO'90, (Springer-Verlag, 1991), pp. 582–599

2 WEBSTER, A.F., and TAVARES, S.E.: 'On the design of S-boxes'. Advances in Cryptology: Proc. CRYPTO'85, (Springer-Verlag, 1986), pp. 523–534

Interferometry with Faraday mirrors for quantum cryptography

H. Zbinden, J.D. Gautier, N. Gisin, B. Huttner, A. Muller and W. Tittel

Indexing term: Quantum cryptography

Quantum cryptography over 23km of installed telecommunications fibre using a novel interferometer with Faraday mirrors is presented. The interferometer needs no alignment nor polarisation control and features 99.8% fringe visibility. A secret key of 20kbit length with an error rate of 1.35% for 0.1 photon per pulse was produced.

In cryptography, safety is normally obtained by exchanging a secret key between the two users, Alice and Bob. In quantum cryptography (QC) the key is exchanged through a quantum channel. Its security is based on the fact that any measurement of a quantum system will inevitably modify the state of this system. Therefore an eavesdropper, Eve, might obtain information out of a quantum channel by performing a measurement, but the legitimate users will detect her and hence not use the key. In practice, the quantum system is a single photon propagating through an optical fibre, and the key can be encoded by its polarisation or by its phase, as first proposed by Bennett and Brassard [1]. In 1992, quantum cryptography was for the first time experimentally demonstrated over 30cm in air with polarised photons [2]. Since then, several groups have presented realisations of both the polarisation [3] and the phase coding scheme in optical fibres over lengths of up to 30km [4, 5].

However, all quantum cryptography systems face two main difficulties. The first problem is the need for continuous alignment of the system. In polarisation-based systems, the polarisation needs to be maintained stable over tens of kilometres, in order to keep Alice and Bob's polarisers aligned. In interferometric systems, usually based on two unbalanced Mach-Zehnder interferometers, one interferometer has to be adjusted to the other every few seconds to compensate for thermal drifts [4]. The second problem is the high noise of photon counters at 1300nm which essentially determines the error rate (ER) of the key.

In this Letter, we present the creation of a secret key over 23 km of installed telecommunications fibre using a recently introduced interferometric system with Faraday mirrors [6]. This phase-coding setup needs no alignment of the interferometer, nor polarisation control. It features excellent fringe visibility and stability. Moreover, we show that the performance of Ge-APD photon counters can be considerably improved using fast active biasing electronics.

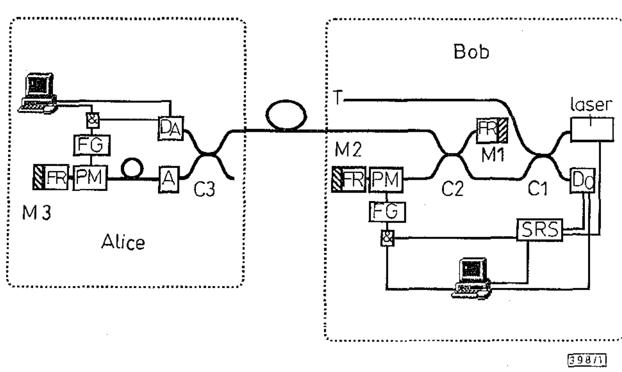

Fig. 1 *Schematic diagram of interferometric QC system with Faraday mirrors*

Our novel quantum cryptography scheme is shown in Fig. 1. In principle, we have an unbalanced Michelson interferometer at Bob's (beamsplitter C2) with one long arm going to Alice. The laser pulse impinging on C2 is split in two pulses, P1 and P2. P2 propagates through the short arm first (mirror M2 then M1) and then travels to Alice and back, whereas P1 is going to Alice first and passes through the short arm on its way back. Both pulses follow exactly the same path and will interfere at C2. To encode their bits, Alice is acting with her phase modulator (PM) only on