# CÓDIGOS PRODUTO MULTINÍVEIS PARA CORREÇÃO DE ERROS BI-SEPARÁVEIS COM APLICAÇÕES EM CRIPTOGRAFIA

**Ricardo M. C. de Souza e J. C. de Souza**

Grupo de Pesquisas em Comunicações - CODEC

Departamento de Eletrônica e Sistemas - UFPE

E-mail Ricardo@npd.ufpe.br CP 7800 50732-970 Recife PE

**Sumário** - Neste trabalho um novo cripto-sistema de chave privada que utiliza códigos produto multiníveis é apresentado. O sistema se diferencia de outros sistemas de cifragem que empregam codificação de canal, em dois aspectos, a saber (1) por explorar a existência de códigos com uma capacidade de correção de erros bi-separáveis muito maior que a de correção de erros aleatórios e (2) por fazer uso de uma estrutura de configurações de erro que não é encontrada em um canal de comunicação real. Aspectos de codificação e decodificação são apresentados e a capacidade do novo cripto-sistema de resistir aos principais ataques que são tipicamente usados contra métodos de cifragem baseados em códigos corretores de erros é discutida.

**Abstract** - This paper describes a new private-key cryptosystem based on multilevel product codes, which are single random-error-correcting and have the capacity for correcting bi-separable errors of arbitrary weight . The encryption process makes use of an error structure that is not found in any real communication channel and that leads to a superior performance in comparison with similar schemes. Aspects of the encoding and decoding operations are discussed and the capacity of the cryptosystem to resist the attacks that are typically used against encryption schemes based on error control codes is analysed.

Palavra chave: Criptografia, Chave Privada, Codificação de Canal, Códigos Produto.

## 1. INTRODUÇÃO

O primeiro cripto-sistema baseado em códigos corretores de erros foi proposto por McEliece, em 1978 [1]. Tratava-se de um cripto-sistema de chave pública que empregava códigos Goppa e cuja concepção está fundamentada no fato de que o problema geral de decodificação de um código linear é NP-completo. Em 1989, Rao e Nam introduziram uma versão de chave privada do cripto-sistema de McEliece, que permitia o uso de códigos mais simples [2]. Desde então, vários outros cripto-sistemas baseados em códigos corretores de erros foram propostos [3], [4], [5], [6]. As idéias apresentadas em [5] representam uma nova abordagem para o assunto, no sentido de que o uso de códigos corretores de erros em surto foi introduzido. Nesse contexto, a segurança do cripto-sistema não está na complexidade computacional de decodificar um código linear e sim na dificuldade de se corrigir um número de erros que está além da capacidade de correção de um dado código. Em [6], estas idéias foram estendidas e um novo cripto-sistema de chave privada foi proposto.

Neste artigo, o uso de códigos produto, definidos em GF(q), para fins criptográficos, é investigado. Trata-se de códigos que tem uma capacidade de correção de erros aleatórios de apenas um erro por bloco e cuja capacidade de correção de erros bi-separáveis é uma função de q e das dimensões do código, podendo assim assumir um valor arbitrário.

Na próxima seção alguns fatos básicos sobre o uso de códigos corretores de erros em surto para fins criptográficos são revistos. Na seção III o conceito de erro bi-separável é introduzido e uma classe de códigos para correção de tais erros é construída. A descrição de um novo cripto-sistema de chave privada baseado nestes códigos é feita na seção IV. As conclusões relativas à pesquisa relatada neste trabalho são apresentadas na seção V.

## 2. PRELIMINARES

Denota-se por $B(n,k,d,b)$ um código de bloco linear corretor de erros em surto, de comprimento $n$, dimensão $k$, distancia de Hamming mínima $d$, capaz de corrigir surtos simples de comprimentos até $b$. Um surto de comprimento $\lambda$ e peso de Hamming $w$ é uma $n$-upla cujas $w$ componentes não nulas estão confinadas a $\lambda$ posições consecutivas, a primeira e a

última das quais são não nulas. Supõe-se que $w_{min} \leq w \leq \lambda \leq b$, onde $w_{min}$ é um número fixo maior que $t$, $b$ $t$ e $d$ $2t - 1$, isto é, $t$ é a capacidade de correção de erros aleatórios do código.

Para cifrar informação através de $B(n,k,d,b)$ procede-se da seguinte maneira. Considerando por texto claro a $k$-upla $m = (m_1, m_2, \ldots\ldots, m_k)$, o texto cifrado $c = (c_1, c_2, \ldots\ldots, c_k, \ldots\ldots, c_n)$ pode ser obtido a partir de $c = (mG + e_{\lambda,w})P$, onde $G$ é a matriz geradora de $B(n,k,d,b)$, $e_{\lambda,w}$ é um surto de comprimento $\lambda$ e peso $w$, gerado aleatóriamente no transmissor e $P$ é uma matriz de permutação $n \times n$. A operação de decifragem é executada em duas etapas. Primeiramente, a permutação $P$ é removida do texto cifrado computando-se $c' = c\,P^{-1} = mG - e_{\lambda,w}$. Em seguida $\boldsymbol{m}$ é recuperado a partir de $c'$ através de algum algorítmo para correção de surtos, uma vez que $c'$ é uma palavra código de $B(n,k,d,b)$ corrompida pelo surto $e_{\lambda,w}$. Nesse contexto, as matrizes $G$ e $P$ definem a chave privada do sistema.

Existem essencialmente duas estratégias para atacar um cripto-sistema de chave privada que emprega códigos corretores de erros em surto, a saber (1) aplicar um ataque por texto claro escolhido para encontrar $G' = GP$ e então determinar $G$ e $P$ a partir de $G'$ e (2) recuperar $\boldsymbol{m}$ a partir de $c$ sem o conhecimento da chave. A capacidade do cripto-sistema de resistir a esses ataques está relacionada com o número de códigos que são combinatorialmente equivalentes a $B(n,k,d,b)$ e com o número total de configurações de erros que o código é capaz de corrigir.

# 3. CÓDIGOS PARA CORREÇÀO DE ERROS BI-SEPARÁVEIS

Cripto-sistemas baseados em códigos para controle de erros atuam modificando a estrutura de um dado código, de modo a preservar apenas sua linearidade, tornando-o incapaz de corrigir os erros que são gerados de forma aleatória no transmissor. Na busca por configurações de erros e códigos que se adequem a esta idéia, a escolha por surtos e códigos corretores de erros em surto é natural. Entretanto, é importante observar que a estrutura de erro a ser considerada não precisa necessariamente existir em um canal real de comunicação, uma vez que, para fins criptográficos, eles são produzidos artificialmente como parte do processo de cifragem.

A seguir, um novo tipo de configuração de erro é definido e uma classe de códigos de bloco lineares capaz de corrigí-los é construída.

**Definição 1** - O mapeamento direto de parâmetros $r$ e $s$, denotado por MD($r$,$s$), associado ao vetor $v = (v_1, v_2, \ldots\ldots, v_{rs})$ é aquele que mapeia as $rs$ componentes de $v$ na matriz

$$V = \begin{bmatrix} v_1 & v_2 & \cdots & v_s \\ v_{s+1} & v_{s+2} & \cdots & v_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ v_{(r-1)s+1} & v_{(r-1)s+2} & \cdots & v_{rs} \end{bmatrix}$$

e o mapeamento direto inverso de parâmetros $r$ e $s$ (MDI($r$,$s$)) é o que mapeia a matriz $V$ no vetor $v$.

**Definição 2** - Seja $e = (e_1, e_2, \ldots. e_{rs})$ um vetor de comprimento $rs$ cujas componentes são elementos de $GF(q)$. O vetor $e$ será um erro bi-separável $r$ x $s$ *sobre* $GF(q)$, denotado por EBS($r$,$s$), se satisfizer às condições

1. As suas componentes não nulas são elementos distintos de $GF(q)$.

2. Quando suas componentes forem mapeadas em uma matriz $r$ x $s$ usando o MD($r$,$s$), cada linha e cada coluna da matriz contenha, no máximo, uma componente não nula.

Da primeira condição na definição 2, vê-se que o número máximo de componentes não nulas de um EBS($r$,$s$) com componentes em $GF(q)$ é $q$-$1$. Da segunda condição, vê-se que o mesmo pode ter, no máximo, min ($r$,$s$) componentes não nulas. Portanto, o peso máximo de um EBS($r$,$s$) sobre $GF(q)$ é $w_{máx}$ = min ($q$-$1$, min ($r$,$s$)). O número de tais erros de peso $w$ é

$$N_{EBS(r,s)}(w) = \binom{q-1}{w} \prod_{i=1}^{w} (r+1-i)(s+1-i) \tag{1}$$

$w = w_{min}, \ldots, w_{max}$. Um código produto sobre $GF(q)$ capaz de corrigir um erro aleatório por bloco $(t = 1)$ e capaz de corrigir EBS$(r+1, s-1)$'s de peso $\leq w_{máx}$ é mostrado na figura 1. Os códigos das linhas e das colunas são códigos de um único dígito de paridade, $C_1(N_1, K_1, D_1)$ e $C_2(N_2, K_2, D_2)$ respectivamente, cujos parâmetros são $N_1 = s - 1$, $K_1 = s$, $D_1 = 2$ e $N_2 = r + 1$, $K_2 = r$, $D_2 = 2$. Temos, portanto, $r$ símbolos de paridade das linhas, $s$ símbolos de paridade das colunas e um símbolo de paridade sobre as paridades.

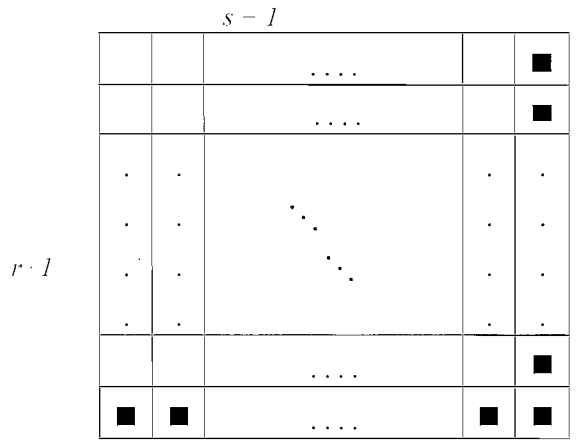


*Figura 1. Código produto de dimensão rs e comprimento (r−1)(s−1)*

código produto resultante $C(n, k, d)$ tem comprimento $n = (r-1)(s-1)$, dimensão $k = rs$, distância mínima $d = 4$ e taxa

$$R = \frac{rs}{(r+1)(s+1)} \tag{2}$$

Uma palavra código de $C$ é uma matriz $V((r-1)\,x(s-1))$. Para transformá-la em um vetor de comprimento $(r+1)(s+1)$, utiliza-se o MDI $(r-1, s-1)$. Os algoritmos de codificação e decodificação são descritos a seguir.

## Codificação

Para codificar uma mensagem **m**, que é um vetor de comprimento $k = rs$ , os bits são colocados nas posições de informação do arranjo em alguma ordem. Em seguida, calculam-se as paridades $(p_1, p_2, \ldots, p_{s+r}, p_{s+r+1})$ e a matriz resultante é transformada em um vetor de comprimento $(r-1)(s-1)$ utilizando-se o MDI$(r + 1, s + 1)$ (figura 2)

| | | | | |
|---|---|---|---|---|
| $m_{i_1}$ | $m_{i_2}$ | .... | $m_{i_s}$ | $p_1$ |
| $m_{i_{s+1}}$ | $m_{i_{s+2}}$ | .... | $m_{i_{2s}}$ | $p_2$ |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| $m_{i_{(r-1)s+1}}$ | $m_{i_{(r-1)s+2}}$ | .... | $m_{i_{rs}}$ | $p_r$ |
| $p_{s+1}$ | $p_{s+2}$ | .... | $p_{s+r}$ | $p_{s+r+1}$ |

Figura 2. Disposição dos rs bits de mensagem e dos $s - r + 1$ bits de paridade no arranjo

## Decodificação

Na decodificação, os bits do vetor recebido $r$ são recolocados no arranjo utilizando-se o MD($r + 1, s + 1$), e os vetores síndrome vertical e horizontal, respectivamente $v = (v_1, v_2, ..., v_{s+1})$ e $h = (h_1, h_2, ..., h_{r+1})$ são calculados (figura (3).

Supondo que um EBS($r-1, s-1$) $e$ de peso $w \leq w_{máx}$ tenha ocorrido, denota-se por $a_1, a_2, ..., a_w$ os valores das componentes não nulas de $e$. Pela definição de EBS, quando $e$ for mapeado usando o MD($r + 1, s + 1$), no arranjo, cada uma das suas $w$ componentes não nulas ficará em uma linha diferente. Assim, a síndrome horizontal terá $w$ componentes não nulas, cujos valores são $a_1, a_2, ..., a_w$. Da mesma forma, a síndrome vertical terá peso $w$ e os mesmos valores para as componentes não nulas. Assim, vê-se que a posição do arranjo correspondente à componente de $e$ de valor $a_x$ será a posição $(i, j)$, onde $i$ e $j$ são tais que $h_i = a_x$ e $v_j = a_x$, $x = 1, 2, ..., w$. O erro é corrigido e os bits de mensagem são retirados na mesma ordem em que foram colocados. Neste caso, a complexidade dos algorítmos de codificação e decodificação será de $O(k)$ somas em $GF(q)$.

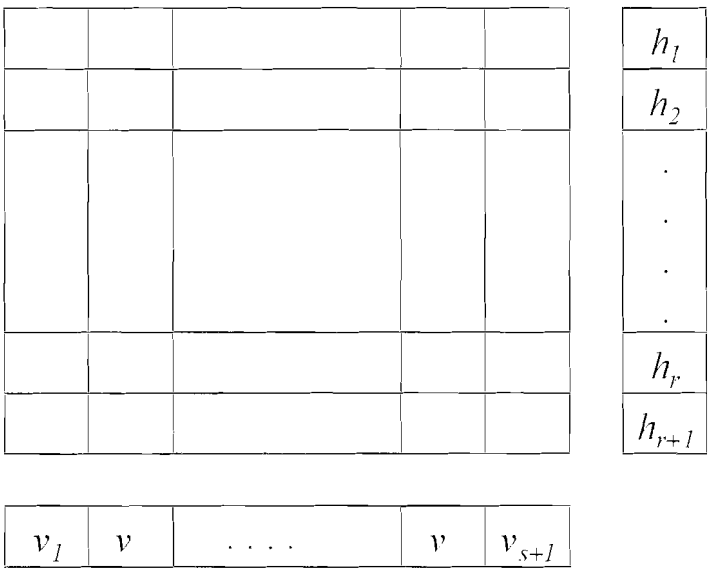


*Figura 3. Síndromes vertical e horizontal*

O número de códigos nessa classe pode ser calculado observando-se que cada ordem de colocação dos $k$ bits de mensagem no arranjo define uma matriz geradora diferente e, portanto, um código diferente. Cada uma dessas ordens de colocação diferentes pode ser obtida, simplesmente, permutando-se o vetor de mensagem $\boldsymbol{m}$ de comprimento $k = rs$ e, em seguida, utilizando o MD ($r$,$s$) para mapeá-lo na submatriz das posições de informação do arranjo. Observando também que a linha e a coluna de paridade não precisam ser as indicadas na figura 1, concluímos que o número de códigos na classe é

$$(r+1)(s+1)(rs)! \tag{3}$$

Como, independente da ordem em que os bits de mensagem são colocados e das posições da linha e da coluna de paridade, as palavras são as mesmas a menos de uma permutação, todos esses $(r-1)\,(s-1)\,(rs)!$ códigos são equivalentes. Portanto, cada matriz pode ser representada por uma permutação de ordem $k$, e dois inteiros, um entre $1$ e $r$ e outro entre $1$ e $s$, indicando as posições da linha e da coluna de paridade.

Para determinar o número $N_B$ de códigos combinatorialmente equivalentes a um dado código da classe, é preciso estabelecer de quantas maneiras os bits de mensagem podem ser colocados no arranjo de modo a preservar as equações de paridade. Observando que permutar colunas, ou linhas, das posições de mensagem do arranjo preserva as equações de paridade e que qualquer outra modificação nos bits de mensagem gera equações de paridade diferentes, concluímos que

$$N_B = (r+1)!(s+1)! \tag{4}$$

# 4. DESCRIÇÃO DO CRIPTO-SISTEMA

A cifragem de chave privada baseada na classe de códigos introduzida na seção anterior, consiste em obter o texto cifrado $c$ a partir de

$$c = (mG - e_{\lambda,w})P \tag{5}$$

onde $e_{\lambda,w}$ é um erro bi-separável aleatório de comprimento $\lambda$ e peso $w$, $P$ é uma matriz de permutação $n \times n$ e $G$ é a matriz geradora $k \times n$ do código utilizado. O texto claro $\boldsymbol{m}$ e o texto cifrado $c$ são, respectivamente, vetores linha de comprimento $k$ e $n$. A decifragem de $c$ requer sua pré-multiplicação pelo inverso da matriz $P$, após o que o texto claro $\boldsymbol{m}$ pode ser recuperado através do algoritmo de decodificação descrito anteriormente. As técnicas de criptoanálise do sistema partem do princípio de que o texto cifrado pode ser reescrito na forma

$$c = mG' + e'_{\lambda,w} \tag{6}$$

onde $G' = GP$ e $e'_{\lambda,w} = e_{\lambda,w}P$ e consideram três alternativas principais :

## 1. Um ataque por texto claro escolhido para achar G'

$G'$ pode ser descoberta através de um ataque por texto claro escolhido como o realizado no cripto-sistema de Rao-Nam [2]. No esquema que estamos considerando, entretanto, é impossível, em princípio, decodificar usando $G'$, porque $G'$ é a matriz geradora de um código linear capaz de corrigir até $t$ erros aleatórios e $e'_{\lambda,w}$ é um vetor erro com peso $w > t$. Assim, não se pode usar o algoritmo de Korzhik e Turkin para decodificar [7]. De fato, mesmo se uma busca exaustiva fosse viável, o criptoanalista teria problemas, pois existem várias palavras a uma distancia $\leq w$ do vetor $c$ e não seria possível identificar a palavra transmitida.

Como o criptoanalista não pode usar as técnicas de decodificação do código gerado por $G'$, restam duas possibilidades, a saber, fatorar $G'$ em $G$ e $P$ ou recuperar $\boldsymbol{m}$ de $c$ sem conhecer $G$ e $P$.

## 2. Fatorando G' em G e P

Neste caso, a segurança reside no número $N_B$ de códigos corretores de erros bi-separáveis que são combinatorialmente equivalentes para um dado conjunto de parâmetros $n,k,b$. Em cada tentativa, o criptoanalista escolhe uma das $N_B$ permutações que levam $G'$ em $G''$, a matriz geradora de um dos $N_B$ códigos da classe. Para validar uma solução, é preciso decifrar um criptograma $c$ de um par texto claro / texto cifrado conhecido, $(\boldsymbol{m},c)$. O fator de trabalho desse ataque é

$$T_1 = \frac{N_B}{2} \text{ x complexidade de decodificação} \tag{7}$$

## 3. Recuperando m de c sem Conhecer G e P

Neste ataque, proposto por Lee e Brickell [8], um conjunto de $k$ bits é selecionado aleatoriamente de $c$. Este conjunto é testado exaustivamente para padrões de erro de até $j$ erros. Se um padrão com $j$ ou menos erros é encontrado, o algoritmo para. Caso contrário, um novo conjunto de $k$ bits é selecionado. A probabilidade de se ter até $j$ erro, nas $k$ posições $P_j$, é dada por

$$P_j = \sum_{i=0}^{j} P(i \text{ erros nas } k \text{ posições}) \tag{8}$$

onde

$$\text{P (i erros nas k posições )} = \sum_{w=i}^{b} \frac{\binom{k}{i}\binom{n-k}{w-i}}{\binom{n}{w}} x \frac{N_e(w)}{N_{te}(w_{\min})} \tag{9}$$

com $N_e(w)$ e $N_{te}(w_{\min})$ denotando, respectivamente, o número de padrões de erro de peso $w$ e o número total de padrões de erro para um dado $w_{\min}$, ou seja

$$N_{te}(w_{\min}) = \sum_{i=0}^{b} N_e(w) \tag{10}$$

Portanto, o número médio de escolhas de $k$ bits é $P_j^{-1}$ e, para cada escolha, $N_j$ testes são realizados, onde

$$N_j = \sum_{i=0}^{j} \binom{k}{i} \tag{11}$$

e o fator de trabalho desse ataque é $T_2' = P_j^{-1}(k^{\alpha} + N_j k^{\beta})$, onde $2<\alpha<3$ e $1<\beta\leq 2$ (os fatores $\alpha$ e $\beta$ estão associados, respectivamente, à solução de sistemas lineares de $k$ equações e a inversão de matrizes $k \times k$). O criptoanalista então escolhe $j$ de modo a minimizar o fator de trabalho $T_2'$, isto é, $j$ é a solução de $T_2 = min_j T_2'$. Para $r = 18$, $s = 21$, $q = 32$ e $w_{min} = 19$, resulta $T_2 = 1.4 \times 10^{28}$ [9].

Os fatores de trabalho referentes aos ataques mencionados apresentam valores que não comprometem a segurança do cripto-sistema quando o mesmo é implementado através da classe de códigos introduzida na seção III. Entretanto, sua utilização requer a introdução de uma matriz de embaralhamento $S$. Isto se deve ao fato das $(r+1)!(s+1)!$ permutações que preservam as equações de paridade, transformarem um EBS$(r+1, s+1)$ em outro EBS$(r+1, s+1)$ . Dessa forma, conhecendo a classe de códigos utilizada, o criptoanalista precisa apenas ordenar as colunas da matriz $G'$, de modo que ela seja uma das matrizes geradoras dos códigos combinatorialmente equivalentes ao código gerado por $G$. Em outras palavras, a matriz $GP$ contém uma estrutura óbvia que precisa ser escondida.

Usando $G' = SGP$, o criptoanalista terá que achar uma das $(r+1)!(s+1)!$ matrizes que servem para decodificar entre todas as $(r+1)(s+1)(rs)!$ matrizes geradoras da classe de códigos. A probabilidade de se escolher uma matriz que sirva é

$$P_3 = \frac{(r+1)!(s+1)!}{(rs)!(r+1)(s+1)} = \frac{r!s!}{(rs)!} \tag{12}$$

Portanto, o criptoanalista terá que fazer, em média,

$$T_3 = \frac{(rs)!}{r!s!} \tag{13}$$

tentativas antes de encontrar uma matriz que sirva. Os demais ataques e os fatores de trabalho correspondentes são os mesmos que foram apresentados anteriormente, com a exceção de que o número de padrões de erro de peso $w$ será dado por

$$N_e(w) = N_{EBS(r+1,\, s+1)}(w) = \binom{q-1}{w} \prod_{i=1}^{w} (r+2-i)(s+2-i) \tag{14}$$

$w = w_{min}, \ldots, w_{max}$, e o número de testes para cada escolha de $k$ componentes do vetor $c$, $N_j$, será dado por

$$N_j = \sum_{i=0}^{j} \binom{k}{i} (q-1)^j \tag{15}$$

No código descrito acima, para $r+s-1$ componentes de redundância, o código corrige EBS($r$+1, $s$+1)'s de peso até min($r$,$s$) (supondo q > min($r$,$s$)). Como é desejável que para uma dada redundância o código possa corrigir erros com maior peso possível, deve-se ter r≈s.

# 5. CONCLUSÕES

Este trabalho descreve a concepção de um novo cripto-sistema de chave privada que emprega códigos produto multiníveis. O processo de cifragem proposto faz uso de um tipo de configuração de erro que não existe em um sistema real de comunicação. Tal característica, embora pouco relevante devido ao fato de que os erros utilizados em um cripto-sistema bascado em códigos para controle de erros são gerados artificialmente no transmissor, resulta no aumento do número de padrões de erros disponíveis no sistema, o que permite ao mesmo resistir aos principais ataques conhecidos na literatura. Os códigos utilizados tem taxas altas e as operações de codificação e decodificação apresentam complexidade computacional linear, o que permite implementar o cripto-sistema a velocidades comparáveis aos principais sistemas atualmente em uso.

## Agradecimentos

Este trabalho recebeu apoio do *Conselho Nacional de Desenvolvimento Científico e Tecnológico* - CNPq.

# REFERÊNCIAS

[1] R.J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory", *DSN Progress Report* 42-44, pp. 114-116, Jet Propulsion Laboratory, Pasadena, CA, janeiro/fevereiro, 1978.

[2] T.R.N. Rao e K. Nam, "Private-Key Algebraic-Code Encryption", *IEEE Transactions*, vol. IT-35, Nº 4, pp. 829-833, julho 1989.

[3] W. Ximmer, "Digital Signature Scheme Based on Error-Correcting Codes", *Electronics Letters*, vol. 26, Nº 13, pp. 898-899, junho 1990.

[4] E.M. Gabidulin, "Ideals Over a Non-Commutative Ring and Their Applications in Cryptography", *Eurocrypt 91*, Lecture Notes in Computer Science, vol. 547, 1991.

[5] F.M.R. Alencar, A.M.P. Léo e R.M. Campello de Souza, "Private-Key Burst Correcting Code Encryption", *Proceedings of the IEEE International Symposium on Information Theory*, pp. 227, janeiro 1993.

[6] R.M. Campello de Souza e J. Campello de Souza, "Array Codes For Private-Key Encryption", *Electronics Letters*, vol. 30, Nº 17, pp. 1394-1396, agosto 1994.

[7] V.I. Korzhik e A.I. Turkin, "Cryptanalysis of McEliece's Public-key Cryptosystem" em *Advances in Cryptology*, Eurocrypt 91 Proceedings, pp. 68-70, Springer-Verlag, 1991.

[8] P. J. Lee e E. F. Brickell, "An observation on the Security of McEliece's Public-Key Cryptosystem", em *Advances in Cryptology*, Eurocrypt 88 Proceedings, pp. 275-280, Springer-Verlag, 1988.

[9] J. Campello de Souza, "Sistemas Criptográficos Baseados em Códigos Corretores de Erros", *Dissertação de Mestrado*, Departamento de Eletrônica e Sistemas, UFPE, agosto 1994.

**RICARDO MENEZES CAMPELLO DE SOUZA** formou-se em Engenharia Elétrica pela Universidade Federal de Pernambuco em 1974, obteve o título de Mestre em Ciências pela mesma Universidade em 1979 e o titulo de PhD pela University of Manchester, Inglaterra, em 1983, ambos em Engenharia Elétrica. Desde 1979 é Professor do Departamento de Eletrônica e Sistemas da UFPE, onde foi coordenador do Programa de Pós-graduação em Engenharia Elétrica no período 1984-1987, Chefe do Departamento no período 1987-1992 e atualmente ocupa a posição de Professor Adjunto. Seus interesses de pesquisa incluem matemática discreta, teoria da codificação, criptografia e processamento digital de sinais.

**JORGE CAMPELLO DE SOUZA** formou-se em Engenharia Elétrica pela Universidade Federal de Pernambuco em 1992, onde obteve o título de Mestre em Ciências em Engenharia Elétrica em 1994. Atualmente encontra-se cumprindo programa de doutoramento no Information Systems Laboratory, Stanford University. Seus interesses de pesquisa incluem matemática discreta, codificação de canal, criptografia e processamento digital de sinais.