Melhorando o ataque de reação contra o QC-MDPC McEliece

Thales Paiva
Prof. Dr. Routo Terada (orientador)
{tpaiva, rt}@ime.usp.br

14 de fevereiro de 2019

Instituto de Matemática e Estatística da Universidade de São Paulo

Melhorando o ataque de reação contra o QC-MDPC McEliece

QC-MDPC McEliece

- Esquema de encriptação de chave pública elaborado em 2013
- Variante do esquema de McEliece com chaves compactas
- QC-MDPC = *quasi-cyclic moderate density parity-check*
- Até 2016, era considerado um esquema pós-quântico

Ataque de reação

- Obtém informação através da decifrabilidade de textos cifrados
- Usa a informação obtida para reconstruir a chave
- No caso de um esquema seguro contra ataques de texto cifrado escolhido (IND-CCA), só são permitidos textos cifrados aleatórios como desafios

Melhorando o ataque

- Duas propostas de algoritmos para a reconstrução de chave

Motivação

Por que esquemas criptográficos com códigos corretores de erros?

- Esquemas baseados em códigos usam um problema supostamente difícil até para computadores quânticos
- Algoritmos quânticos ameaçam RSA [**?**] e Curvas Elípticas [**?**]
- Outros esquemas supostamente pós-quânticos são baseados em:
  - Reticulados
  - Hashes
  - Equações quadráticas multivariadas
  - Isogenias de curvas elípticas
- Esquema de McEliece é o principal dos baseados em códigos

Criptografia com Códigos Corretores

Esquema de McEliece [?]

- Encriptação e Decriptação mais eficientes que RSA e CE [**?**]
- Baseado no problema $\mathcal{NP}$-difícil da decodificação [**?**]
  (talvez dê uma falsa sensação de segurança?)
- Principal problema é que usa **chaves públicas muito grandes**

Nível de segurança aproximado	Tamanho da chave pública [**?**] (Bytes)		
	Curvas Elípticas	RSA	McEliece
80	20	128	65006
128	32	384	261702
256	64	1920	1046739

Variantes do esquema de McEliece

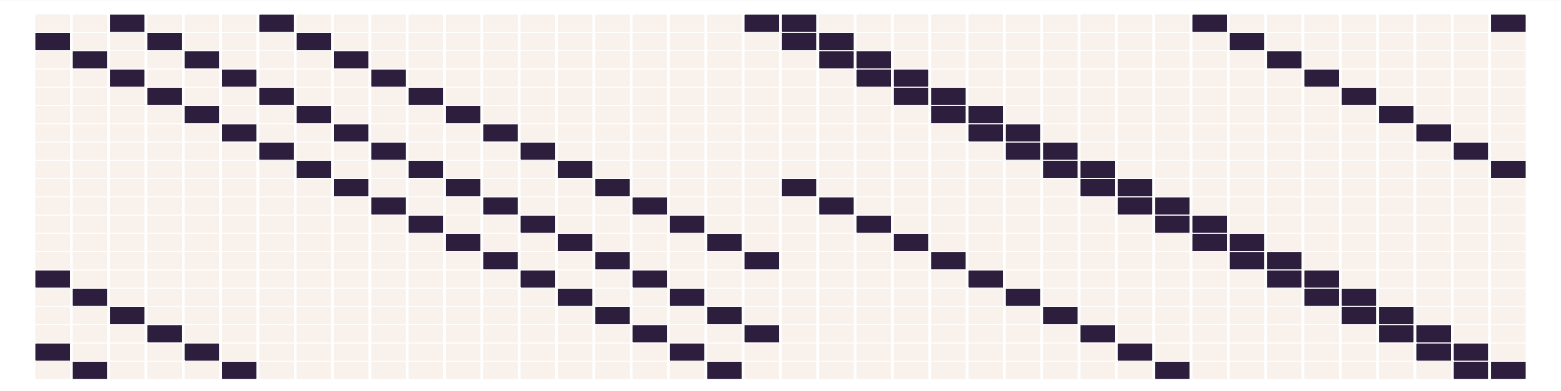
- O original usa códigos de Goppa binários e irredutíveis [**?**, **?**]
- Vários códigos propostos para chaves compactas [**?**, **?**, **?**, **?**]
- A maioria foi quebrada pouco depois de sua publicação [**?**, **?**, **?**]
- Até 2016 a variante **QC-MDPC** [**?**] era a mais promissora [**?**]

Tabela 1: Parâmetros sugeridos para cada nível de segurança [**?**]

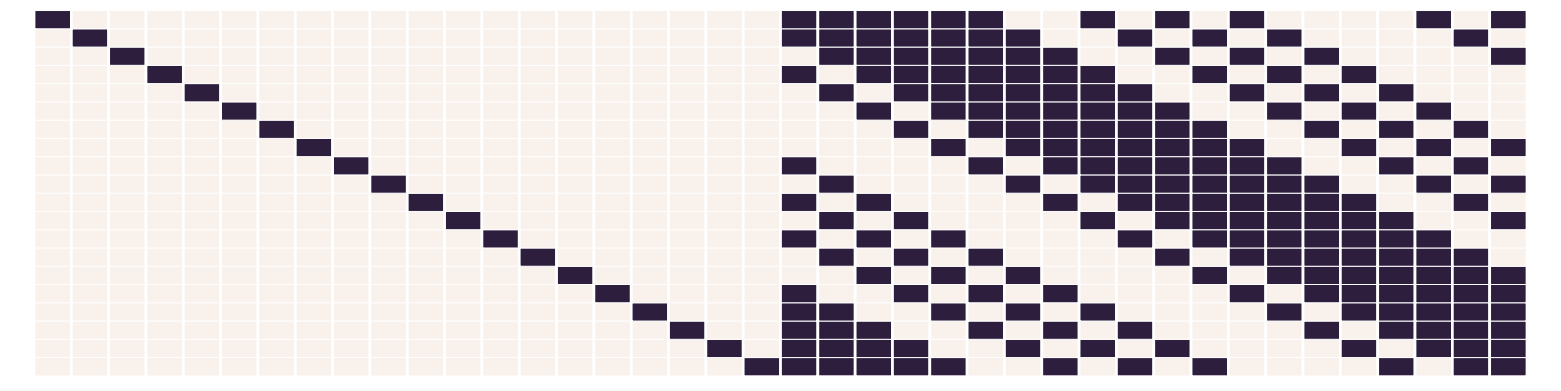
Segurança	n_0	n	r	w	t	Tamanho da chave (Bytes)
80	2	9602	4801	90	84	601
128	2	19714	9857	142	134	1233
256	2	65542	32771	274	264	4097

Códigos QC-MDPC

$\boldsymbol{H} = [\boldsymbol{H}_0 | \boldsymbol{H}_1]$, onde $\boldsymbol{H}_0$ e $\boldsymbol{H}_1$ são cíclicas e esparsas

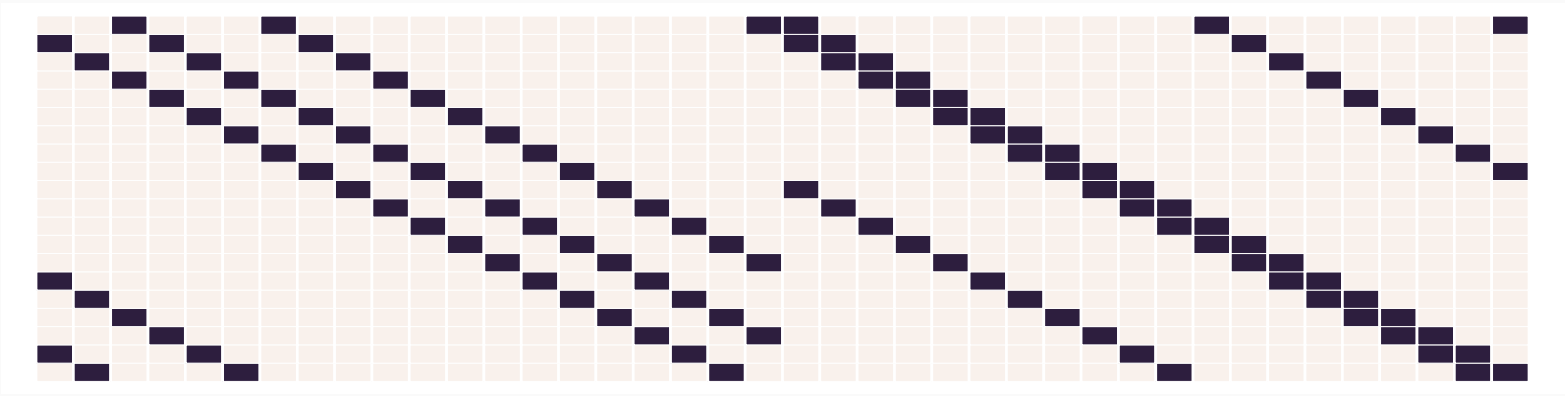


$\boldsymbol{G} = \left[\quad \boldsymbol{I} \quad | \quad (\boldsymbol{H}_1^{-1}\boldsymbol{H}_0)^T \right]$ é sistemática e densa

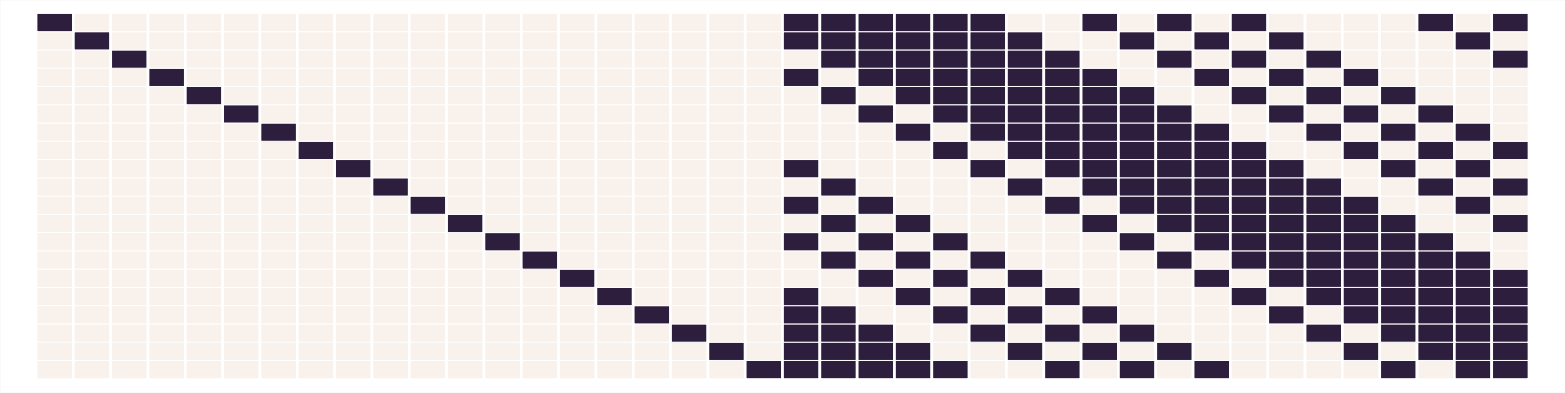


Códigos QC-MDPC

$\boldsymbol{H} = [\boldsymbol{H}_0 | \boldsymbol{H}_1]$, onde $\boldsymbol{H}_0$ e $\boldsymbol{H}_1$ são cíclicas e esparsas



$\boldsymbol{G} = \left[\quad \boldsymbol{I} \quad | \quad (\boldsymbol{H}_1^{-1}\boldsymbol{H}_0)^T \right]$ é sistemática e densa



QC-MDPC McEliece

- **Chave privada**
  - É a matriz de paridade $\boldsymbol{H} = [\boldsymbol{H}_0|\boldsymbol{H}_1]$ de um código QC-MDPC
  - Basta guardar as primeiras linhas de $\boldsymbol{H}_0$ e $\boldsymbol{H}_1$ denotadas por $\boldsymbol{h}_0$ e $\boldsymbol{h}_1$
- **Chave pública**:
  - Matriz geradora densa $\boldsymbol{G}$
  - Basta guardar a primeira linha $\boldsymbol{g}$ do bloco cíclico $\boldsymbol{G}$
  - t, o número de erros que o código corrige com alta probabilidade
- **Encriptação** de uma mensagem $\boldsymbol{m}$:
  - Escolha aleatoriamente um vetor de erro $\boldsymbol{e}$ binário de peso t
  - Devolva $\boldsymbol{c} = \boldsymbol{mG} + \boldsymbol{e}$
- **Decriptação** de um texto encriptado $\boldsymbol{c}$
  - Seja Ψ um algoritmo de decodificação para códigos QC-MDPC
  - Corrija os erros de $\boldsymbol{c}$ obtendo $\boldsymbol{c}' = \boldsymbol{mG} = \Psi(\boldsymbol{c})$
  - Se $\Psi(\boldsymbol{c})$ falhar, peça para o remetente encriptar com outro erro
  - Resolva o sistema linear sobredeterminado e obtenha $\boldsymbol{m}$

Ataque contra o QC-MDPC McEliece [?]

- A probabilidade de o decodificador falhar é menor se $\boldsymbol{e}$ e $\boldsymbol{H}$ compartilham certas propriedades
- A cada sucesso ou falha de decodificação a atacante obtém informação sobre $\boldsymbol{H}$
- A atacante pode obter grande quantidade de informação sobre $\boldsymbol{H}$ com um número suficiente de desafios de decodificação
- Com informação suficiente, a matriz secreta $\boldsymbol{H}$ pode ser reconstruída

Mas que informação é possível obter através de testes de decodificação?

Espectro

Definição de espectro de um vetor binário

- O espectro de um vetor $\boldsymbol{v}$, denotado por $\sigma(\boldsymbol{v})$, é o conjunto de menores distâncias cíclicas entre posições não-nulas de $\boldsymbol{v}$
- Um exemplo
  - Seja $\boldsymbol{v} = [0\,1\,1\,0\,0\,0\,0\,0\,1]$
  - Então $\sigma(\boldsymbol{v}) = \{1, 2, 3\}$

Tome um texto encriptado $\boldsymbol{c} = \boldsymbol{mG} + [\boldsymbol{e}_0|\boldsymbol{e}_1]$

- Guo et al. mostraram que a probabilidade de $\Psi(\boldsymbol{c})$ falhar é menor quanto maiores os tamanhos dos conjuntos

$$\sigma(\boldsymbol{e}_0) \cap \sigma(\boldsymbol{h}_0) \text{ e } \sigma(\boldsymbol{e}_1) \cap \sigma(\boldsymbol{h}_1)$$

- O ataque de reação recupera os espectros dos vetores $\boldsymbol{h}_0$ e $\boldsymbol{h}_1$

Reconstrução de h_0 a partir de seu espectro

- Tome $\boldsymbol{h}_0 = [0\,1\,1\,0\,0\,0\,1\,0\,0]$, então $\sigma(\boldsymbol{h}_0) = \{1, 4\}$
- Suponha que não sabemos $\boldsymbol{h}_0$ mas temos as seguintes informações parciais:
  - $\boldsymbol{h}_0$ tem 3 entradas não nulas
  - $\boldsymbol{h}_0$ tem tamanho $r = 9$
  - $D_0 = \{2\}$ é um conjunto de distâncias fora de $\sigma(\boldsymbol{h}_0)$
  - $s_0 = 1$ é uma distância pertencente a $\sigma(\boldsymbol{h}_0)$

?	?	?	?	?	?	?	?	?

Reconstrução de h_0 a partir de seu espectro

- Tome $\boldsymbol{h}_0 = [0\,1\,1\,0\,0\,0\,1\,0\,0]$, então $\sigma(\boldsymbol{h}_0) = \{1, 4\}$
- Suponha que não sabemos $\boldsymbol{h}_0$ mas temos as seguintes informações parciais:
  - $\boldsymbol{h}_0$ tem 3 entradas não nulas
  - $\boldsymbol{h}_0$ tem tamanho $r = 9$
  - $D_0 = \{2\}$ é um conjunto de distâncias fora de $\sigma(\boldsymbol{h}_0)$
  - $s_0 = 1$ é uma distância pertencente a $\sigma(\boldsymbol{h}_0)$

1	**1**	?	?	?	?	?	?	?

Reconstrução de h_0 a partir de seu espectro

- Tome $\boldsymbol{h}_0 = [0\,1\,1\,0\,0\,0\,1\,0\,0]$, então $\sigma(\boldsymbol{h}_0) = \{1, 4\}$
- Suponha que não sabemos $\boldsymbol{h}_0$ mas temos as seguintes informações parciais:
  - $\boldsymbol{h}_0$ tem 3 entradas não nulas
  - $\boldsymbol{h}_0$ tem tamanho $r = 9$
  - $D_0 = \{2\}$ é um conjunto de distâncias fora de $\sigma(\boldsymbol{h}_0)$
  - $s_0 = 1$ é uma distância pertencente a $\sigma(\boldsymbol{h}_0)$

1	1	**0**	**0**	?	?	?	**0**	**0**

Reconstrução de h_0 a partir de seu espectro

- Tome $\boldsymbol{h}_0 = [0\,1\,1\,0\,0\,0\,1\,0\,0]$, então $\sigma(\boldsymbol{h}_0) = \{1, 4\}$
- Suponha que não sabemos $\boldsymbol{h}_0$ mas temos as seguintes informações parciais:
  - $\boldsymbol{h}_0$ tem 3 entradas não nulas
  - $\boldsymbol{h}_0$ tem tamanho $r = 9$
  - $D_0 = \{2\}$ é um conjunto de distâncias fora de $\sigma(\boldsymbol{h}_0)$
  - $s_0 = 1$ é uma distância pertencente a $\sigma(\boldsymbol{h}_0)$

1	1	0	0	**1**	**0**	**0**	0	0

Reconstrução de h_0 a partir de seu espectro

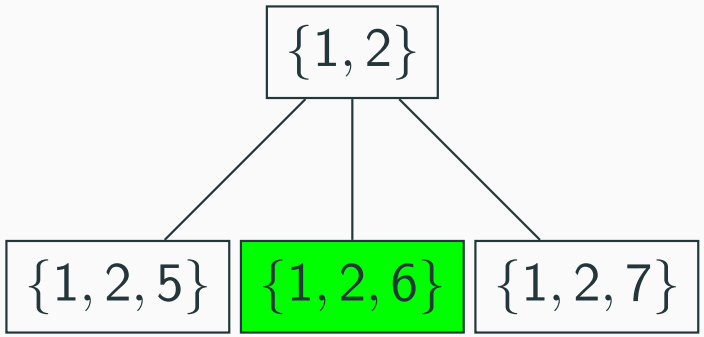
- Tome $\boldsymbol{h}_0 = [0\,1\,1\,0\,0\,0\,1\,0\,0]$, então $\sigma(\boldsymbol{h}_0) = \{1, 4\}$
- Suponha que não sabemos $\boldsymbol{h}_0$ mas temos as seguintes informações parciais:
  - $\boldsymbol{h}_0$ tem 3 entradas não nulas
  - $\boldsymbol{h}_0$ tem tamanho $r = 9$
  - $D_0 = \{2\}$ é um conjunto de distâncias fora de $\sigma(\boldsymbol{h}_0)$
  - $s_0 = 1$ é uma distância pertencente a $\sigma(\boldsymbol{h}_0)$

1	1	0	0	**0**	**1**	**0**	0	0

Reconstrução de h_0 a partir de seu espectro

- Tome $\boldsymbol{h}_0 = [0\,1\,1\,0\,0\,0\,1\,0\,0]$, então $\sigma(\boldsymbol{h}_0) = \{1, 4\}$
- Suponha que não sabemos $\boldsymbol{h}_0$ mas temos as seguintes informações parciais:
  - $\boldsymbol{h}_0$ tem 3 entradas não nulas
  - $\boldsymbol{h}_0$ tem tamanho $r = 9$
  - $D_0 = \{2\}$ é um conjunto de distâncias fora de $\sigma(\boldsymbol{h}_0)$
  - $s_0 = 1$ é uma distância pertencente a $\sigma(\boldsymbol{h}_0)$

1	1	0	0	**0**	**1**	**0**	0	0



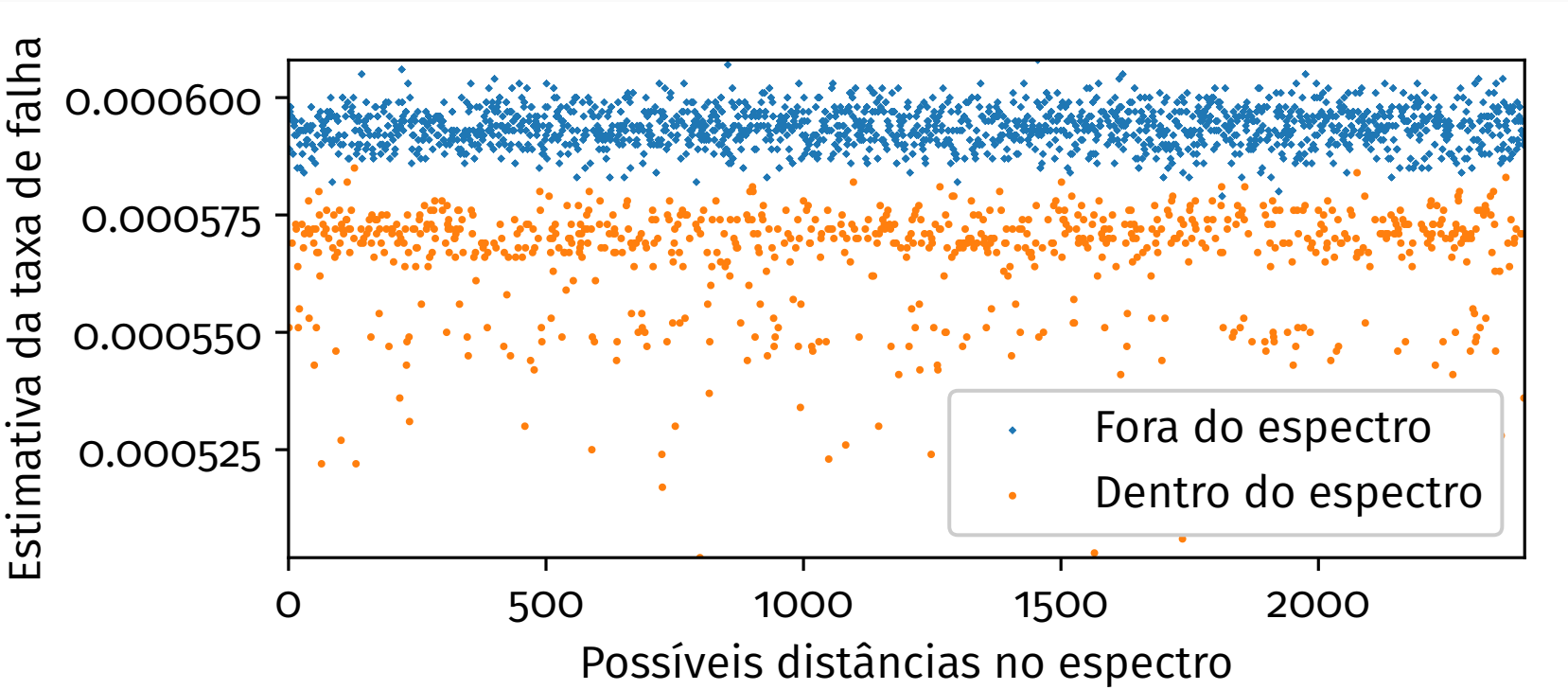
Reconstrução de h_0 a partir de seu espectro

- Guo et al. só fazem testes para o nível de segurança de 80 bits e considerando que D_0 contém todas as distâncias fora do espectro
- Seu algoritmo tem tempo médio de execução de 144s
- O argumento para o desempenho do algoritmo é que ramos ruins da árvore de busca são podados relativamente cedo
- Pode ser difícil paralelizar a busca em profundidade
- O desempenho do algoritmo pode ser muito ruim para níveis de segurança mais altos

Nível de segurança λ	Número médio de caminhos	Extrapolação para o tempo médio de execução
80	$2^{25.45}/2$	144s [**?**]
128	$2^{39.72}/2$	$\sim$ 32 dias
256	$2^{61.95}/2$	$\sim$ 422930 anos

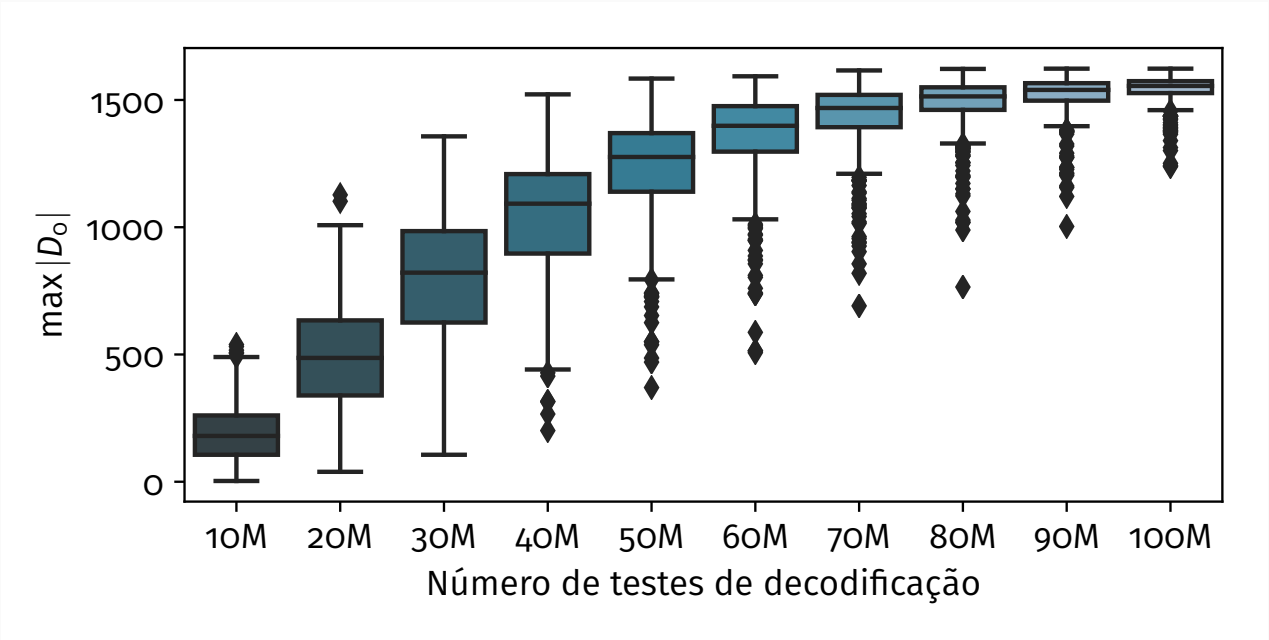
Como classificar distâncias em $\sigma(\boldsymbol{h}_0)$ e $\sigma(\boldsymbol{h}_1)$

- Estimar as taxas de falha de decodificação para cada distância
- As taxas de falha são usadas para classificar as distâncias
- É construído um conjunto D_0 com distâncias fora do espectro
- D_0 **não pode** conter distâncias dentro do espectro



Como o número de testes de decodificação afeta $\max|D_0|$

- O algoritmo de Guo et al. precisa de em torno de 200M de testes
- Testes de decodificação são caros pois envolvem comunicação com o decodificador
- Determinar o número de testes para um ataque bem sucedido é importante para estimar a vida útil de uma chave secreta



As simulações foram realizadas com o auxílio dos recursos de HPC disponibilizados pela Superintendência de Tecnologia da Informação da Universidade de São Paulo.

Nosso objetivo

Propor novos algoritmos de reconstrução de chave que:

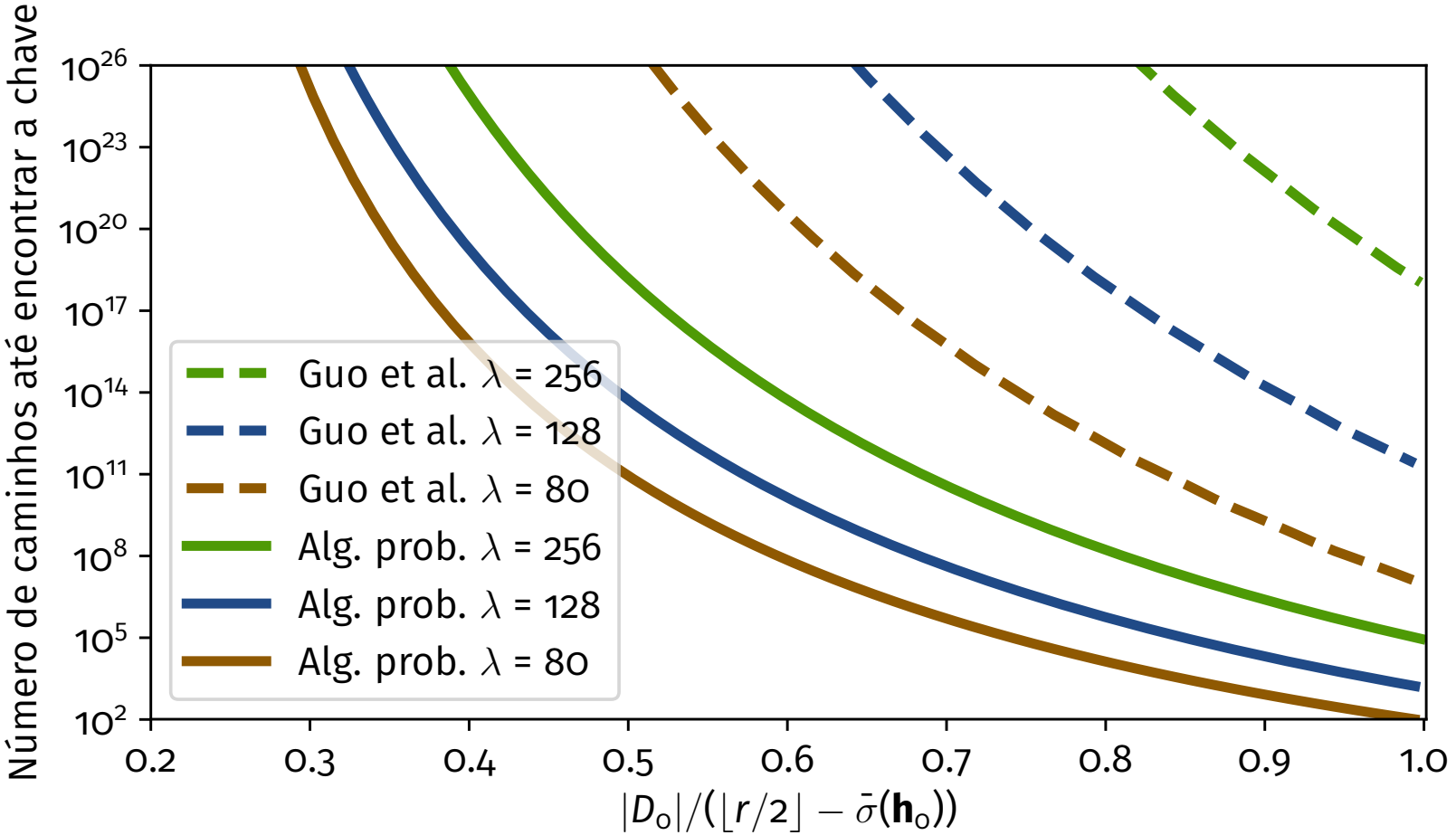
- Sejam mais eficientes do que o de Guo et al.
- Consigam lidar com menores tamanhos de $|D_0|$

Algoritmo probabilístico de reconstrução

- Similar ao algoritmo de Guo et al. mas escolhe aleatoriamente onde colocar ‘1’s
- A cada iteração, percorre um caminho aleatório na árvore de busca da raiz até uma de suas folhas
- Encontrar a chave se reduz a poucas escolhas certas no começo de cada iteração
- Evita que o algoritmo de reconstrução fique preso em ramos ruins da árvore de busca
- Principal problema é não ser eficiente nos níveis de segurança mais altos do que 80 quando $|D_0|$ não é muito grande

Algoritmo probabilístico de reconstrução

O algoritmo probabilístico percorre, em média, bem menos caminhos do que o de Guo et al.



Desempenho do algoritmo probabilístico

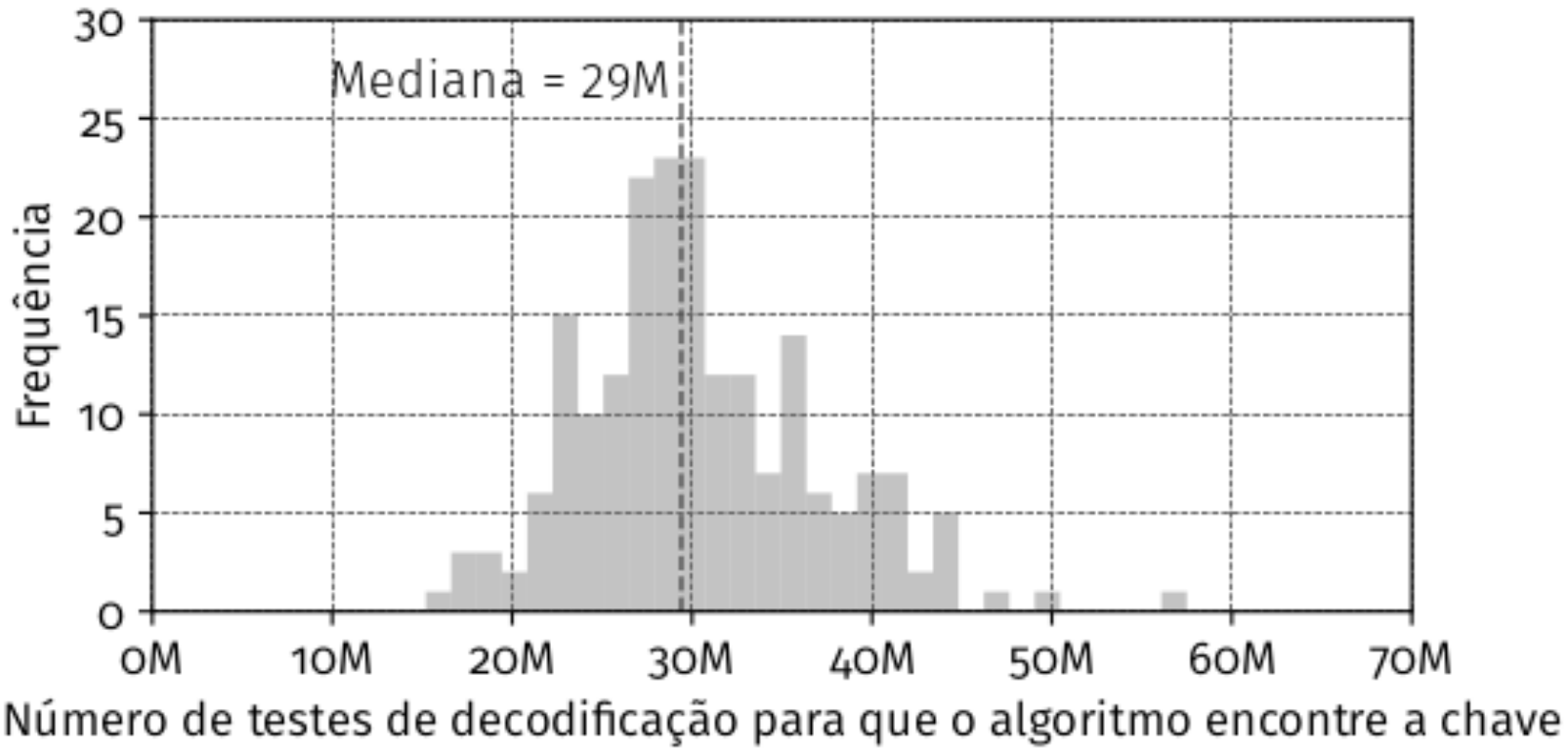
- O algoritmo é mais eficiente que o algoritmo de Guo et al. em todos os tamanhos realistas de $|D_0|$
- Ele é paralelizável trivialmente

Nível de segurança λ	$\lvert D_0 \rvert$ normalizado	Tempo médio de execução
80	62.92%	**142s**
128	77.27%	**165s**
256	89.45%	**251s**

Considerando o algoritmo executado em 16 cores de um processador Intel Xeon E7-2870 @ 2.40GHz

Algoritmo probabilístico de reconstrução

- $|D_0| = 1000$ é suficiente para que o algoritmo funcione eficientemente
- Basta que $|D_0|$ ou $|D_1|$ sejam ≥ 1000



Algoritmo iterativo de reconstrução

- Explora a relação linear $\boldsymbol{h}_1\boldsymbol{B} = \boldsymbol{h}_0$, onde $\boldsymbol{B} = \boldsymbol{H}_1^{-1}\boldsymbol{H}_0$ é pública
- Usa testes de decodificação para obter conjuntos D_0 e D_1, que contêm distâncias fora dos espectros de $\boldsymbol{h}_0$ e $\boldsymbol{h}_1$
- Usa s_0 e s_1, que são distâncias nos espectros de $\boldsymbol{h}_0$ e $\boldsymbol{h}_1$
- Usando s_0 e D_0, encontram-se posições de entradas nulas de uma rotação de $\boldsymbol{h}_0$
- Usando s_1 e D_1, encontram-se posições de entradas nulas de uma rotação de $\boldsymbol{h}_1$
- Com um número suficiente de posições nulas conhecidas, pode-se obter o vetor $\boldsymbol{h}_1$ através da resolução de um sistema linear homogêneo

Desempenho do algoritmo iterativo

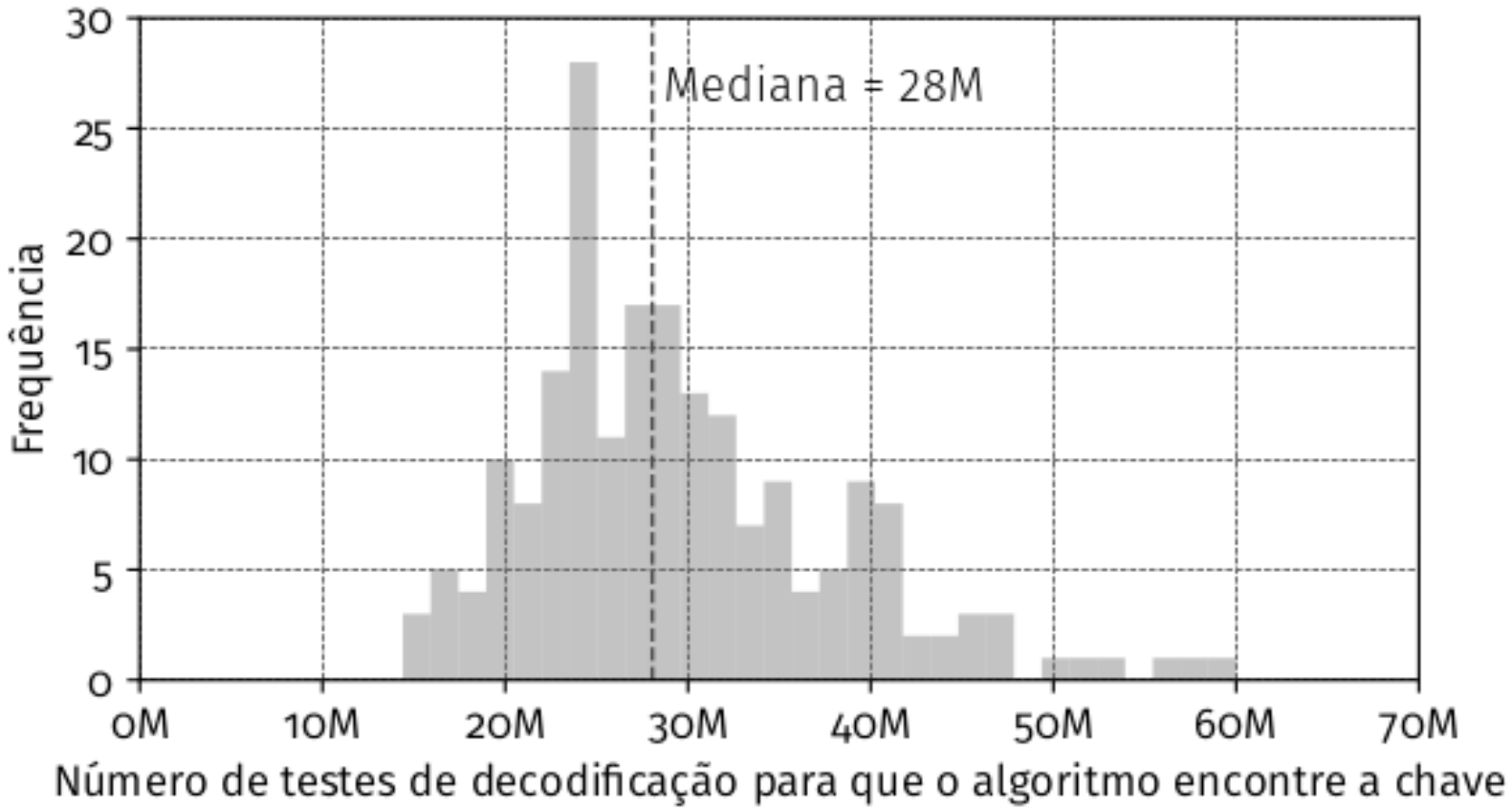
- O algoritmo pode falhar em encontrar a chave
- Sejam z_0 e z_1 o número de posições nulas conhecidas em $\boldsymbol{h}_0$ e $\boldsymbol{h}_1$, a probabilidade de falha é $\approx 2^{z_0+z_1-r}$

Considerando uma probabilidade de falhar de 2^{-20} para 99% dos códigos, o desempenho do algoritmo é

Nível de segurança λ	Tempo médio por iteração	$\lvert D_0\rvert = \lvert D_1\rvert$ normalizados	Tempo médio de execução
80	0.0169s	45.63%	**41s**
128	0.1710s	49.58%	**14m3s**
256	3.4179s	52.34%	**15h33m24s**

Considerando um processador i7 870 Lynnfield @ 2.93GHz. As operações de álgebra linear foram feitas usando a biblioteca M4RI [**?**]

Número necessário de testes de decodificação



Conclusão

- Ambos os algoritmos usam menos informação e são mais eficientes do que o de Guo et al.
- O algoritmo probabilístico sofre os mesmos problemas de escalamento que o de Guo et al.
- O algoritmo iterativo é muito mais robusto em relação à quantidade de informação
- Os dois algoritmos foram analisados tanto teoricamente quanto na prática, com implementações em C

Conclusão

- O algoritmo probabilístico pode ser mais eficiente que o iterativo quando o D_0 está quase completo
- O número necessário de decodificações para ambos os algoritmos é similar para o nível de segurança de 80 bits
- Mas é esperado que o algoritmo iterativo precise de menos decodificações do que o probabilístico em níveis de segurança mais altos
- O algoritmo iterativo foi submetido à revista Transactions on Fundamentals of Electronics, Communications and Computer Sciences do IEICE

Trabalhos futuros

- Considerar variantes dos algoritmos para tratar de casos em que há informação limitada sobre os espectros (chave tem vida útil)
- Combinar o algoritmos probabilístico com o iterativo usando heurísticas
- Melhorar o algoritmo de estimação do espectro para escolher conjuntos de desafios (aleatórios) com maior probabilidade de falha de decodificação
- Verificar como o ataque afeta o MDPC-McEliece
- Estender a análise do número necessário de decodificações para níveis de segurança mais altos e outros decodificadores
- Proteger o esquema encontrando decodificadores que falham com probabilidade independente do espectro do código

Referências i


D. Augot, L. Batina, D. J. Bernstein, J. Bos, J. Buchmann, W. Castryck, O. Dunkelman, T. Güneysu, S. Gueron, A. Hülsing, et al.
Initial recommendations of long-term secure post-quantum systems.
2015.


T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani.
Reducing key length of the McEliece cryptosystem.
In *Progress in Cryptology–AFRICACRYPT 2009*, pages 77–97. Springer, 2009.


E. R. Berlekamp.
Goppa codes.
IEEE Transactions on Information Theory, 19(5):590–592, 1973.

Referências ii


E. R. Berlekamp, R. J. McEliece, and H. C. Van Tilborg.
On the inherent intractability of certain coding problems.
IEEE Transactions on Information Theory, 24(3):384–386, 1978.


D. J. Bernstein, T. Chou, and P. Schwabe.
McBits: fast constant-time code-based cryptography.
In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 250–272. Springer, 2013.


J.-C. Faugere, A. Otmani, L. Perret, F. De Portzamparc, and J.-P. Tillich.
Structural cryptanalysis of McEliece schemes with compact keys.
Designs, Codes and Cryptography, 79(1):87–112, 2016.

Referências iii


J.-C. Faugere, A. Otmani, L. Perret, and J.-P. Tillich.
Algebraic cryptanalysis of McEliece variants with compact keys.
In *Advances in Cryptology–Eurocrypt 2010*, pages 279–298. Springer, 2010.


P. Gaborit.
Shorter keys for code based cryptography.
In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, 2005.


V. D. Goppa.
A new class of linear correcting codes.
Problemy Peredachi Informatsii, 6(3):24–30, 1970.

Referências iv


Q. Guo.
Apresentação de Guo na Asiacrypt.
`https://youtu.be/tKvDdGLJLZc?t=1006`, 2016.


Q. Guo, T. Johansson, and P. Stankovski.
A key recovery attack on MDPC with CCA security using decoding errors.
In *22nd Annual International Conference on the Theory and Applications of Cryptology and Information Security (ASIACRYPT)*, 2016.


Martin Albrecht and Gregory Bard.
The M4RI Library – Version 20121224.
The M4RI Team, 2012.

Referências v


R. J. McEliece.
A public-key cryptosystem based on algebraic coding theory.
Deep Space Network Progress Report, 44:114–116, 1978.


R. Misoczki and P. S. Barreto.
Compact McEliece keys from Goppa codes.
In *Selected Areas in Cryptography*, pages 376–392. Springer, 2009.


R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. Barreto.
MDPC-McEliece: New McEliece variants from moderate density parity-check codes.
In *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 2069–2073. IEEE, 2013.

Referências vi


A. Otmani, J.-P. Tillich, and L. Dallot.
Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes.
Mathematics in Computer Science, 3(2):129–140, 2010.


R. L. Rivest, A. Shamir, and L. M. Adleman.
A method for obtaining digital signatures and public-key cryptosystems.
Communications of the ACM, 21(2):120–126, 1978.


A. Shokrollahi, C. Monico, and J. Rosenthal.
Using low density parity check codes in the McEliece cryptosystem.
In *IEEE International Symposium on Information Theory (ISIT 2000)*, page 215, 2000.


P. Shor.
Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer.
SIAM Journal on Computing, 26(5):1481–1509, 1997.