Word-Hunt: A LSB Steganography Method with Low Expected Number of Modifications per Pixel

R. Tavares and F. Madeiro

Abstract— **Least Significant Bit (LSB) steganography is a well known technique which operates in the spatial domain of digital images. In this paper, the LSB Word-Hunt (LSB WH) is presented. It is a novel LSB approach inspired by the word-hunt puzzle. The main focus of LSB WH is to reduce the Expected Number of Modifications per Pixel (ENMPP) when compared to other methods in the literature. The results show that LSB WH has an ENMPP around 0.315, for natural images with high entropy on the second and the third least significant bits. Results also show that the new method is robust to the statistical chi-square attack.**

Keywords— **image processing, LSB steganography, ENMPP reduction.**

I. Introdução

ESTEGANOGRAFIA é a ciência e a arte de ocultação de informação em um objeto hospedeiro, buscando obter confidencialidade ao ocultar a própria existência da informação [1-10].

O campo de estudos da esteganografia vem crescendo massivamente desde o século XXI. Evidenciada pelo constante aumento no número de artigos científicos publicados [1], vem se mostrando uma área relevante e atual.

O método do Bit Menos Significativo (LSB, *least significant bit*) é uma técnica conhecida que opera no domínio espacial de imagens digitais. O método consiste em uma substituição simples dos bits menos significativos dos pixels da imagem pelos bits da informação a ser escondida. O impacto visual na imagem é pequeno, levando em consideração que apenas os LSBs são manipulados. A carga útil do método de esteganografia LSB é de 1 bpp (bit por pixel). Ressalte-se que a carga útil representa o espaço disponível para ocultar informações. A carga utilizada é medida por uma taxa percentual, denotada por α [1]. Apesar do baixo impacto visual, a robustez do método para ataques de esteganálise, contra-arte que tem como propósito constatar a presença de dados ocultos [1,3,11-22], é pequena. Uma contribuição de Westfeld e Pfitzmann [11] foi observar que dois valores de pixel que apenas diferem pelo LSB, denominados como um Par de Valores (PoV, *pair of values*), tendem a aproximar seus números de ocorrências se o método LSB for utilizado para esconder uma informação de alta entropia em seus bits. Uma consequência da observação supracitada é o histograma da imagem se tornar alvo para ataques estatísticos, como o ataque do Qui-Quadrado [1,11]. Quanto maior for a carga utilizada, maior a chance do êxito do ataque.

J. R. C. Tavares, Universidade de Pernambuco (UPE), Pernambuco, Brasil, jrt_ppges@poli.br

F. Madeiro, Universidade de Pernambuco (UPE), Pernambuco, Brasil, madeiro@poli.br

Sharp [4] introduziu um método de esteganografia denominado LSB Matching (LSB M). A técnica utiliza métodos auxiliares para prover criptografia e compressão aos dados secretos, na chamada etapa de preparação, com a expectativa de que seus bits apresentem uma alta entropia ao final da etapa. O método utiliza uma sequência pseudo-aleatória para leitura dos pixels, gerada por uma estego-chave, termo dado a uma chave utilizada para propósitos de esteganografia [1]. A função de ocultação do LSB M altera o bit menos significativo, quando necessário, através de uma adição ou subtração unitária no valor do pixel, com a escolha da operação sendo aleatória para cada bit dos dados secretos. Essa técnica evita o problema de aproximação dos PoVs.

O número esperado de modificações por pixel (ENMPP, *expected number of modifications per pixel*) é uma taxa que estabelece uma relação entre carga embutida e modificações na imagem de cobertura [5,6]. Para o LSB M, devido à alta entropia nos bits da informação secreta, a probabilidade de um pixel necessitar alterar seu LSB é 0,5. Logo, para cada dois bits ocultados, deverá ser alterado, em média, apenas 1 pixel, unitariamente. O ENMPP do LSB M pode ser descrito como 0,5, indicando que a carga embutida corresponde ao dobro do número de modificações sofridas pela imagem.

Mielikainen [5] introduziu o LSB Matching Revisited (LSB MR), que é uma melhoria do LSB M. A informação secreta ainda é submetida à mesma etapa de preparação e o percurso pseudo-aleatório continua em uso. A função de ocultação do LSB MR utiliza um par de pixels para esconder um par de bits. O primeiro bit oculto é recuperado através de uma leitura do LSB do primeiro pixel. O segundo, através de uma função binária dos dois pixels. O ENMPP para o LSB MR é 0,375.

Sarreshtedari *et al.* [6] introduziram o LSB One-Third (LSB OT), nomeado em referência ao ENMPP alcançado. O método utiliza a mesma etapa de preparação e percurso pseudo-aleatório de seus predecessores LSB M e LSB MR. O LSB OT esconde três bits secretos em uma tríade de pixels, ao aplicar uma função binária para cada combinação de dois pixels. O resultado principal dessa abordagem é a redução do ENMPP em relação às técnicas supramencionadas, que chega a uma taxa de aproximadamente 0,333.

Este trabalho apresenta um aprimoramento do LSB Word-Hunt (LSB WH), previamente introduzido em [7], cuja abordagem foi inspirada em um conhecido passatempo de localizar informações escondidas, o jogo de caça-palavras. Mudanças substanciais incluem: fixação dos tamanhos dos conjuntos de pixels e dos feixes da informação secreta; redução do espaço de busca; eliminação do uso de marcadores especiais; alteração do esquema de busca dos feixes de bits a serem ocultos. Este aprimoramento foi concebido para prover

uma redução do ENMPP em relação aos métodos da literatura supramencionados.

A redução do número esperado de modificações por pixel é um importante fator para o aumento da segurança de sistemas de esteganografia, sendo também explorada na pesquisa de outros autores [5, 6].

O restante deste artigo encontra-se organizado da seguinte maneira: a Seção II apresenta o método proposto. Na Seção III é apresentada a análise probabilística do LSB WH, com o intuito de verificar, de forma teórica, a qualidade do algoritmo no que diz respeito ao número de modificações por pixel. A Seção IV trata dos resultados e discussões e, finalmente, as conclusões são apresentadas na Seção V.

II. LSB WORD-HUNT

O jogo de caça-palavras tem por objetivo a localização de determinadas palavras em uma matriz de letras. O jogador é guiado por um conjunto de regras, que geralmente são simples: as palavras podem ser formadas em uma linha horizontal, vertical ou diagonal e, em algumas versões, podem ser formadas na ordem inversa. A Fig. 1 apresenta um tabuleiro de caça-palavras.

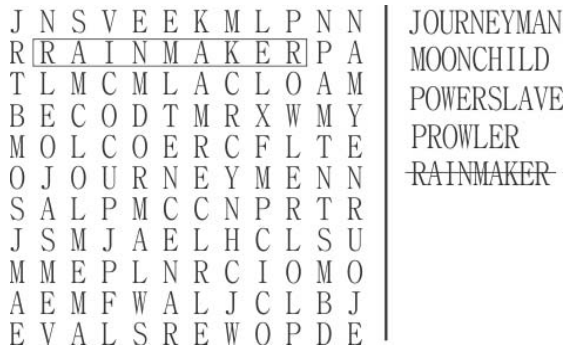

Figura 1. Tabuleiro de um jogo de caça-palavras, representando uma partida em andamento.

O LSB Word-Hunt é um método inspirado no jogo de caça-palavras. Os planos de bits 2-LSB (*second least significant bit*) e 3-LSB (*third least significant bit*) funcionam como a região de busca, onde as palavras, formadas por dígitos binários, são localizadas. A informação a ser oculta é dividida em feixes de bits. Cada feixe é localizado em um ponto da região de busca, mediante uma regra, tal qual o jogo original. O plano LSB guarda as informações necessárias para recuperar as palavras previamente localizadas, sendo o único plano que sofre modificações ao longo do processo.

Um conjunto de 16 pixels (PS, *pixel set*), selecionados através de uma leitura regida por uma sequência pseudo-aleatória, guarda um feixe de 6 bits do dado secreto. Uma palavra binária de 16 bits, denominada código de extração (EC, *extraction code*), rege o processo de busca/extração de feixes. Um código de extração contém as seguintes informações:

(a) **Ponto de partida** – Sub-código composto pelos 4 primeiros bits do EC. Indica qual dos 16 pixels é o ponto de partida para o processo de busca/extração do feixe no conjunto, tal que o valor 0 represente o 1º pixel do PS.
(b) **Plano de busca** – 5º bit do EC. Indica o plano de bits utilizado para a busca. O valor 0 indica o plano 2-LSB e o valor 1 indica o plano 3-LSB.
(c) **Alternador de planos** – 6º bit do EC. Quando possuir valor 1, alterna o plano de busca após cada iteração do processo de busca/extração.
(d) **Inversor de bits** – 7º bit do EC. Aplica inversão de valor para cada bit alcançado durante o processo de busca, se fixado em 1.
(e) **Índice da regra** – Sub-código composto pelos 9 últimos bits do EC. Indica o índice da regra utilizada para leitura do feixe.

Todo o processo é regido por uma chave simétrica que contém 512 regras de formação de palavra diferentes. Para cada regra do conjunto, estão associados dois bytes, que representam valores de incremento nas coordenadas X e Y da matriz de bits de um plano da imagem. Os incrementos são aplicados sistematicamente ao início de cada iteração. Cada byte representa um inteiro entre -128 e 127. Este incremento é regido por uma operação de módulo com a dimensão da imagem, de modo a evitar localidades de coordenadas inválidas. Os valores de incremento, durante a criação da estego-chave, devem ser selecionados com auxílio de elementos de aleatoriedade, que proporcionem uma distribuição o mais uniforme possível dos valores.

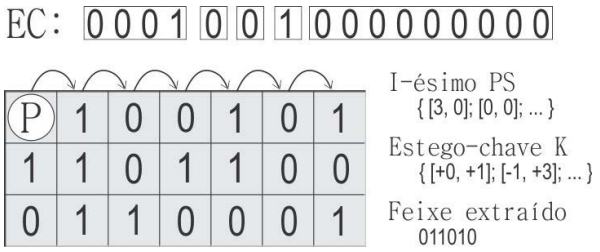

Figura 2. Extração de um feixe da informação escondida, regida pela estego-chave K.

A Fig. 2 apresenta um processo de recuperação de um feixe da informação para o *i*-ésimo conjunto de pixels da estego-imagem. O ponto de partida é indicado pelo EC como o 2º pixel do PS, sendo representado na figura pela letra 'P'. Informações complementares indicam que o plano de busca será o 2-LSB, não haverá alternância entre os planos durante o processo e cada bit alcançado será invertido. O índice da regra, referente à estego-chave K, indica que a linha permanecerá constante, enquanto a coluna receberá o incremento de uma unidade por iteração. O bit alcançado na primeira iteração é um bit 1. Logo, o primeiro bit do feixe será um bit 0, devido à inversão de bits acionada no EC. Após seis iterações é extraído um feixe de seis bits da informação secreta.

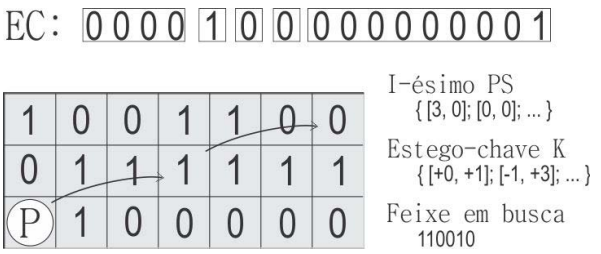

Figura 3. Tentativa sem sucesso de localizar feixe da informação secreta dentro da imagem, regida pela estego-chave K.

O processo de ocultação é realizado PS-a-PS. São testados diferentes códigos de extração, até algum retornar o feixe que está em busca. A ordem dos testes define a qualidade da ocultação, levando em consideração que, após algum EC obter sucesso, os demais não serão testados. Logo, os que apresentam melhores resultados devem ter prioridade na ordem dos testes. A Fig. 3 apresenta uma tentativa sem sucesso de localização, onde o EC empregado não foi capaz de localizar o feixe em questão, por diferir no segundo bit alcançado. Quando um código de extração obtiver sucesso na localização do feixe da informação secreta, o código binário deste é escrito nos LSBs do conjunto de pixels em questão, possibilitando que esse feixe possa ser recuperado da estego-imagem. Essa escrita poderá ocasionar modificações na imagem de cobertura.

Caso um PS não seja capaz de esconder um feixe, após todos os ECs possíveis serem testados, todo o processo deve ser abortado. Isso é um indicativo de que a distribuição dos bits da imagem não é adequada para o processo de ocultação do LSB WH. Nessa situação, aconselha-se a utilização de outra imagem.

Os códigos de extração são ordenados para teste seguindo duas prioridades: redução do número de modificações e manutenção do histograma. Esta buscando robustez contra ataques esteganalíticos, aquela, redução do ENMPP. *A priori*, os ECs são ordenados de acordo com a menor Distância de Hamming (HD, *Hamming Distance*) em relação aos 16 bits componentes dos LSBs do conjunto de pixels, sendo o primeiro EC um código similar ao presente nos LSBs do PS. Para os ECs que possuem semelhantes HD, é realizado um ordenamento com base na manutenção do histograma, o qual visa minimizar o efeito no histograma da imagem de cobertura, contemplando, ainda, robustez ao ataque do Qui-Quadrado.

Um sistema de pontuação rege o ordenamento dos ECs visando à manutenção do histograma – códigos com maior número de pontos são testados antes dos demais. O efeito no histograma é verificado bit-a-bit do código de extração, em comparação com o histograma original da imagem. Caso o bit do EC difira do LSB que o comportará e a modificação do mesmo implique uma aproximação do histograma em relação ao original, um ponto é acrescido ao código. Caso a modificação gere um afastamento do histograma original, uma nova análise é feita: se os números de ocorrências relacionados ao PoV se afastarem, é somado um ponto; caso se aproximem, é subtraído um ponto.

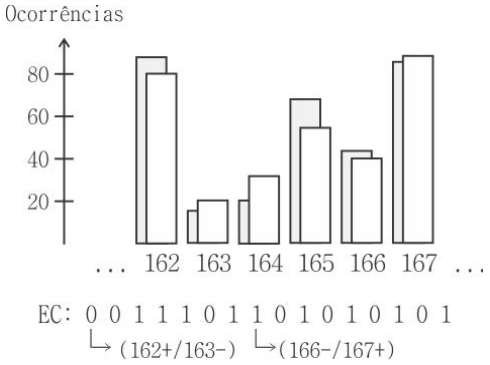

Figura 4. Histograma em análise para pontuação de EC.

A Fig. 4 apresenta um histograma em análise, onde as barras cinzas representam o histograma original, antes de qualquer modificação na imagem, e as barras brancas representam o histograma atual, após o processamento de alguns conjuntos de pixels e, consequentemente, a ocultação de alguns feixes. Um EC em análise difere em dois bits do código presente nos LSBs do PS em questão. O primeiro bit do EC implica um aumento unitário no número de ocorrências do nível de cinza 162 e, consequentemente, uma diminuição unitária para o seu par no PoV: 163. Essa mudança torna o histograma mais próximo do original, um ponto é acrescido ao código de extração. Para o oitavo bit, o efeito da mudança distancia o histograma em relação ao original, porém, como há um distanciamento nas ocorrências do PoV, um ponto é acrescido. O EC em questão possui dois pontos, deverá ser testado antes de ECs que possuam um ponto ou menos, e depois de ECs que possuam três pontos ou mais.

III. Análise Probabilística do LSB Word-Hunt

Considere p a probabilidade de sucesso em uma busca de um feixe da informação. O valor de p está diretamente ligado à disposição dos bits da região de busca de uma imagem. Considere Prob(PS ≠ ∅) a probabilidade de um PS ser capaz de ocultar um feixe da informação. Analogamente, considere Prob(PS = ∅) a probabilidade de um PS não ser capaz de ocultar um feixe da informação, ação ocorrida quando todos os códigos de extração elegíveis não tiverem sucesso em suas respectivas buscas. Temos então Prob(PS ≠ ∅) = 1 − Prob(PS = ∅) = 1 – (1 – p) × (1 – p) × . . . × (1 – p). Generalizando,

$$\text{Prob}(\text{PS} \neq \emptyset) = 1 - (1 - p)^{65536}, \quad (1)$$

sendo 65536 a quantidade de códigos elegíveis, uma vez que o código de extração possui 16 bits. Considere

$$\text{Prob}(\text{PS} \neq \emptyset \mid \text{HD} = d) = 1 - (1 - p)^{C_{16,d}}, \quad 0 \leq d \leq 16, \quad (2)$$

em que $C_{16,d}$ denota a combinação de 16 elementos d a d, a probabilidade de um PS ser capaz de ocultar um feixe, quando apenas ECs com Distância de Hamming igual a d se submeterem ao processo de busca.

Considere Prob(HD(EC) = x) a probabilidade de a Distância de Hamming entre o código de extração e o código

original disposto nos LSBs de um PS ser igual a x. Devido à priorização de mudanças mínimas, códigos de extração com HD > 0 apenas serão testados se todos os demais com menores HD obtiverem falhas de busca. Sendo assim,

$$\text{Prob}(\text{HD}(\text{EC}) = x) = \text{Prob}(\text{PS} \neq \emptyset \mid \text{HD} = x) \times \prod_{i=0}^{x-1}(1 - \text{Prob}(\text{PS} \neq \emptyset \mid \text{HD} = i)), 0 \leq x \leq 16. \quad (3)$$

Finalmente, considere q a quantidade de PSs utilizada em uma ocultação de dados e TC (*total changes)* a quantidade de modificações sofridas durante o processo, medida em bits. Temos então TC = 0 como a melhor situação pós-ocultação, e TC = 16 × q, como a pior. Considere Prob(TC = x) a probabilidade de a ocultação ter alterado x bits da imagem, tal que $0 \leq x \leq 16 \times q$. Consideremos

$$\text{Prob}(\text{TC} = x) = \sum_{i=0}^{n-1} \text{Prob}(\text{TC} = x \mid \text{C} = i), \quad (4)$$

em que C indica a componente de uma série de n componentes utilizadas para o cálculo de Prob(TC = x). Considere $\overrightarrow{a_i}$ como um vetor de 17 elementos que indica, para a componente i, quantos PSs dos q utilizados sofreram determinado número de alterações, em bits, compreendido entre 0 e 16, tal que a_{i_0} indique quantos PSs não sofreram alterações, a_{i_1} indique quantos PSs sofreram alteração em 1 bit, e assim sucessivamente. Temos então

$$\text{Prob}(\text{TC} = x \mid \text{C} = i) = P_q^{(a_{i_0}, a_{i_1}, \ldots, a_{i_{16}})} \times \prod_{j=0}^{16} \text{Prob}(\text{HD}(\text{EC}) = j)^{a_{i_j}}, \quad (5)$$

em que a permutação $P_q^{(a_{i_0}, a_{i_1}, \ldots, a_{i_{16}})}$ é utilizada para considerar as diferentes possibilidades de alterações permutadas pelos PSs. A Tabela I apresenta as componentes dos 5 primeiros possíveis valores de TC, considerando $q \geq 4$.

TABELA I. COMPONENTES *i* DOS 5 PRIMEIROS VALORES DE TC, PARA CÁLCULO DE MUDANÇAS TOTAIS.

TC	*i*	*j*					
		0	1	2	3	4	> 4; ≤ 16
0	0	q	0	0	0	0	0
1	0	q - 1	1	0	0	0	0
2	0	q - 2	2	0	0	0	0
	1	q - 1	0	1	0	0	0
3	0	q - 3	3	0	0	0	0
	1	q - 2	1	1	0	0	0
	2	q - 1	0	0	1	0	0
4	0	q - 4	4	0	0	0	0
	1	q - 3	2	1	0	0	0
	2	q - 2	1	0	1	0	0
	3	q - 2	0	2	0	0	0
	4	q - 1	0	0	0	1	0

Conforme descrito na Tabela I, há duas formas de uma ocultação, após finalizada, modificar dois, e somente dois, bits. A componente 0 indica que dois PSs tiveram um bit alterado. A componente 1 indica que apenas um PS sofreu alteração, porém a mesma foi de dois bits. Em termos matemáticos, podemos assumir

$$\text{TC} = \sum_{j=0}^{16} (j \times a_{i_j}). \quad (6)$$

Ao se considerar entropia máxima dos bits da região de busca, infere-se p = 0,015625. Definindo q = 21, foi obtida a função densidade de probabilidade para o total de mudanças. Observa-se que TC pode assumir um valor entre 0 e 336. Os valores das componentes de $\overrightarrow{a_i}$ foram calculados com a utilização do algoritmo de partição de inteiros ZS2 [23]. A Fig. 5 exibe a função densidade de probabilidade em todo seu intervalo enquanto a Fig. 6 exibe apenas as regiões de maiores probabilidades.

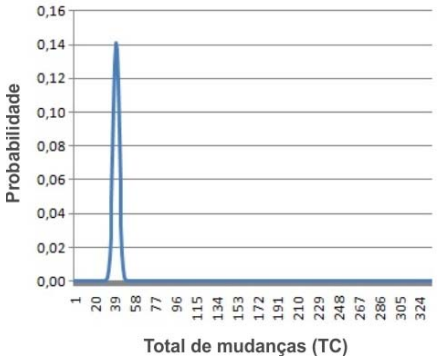

Figura 5. Função densidade de probabilidade do total de mudanças.

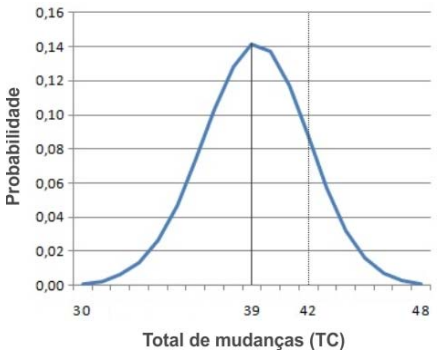

Figura 6. Função densidade de probabilidade do total de mudanças com ênfase nas regiões de maiores probabilidades.

O valor de TC mais provável é o 39, com uma probabilidade de aproximadamente 0,141. Com a utilização de 21 PSs é possível ocultar 126 bits. Para um número de modificações por pixel menor do que 0,333 o total de mudanças tem que ser um valor menor do que 42 bits. De acordo com a função densidade, Prob(TC < 42) ≈ 0,797. O ENMPP da situação mais provável foi de aproximadamente 0,310. O conjunto universo obteve uma probabilidade virtualmente coerente, de valor aproximadamente igual a 0,999. A imprecisão do cálculo foi atribuída à limitação do sistema utilizado em representar números fracionais.

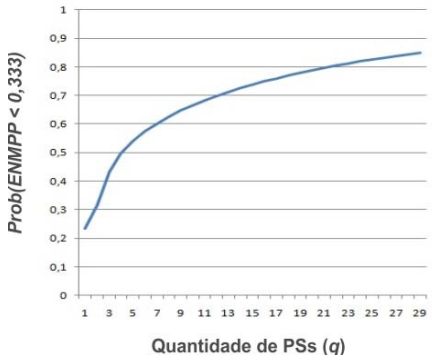

Figura 7. Prob(ENMPP < 0,333) versus Quantidade de PSs (q).

A Fig. 7 apresenta a probabilidade de o método proposto obter uma taxa de modificações menor do que 0,333, para diferentes valores de q. É possível notar que a probabilidade supera a taxa de 80% para valores de q maiores que 21.

IV. Resultados e Discussões

Para a obtenção dos resultados foram utilizadas imagens monocromáticas, convertidas a partir de imagens coloridas disponíveis no Instituto de Processamento de Sinais e Imagem, da Universidade do Sul da Califórnia (USC-SIPI, *University of Southern California – Signal and Image Processing Institute*) [24]. Foi utilizado um subconjunto de 32 imagens, composto por: sub-coleções 1.4 e 1.5; imagens 4.1.07, 4.1.08 e *gray21.512*; coleção *motion*.

Os métodos contemplados na simulação foram: LSB Matching, LSB Matching Revisited, LSB One-Third e o método proposto, LSB Word-Hunt. O sequenciamento pesudo-aleatório para os métodos foi realizado usando o algoritmo LFSR [25].

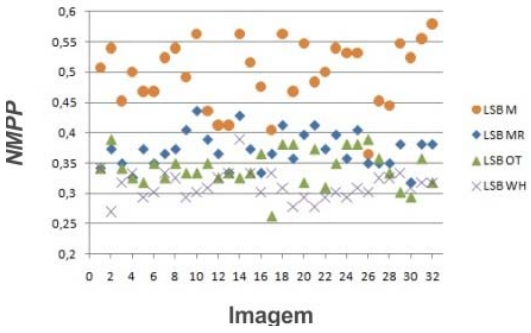

Figura 8. Gráfico de dispersão para número de modificações por pixel (NMPP), considerando o conjunto das 32 imagens utilizadas nas simulações ocultando 126 bits em cada uma delas.

A Fig. 8 apresenta um gráfico de dispersão para o número de modificações por pixel, onde cada imagem ocultou uma informação de alta entropia de seus 126 bits, valor referente à projeção realizada na Seção III, para cada um dos quatro métodos contemplados. O eixo vertical indica o número de modificações por pixel (NMPP, *number of modifications per pixel*), métrica base para a obtenção do ENMPP, que indica o real número de alterações na imagem relacionado com a carga da informação embutida (α). Quanto menor o NMPP mais similar à imagem de cobertura é a estego-imagem.

De acordo com as probabilidades apresentadas na Seção III, a maioria das imagens codificadas pelo LSB WH teve um valor de TC próximo a 39, valor para o qual a probabilidade do total de mudanças, conforme Fig. 5, atinge um valor de pico aproximadamente igual a 0,141, o que resulta um NMPP de aproximadamente 0,310. O menor NMPP obtido pelo método foi de aproximadamente 0,270, enquanto o maior, aproximadamente 0,389. A média teve o valor aproximado de 0,313, com o desvio padrão de aproximadamente 0,023. O LSB M apresentou, para o NMPP, uma média de 0,497, com desvio padrão de 0,054. O LSB MR apresentou uma média de 0,372, com desvio padrão de 0,029. O LSB OT apresentou uma média de 0,341, com desvio padrão de 0,030.

Um segundo conjunto de simulações foi realizado, correspondendo à ocultação de uma informação de alta entropia ocupando toda a carga da imagem utilizada, quando aplicada ao LSB WH, *i. e.* α = 1, e ocupando uma carga α = 0,375 para os demais métodos. A Fig. 9 exibe os resultados.

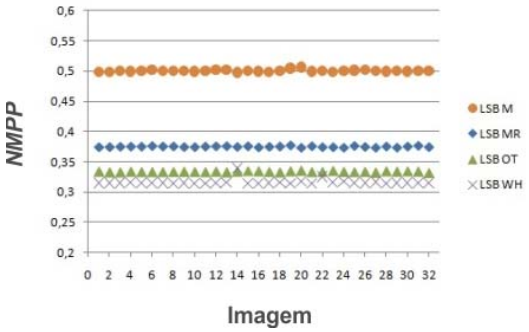

Figura 9. Gráfico de dispersão para modificações por pixel aplicado à ocultação com carga α = 1 para o método LSB WH e α = 0,375 para os métodos LSB M, LSB MR e LSB OT.

Com um maior uso da carga útil observou-se uma redução do desvio padrão do NMPP. O LSB M apresentou uma média de 0,5, com desvio padrão de 0,002. O LSB MR apresentou uma média de 0,375, com desvio padrão de 0,001. O LSB OT apresentou uma média de 0,333, com desvio padrão de 0,001. O LSB WH apresentou uma média de 0,316, com desvio padrão de 0,005. Este desvio foi ocasionado principalmente pela entropia dos bits da região de busca das imagens.

Na Tabela II são apresentados resultados adicionais relacionados ao segundo conjunto de simulações. A tabela contempla, adicionalmente, duas métricas auxiliares: a relação sinal-ruído de pico (PSNR, *Peak signal-to-noise ratio*) e a probabilidade de a imagem esconder dados segundo o Ataque do Qui-Quadrado (Prob(X^2)). A PSNR é uma conhecida métrica no campo de processamento de imagem: quanto maior seu valor mais similar à imagem de cobertura é a estego-imagem. Ainda, como informação complementar, foi calculada a entropia dos bits na região de busca ($H(R)$), para uso específico do método LSB WH. As imagens podem ser visualizadas na Fig. 10.

TABELA II. RESULTADOS ADICIONAIS.

Imagem	Métrica	LSB M	LSB MR	LSB OT	LSB WH	$H(R)$
1.4.01	NMPP	0,4988	0,3742	0,3339	0,3153	0,999
	PSNR	127,59	130,46	131,60	132,24	
	Prob(X^2)	0	0	0	0	
1.4.02	NMPP	0,4993	0,3745	0,3326	0,3151	1
	PSNR	127,58	130,45	131,64	132,24	
	Prob(X^2)	0	0	0	0	
1.5.02	NMPP	0,4979	0,3747	0,3337	0,3393	0,896
	PSNR	127,36	130,12	131,25	131,49	
	Prob(X^2)	0	0	0	0	
1.5.03	NMPP	0,5006	0,3759	0,3346	0,3141	1
	PSNR	127,55	130,41	131,57	132,27	
	Prob(X^2)	0,13789	0,00576	0,00113	0,00003	
1.5.05	NMPP	0,4986	0,3749	0,3333	0,3152	1
	PSNR	127,59	130,44	131,62	132,25	
	Prob(X^2)	0,00088	0,00045	0,00099	0	
1.5.06	NMPP	0,5010	0,3755	0,3324	0,3159	1
	PSNR	127,55	130,43	131,65	132,22	
	Prob(X^2)	0	0	0	0	

Quando analisado todo o conjunto, pôde-se observar que para $H(R) \approx 1$, o ENMPP para o método LSB WH é 0,315, aproximadamente. Como a imagem 1.5.02 possui $H(R) \approx 0,9$, obteve-se para ela um número de modificações por pixel superior a 0,315. Precisamente, foi obtido um NMPP de 0,3393 para a imagem 1.5.02 com o método proposto. A PSNR apresentou uma forte correlação negativa com o NMPP. Para o Ataque do Qui-Quadrado, todos os métodos apresentaram robustez. Apesar de o LSB WH alterar exclusivamente os bits menos significativos, a etapa de manutenção do histograma foi capaz de evitar os problemas causados pelos pares de valores.

O consumo médio de tempo, em relação ao processo de ocultação de informação do LSB WH, foi de 1,2 s por KB de informação oculta. Os processos de ocultação e extração para os métodos LSB M, LSB MR e LSB OT, e o processo de extração de informação do método LSB WH obtiveram um consumo inferior a 25 ms por KB de informação oculta. Os valores de tempo supramencionados foram obtidos por meio de simulações realizadas com um *Notebook Acer* de processador *Intel Core* i3-370M e 2 GB de memória RAM do tipo DDR3, no sistema operacional *Windows 7*.

V. CONCLUSÕES

Este artigo apresentou uma melhoria do método proposto em [7], que não foi originalmente concebido para possuir um baixo ENMPP. O método apresentado neste trabalho foi proposto tendo como alvo a redução do número esperado de modificações por pixel, quando comparado com outros métodos da literatura. Esta melhoria foi possível em detrimento da carga útil do sistema, que sofreu uma redução de 62,5% em relação aos demais métodos abordados.

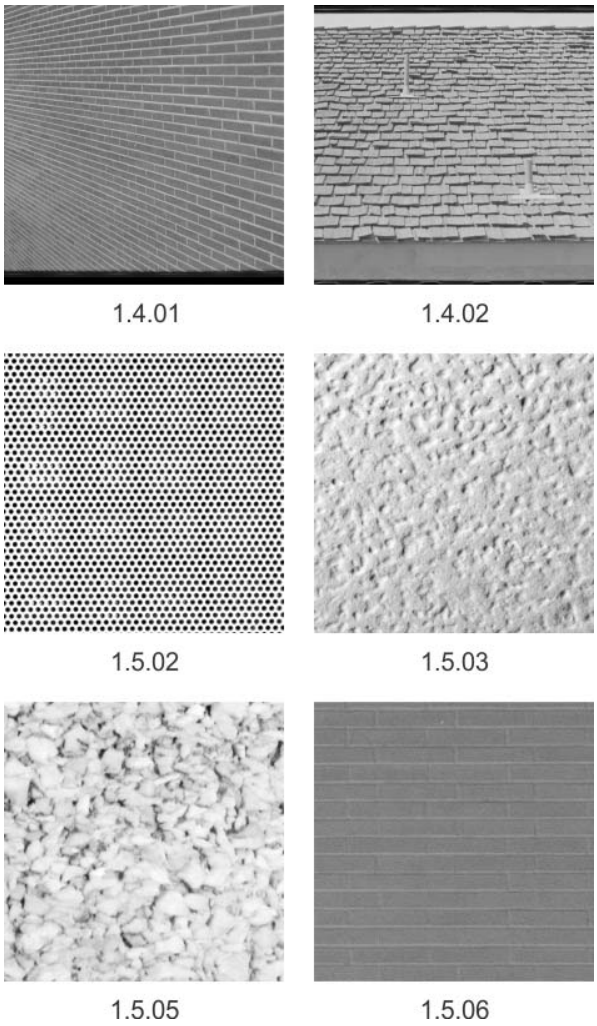

Figura10. Imagens de cobertura referenciadas na Tabela II.

O ENMPP para o novo método apresentou dependência de uma característica da imagem utilizada: a entropia dos bits para os planos 2-LSB e 3-LSB. O conjunto de imagens usadas para a simulação indicou que a maioria delas possuía uma entropia adequada para a aplicação. Imagens com tal entropia próxima a 1 tenderam a um ENMPP de 0,315. Para o método apresentado em [7] observa-se, em geral, valores de NMPP compreendidos entre 0,337 e 0,601.

Apesar de o novo método alterar única e exclusivamente bits do plano menos significativo, a nova abordagem foi capaz de evitar o problema dos pares de valores, ao aplicar uma manutenção ao histograma da imagem.

O tempo gasto pelo método proposto para a ocultação de informação em imagens digitais mostrou-se maior que o tempo gasto pelos métodos LSB M, LSB MR e LSB OT.

A redução do número de modificações por pixel é um forte aliado para a segurança, mas, por si só, não a garante. Estudos futuros incluem simulações para outros ataques de esteganálise, adaptando o método, se necessário, manipulando a ordem de teste dos feixes para aumentar a qualidade da ocultação.

AGRADECIMENTOS

Os autores agradecem o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e da Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE).

REFERÊNCIAS

[1] J. Fridrich, *Steganography in digital media: Principles, algorithms, and applications*. Cambridge University Press, 2009.

[2] A. Cheddad, K. Condell e P. Mc Kevitt, *Digital image steganography: Survey and analysis of current methods*, IEEE Signal Processing, vol. 90, nº 3, pp. 727-752, 2010.

[3] B. Li, J. He, J. Huang e Y. Q. Shi, *A survey on image steganography and steganalysis,* Journal of Information Hiding and Multimedia Signal Processing, vol. 2, nº 2, pp. 142-172, 2011.

[4] T. Sharp, *An implementation of key-based digital signal steganography*, in Proc. Information Hiding Workshop, vol. 2137, pp. 13-26, 2001.

[5] J. Mielikainen, *LSB matching revisited*, IEEE Signal Processing Letters, vol. 13, nº 5, pp. 285-287, 2006.

[6] S. Sarreshtedari, M. Ghotbi e S. Ghaemmaghami, *One-third probability embedding: Less detectable LSB steganography*, IEEE International Conference on Multimedia and Expo, pp. 1002-1005, 2009.

[7] J. R. Tavares, J. B. Lima e F. Madeiro, *LSB word-hunt: Um método de esteganografia para imagens digitais utilizando chave simétrica*, Anais do Congresso de Matemática Aplicada e Computacional, pp. 158-161, 2012.

[8] X. Li, B. Yang, D. Cheng e T. Zeng, *A generalization of LSB matching*, IEEE Signal Processing Letters, vol. 16, nº 2, pp 69-73, 2009.

[9] K. Ghazanfari, S. Ghaemmaghami, e S. Khosravi, *LSB++: an improvement to LSB+ steganography*, in TENCON 2011 - IEEE Region 10 Conference, pp. 364-368, 2011.

[10] K. Qazanfari, R. Safabakhsh, *A new steganography method which preserves histogram: Generalization of LSB++*, Information Sciences, vol 277, pp 90-101, 2014.

[11] A. Westfeld e A. Pfitzmann, *Attacks on steganographic systems - breaking the steganographic utilities Ezstego*, in Jsteg, Steganos, and S-Tools - and Some Lessons Learned, Lecture Notes in Computer Science. Springer-Verlag, pp. 61-75, 2000.

[12] D. Lerch-Hostalot e D. Megías, *LSB matching steganalysis based on patterns of pixel diferences and random embedding,* Computers & Security, vol. 32, pp. 192-206, 2013.

[13] W. Y. Ng, Z. He, D. S. Yeung e P. K. Chan, *Steganalysis classifier training via minimizing sensitivity for different imaging sources*, Information Sciences, vol. 281, pp 211-224, 2014.

[14] R. Cogranne e F. Retraint, *Application of hypothesis testing theory for optimal detection of LSB matching data hiding*, Signal Processing, vol. 93, pp 1724–1737, 2013.

[15] D. Lerch-Hostalot e D. Megías, *LSB matching steganalysis based on patterns of pixel differences and random embedding*, Computers & Security, vol. 32, pp 192–206, 2013.

[16] F. G. Mohammadi e M. S. Abadeh, *Image steganalysis using a bee colony based feature selection algorithm*, Engineering Applications of Artificial Intelligence, vol. 31, pp 35–43, 2014.

[17] R. Cogranne, F. Retraint, C. Zitzmann, I. Nikiforov, L. Fillatre e P. Cornu, *Hidden information detection using decision theory and quantized samples: Methodology, difficulties and results*, Digital Signal Processing, vol. 24, pp 144–161, 2014.

[18] L. Fillatre, *Adaptive steganalysis of least significant bit replacement in grayscale natural images*, IEEE Transactions on Signal Processing, vol. 60, nº 2, pp 556-569, 2012.

[19] S. Tan, *Targeted steganalysis of edge adaptive image steganography based on LSB matching revisited using b-spline fitting*, IEEE Signal Processing Letters, vol. 19, nº 6, pp 336-339, 2012.

[20] C. Yang, F. Liu, X. Luo e Y. Zeng, *Pixel group trace model-based quantitative steganalysis for multiple least-significant bits steganography*, IEEE Transactions on Forensics and Security, vol. 8, nº 1, pp 216-228, 2013.

[21] R. Cogranne e F. Retraint, *An asymptotically uniformly most powerful test for LSB matching detection*, IEEE Transactions on Information Forensics and Security, vol. 8, nº 3, pp 464-476, 2013.

[22] X. Hou, T. Zhang, G. Xiong e B. Wan, *Forensics aided steganalysis of heterogeneous bitmap images with different compression history*, KSII Transactions on Internet and Information Systems, vol. 6, nº 8, pp 1926-1945, 2012.

[23] A. Zoghbi e I. Stojmenovic, *Fast algorithms for generating integer partitions*, International Journal of Computer Mathematics, Tech. Rep., 1994.

[24] A. G. Weber, *The USC-SIPI image database: version 5*, USC-SIPI report #315, Tech. Rep., 1994.

[25] P. Alfke, *Eficient shift registers, LFSR counters, and long pseudo-random sequence generators,* Tech. Rep., application Note, Xilinx Corp, 1995.

Rafael Tavares, pernambucano, nasceu no dia 7 de Julho de 1987. Recebeu o título de Bacharel em Ciências da Computação pela Universidade Católica de Pernambuco (UNICAP), Brasil, em 2011. Recebeu o título de Mestre em Engenharia de Sistemas pela Universidade de Pernambuco (UPE), Brasil, em 2014. Seus principais interesses em pesquisa incluem processamento de sinais, segurança da informação e inteligência computacional.

Francisco Madeiro nasceu em Fortaleza, Ceará, Brasil, em 1972. Recebeu o título de Doutor em Engenharia Elétrica pela Universidade Federal da Paraíba (UFPB), Brasil, em 2001. Atualmente é Professor Associado da Universidade de Pernambuco (UPE), Brasil. Seus principais interesses de pesquisa incluem processamento de sinais, sistemas de comunicação e inteligência computacional. Foi ganhador do prêmio "Destaque em Ensino" da Escola Politécnica de Pernambuco (POLI/UPE), em 2008, e dos prêmios de "Destaque em Pesquisa" e "Destaque em Ensino" da POLI/UPE, em 2013. Tem atuado em projetos de pesquisa e desenvolvimento (P&D) em transmissão digital e processamento de imagem. Desde 2012 é bolsista de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora (DT) do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brasil.