2018-2019春季
信息隐藏课程
第8讲 湿纸编码

中国科学院信息工程研究所
INSITUTE OF INFORMATION ENGNEERING CAS

SKLOIS
信息安全国家重点实验室

赵险峰

中国科学院信息工程研究所
信息安全国家重点实验室

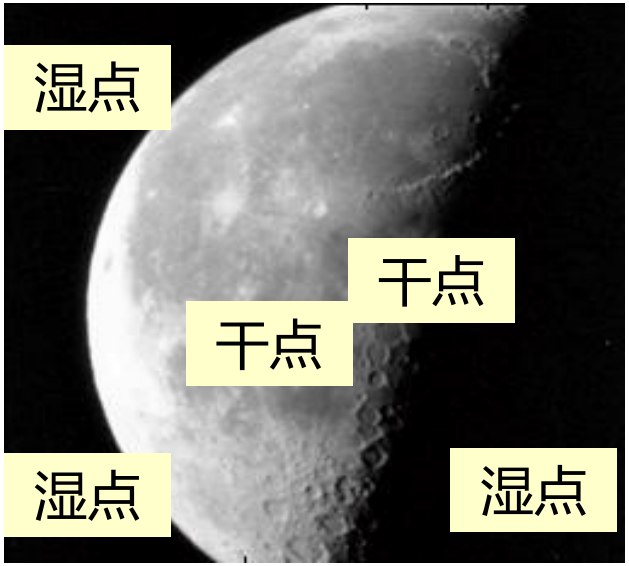
2018年11月

纲要

1. 基本概念
2. 典型“湿点”与“干点”
3. 编码设计准备
4. 湿纸编码基本算法
5. 湿纸编码隐写
6. 文献阅读推荐

1-1 基本概念（问题）

- 为了提高隐写安全，一般希望能够选择嵌入位置，在更隐蔽区域的样点中嵌入信息，而在不够隐蔽处不嵌入或者少嵌入
- 不妨把可以隐写的样点称为“干点（Dry Point）”，把不准备隐写的样点称为“湿点（Wet Point）”。如果每次仅仅在干点上嵌入信息，干点被修改后并不一定满足干点的性质；另外，干点的位置选择也是动态的，因此，若不做相应处理，隐写的接收者将不能提取信息
- 湿纸编码（Wet Paper Coding）是一类解决以上问题的方法



1-2 基本概念（思路）

- 令$\boldsymbol{x}=(x_1,x_2,\cdots,x_n),x_i\in \mathrm{GF}(2)$表示长度为$n$的载体LSB值序列，$\boldsymbol{m}=(m_1,m_2,\cdots,m_q),m_i\in \mathrm{GF}(2)$表示长度为$q$的隐蔽信息比特序列，$\boldsymbol{y}=(y_1,y_2,\cdots,y_n),y_i\in \mathrm{GF}(2)$表示嵌入隐蔽信息后的隐文LSB样点值序列，则湿纸编码通过以下方式传输$\boldsymbol{m}$：仅仅修改$\boldsymbol{x}$的干点使之变为$\boldsymbol{y}$，并使得$H\boldsymbol{y}=\boldsymbol{m}$，其中，$H$为收发双方共享的矩阵（密钥）
- **湿纸编码是一种支持自适应隐写的编码，这种自适应体现在，每次选择嵌入的样点是基于当前载体临时评估的，对不同载体选择的嵌入位置是不同的，这有利于将信息隐藏在被检测风险较小的“干点”区域**

$$H\boldsymbol{y}=H\begin{bmatrix}y_1\\y_2\\\vdots\\y_i\\\vdots\\y_n\end{bmatrix}=\boldsymbol{m}$$

1-3 基本概念（特点）

- MME隐写算法与湿纸编码隐写均体现了一定的自适应隐写思想，MME在全部等价修改方法（可修改≥1个位置）中选择能量最低的方法，但是，MME限于在一个分组中实现嵌入位置的优化选择，且选择的嵌入位置受到较强的**代数关系制约**
- 而湿纸编码能够实现更大范围的位置选择，**编码的制约是，通过修改干点值，满足以下提取方程，制约一般比较宽松**

$$H\boldsymbol{y} = H\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} = \boldsymbol{m}$$

在湿纸编码中，可修改的点（干点）是预先根据点的“好坏”标准设定的，而矩阵编码（含MME）则不能确定，只能在满足代数关系的位置中选取好的点

2-1 典型“湿点”与“干点”（低扰动量化点）1

- “干点”是更适合嵌入的载体样点，“湿点”是更不适合嵌入的载体样点。目前存在判别载体样点是否适合嵌入的多种评估标准，比较典型的评估方法考察了隐写引入的隐写噪声或者被检测风险
- 在前面介绍MME算法时，介绍了隐写对JPEG编码量化的利用：在矩阵编码的等价嵌入方式中，选择使得隐写噪声最小的嵌入方式
- 如果类似MME那样，隐写算法的输入是一个光栅图像，输出是一个JPEG图像，则从降低隐写噪声的角度评价，更适合嵌入的位置是在量化取整前小数部分处于0.5附近的系数
- 令 d_i表示分块DCT系数在JPEG编码量化取整前的数值（不妨设它们大于0），显然 d_i 的小数部分$d_i - \lfloor d_i \rfloor$如果在0.5附近的位置更适合嵌入，即 ε 是一个很小的数，则满足$d_i - \lfloor d_i \rfloor \in [0.5 - \varepsilon, 0.5 + \varepsilon]$的位置更适合嵌入

2-2 典型“湿点”与“干点”（低扰动量化点）2

- 设$d_i - \lfloor d_i \rfloor = 0.5 - \gamma$，$0 \le \gamma \le \varepsilon$，如果该点由于隐写需要被修改（即向另外的方向取整数，而不是保留其JPEG编码的数值），隐写噪声为2γ，在均匀分布下，平均隐写修改噪声是

$$\int_0^{\varepsilon} 2\gamma \times \frac{1}{\varepsilon - 0} d\gamma = \frac{\gamma^2}{\varepsilon}\Big|_0^{\varepsilon} = \varepsilon$$

- 这说明，ε越小隐写平均修改噪声的幅度越小。因此，可以在总共n个系数中选择以下系数位置集合作为干点（**低扰动量化点**）：

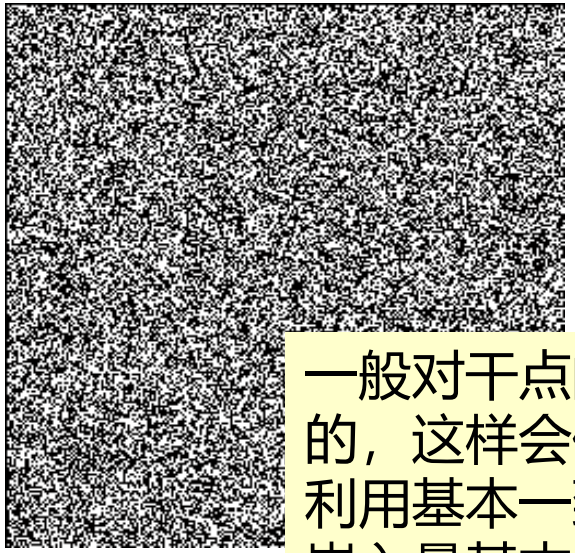
$$S = \{i \mid i \in \{1, \cdots, n\}, d_i - \lfloor d_i \rfloor \in [0.5 - \varepsilon, 0.5 + \varepsilon]\}$$

- 其他系数样点为湿点

2-3 典型“湿点”与“干点”（高纹理点）

- 由于从纹理复杂度较高的区域提取的隐写分析特征较难被分类，因此，研究人员普遍认为这类区域中的样点更适合嵌入。假设$N(x_i)$表示样点x_i的邻域，而$c(N(x_i))$表示该样点的邻域复杂度，设 C 是一个阈值，则可以在总共 n 个系数中选择以下系数位置集合作为干点：

$$S = \{i \mid i \in \{1, \cdots, n\}, c(N(x_i)) > C\}$$



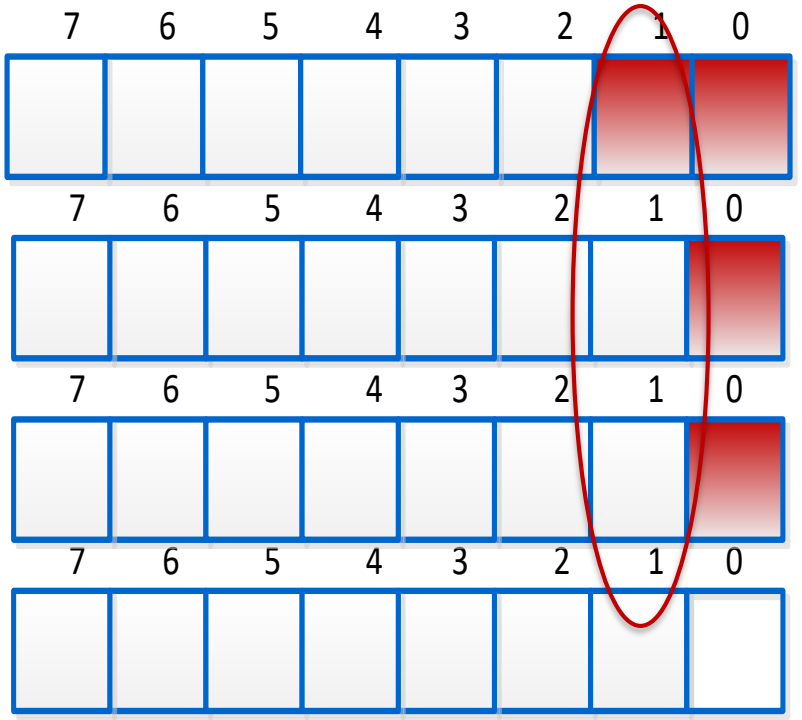
一般对干点的利用是在置乱域进行的，这样会使得每个区域的处理与利用基本一致（如每个同大小区域嵌入量基本一致），也有利与保护各种嵌入参数的安全

2-4 典型“湿点”与“干点”（第二层可利用点）

- 当基本嵌入采用随机加减1修改的LSBM时，当嵌入完毕时，如果还要继续嵌入**次LSB**，则LSB修改位置上的次LSB均为干点，这个集合可以表示为

$$S = \{i \mid i \in \{1, \cdots, n\}, y_i - x_i = 1\}$$

- 其他位置为湿点。以上干点的确定原则是，改动这些位置上的次LSB可以通过控制上一层LSBM是加1还是减1实现，因此并没有增加修改能量



3-1 编码原理（隐写码一般表示）

- 给定载体$\boldsymbol{x}$和隐蔽信息$\boldsymbol{m}$，隐写编码过程 $\boldsymbol{x} \to \boldsymbol{y}$ 可表示为寻找满足如下条件的 $\boldsymbol{y}$

$$\min_{\boldsymbol{y}} D(\boldsymbol{x}, \boldsymbol{y}) \text{，} \quad \text{s.t.} f(\boldsymbol{y}) = \boldsymbol{m}$$

- 映射f由嵌入者和接收者秘密共享；函数$D(\cdot,\cdot)$衡量了嵌入前后载体的差异或者被检测代价，隐写编码在多个满足$f(\boldsymbol{y}) = \boldsymbol{m}$的$\boldsymbol{y}$中，选择其中使$D(\boldsymbol{x}, \boldsymbol{y})$最小的$\boldsymbol{y}$作为嵌入后的载体
- 如，对于矩阵编码类的隐写方法有$f(\boldsymbol{y}) = H\boldsymbol{y} = \boldsymbol{m}$，$H$为校验矩阵，$D(\boldsymbol{x}, \boldsymbol{y})$为嵌入过程中**分组**修改位置的数量；MME也有有$f(\boldsymbol{y}) = H\boldsymbol{y} = \boldsymbol{m}$ 的形式，$D(\boldsymbol{x}, \boldsymbol{y})$是嵌入造成的系数幅值偏差；在以后的课程中将看到，基于STC编码的自适应隐写也可以用类似的矩阵形式表达，但是，$D(\boldsymbol{x}, \boldsymbol{y})$是嵌入引起的局部失真代价**总和（整体最小）**
- 以上解码者均无需关心隐写者修改的具体位置

3-2 编码原理（湿纸码一般表示）1

- 湿纸编码也可以用上述形式表示。其具体特点是，在构造 $H\boldsymbol{y}=\boldsymbol{m}$ 的过程中，隐写算法仅仅修改了 $\boldsymbol{x}$ 的干点使之变为 $\boldsymbol{y}$，并且通过限定修改位置为干点降低 $D(\boldsymbol{x},\boldsymbol{y})$
- 设 $\boldsymbol{x}=\{x_1,x_2,\cdots,x_n\}$ 中，第 $p_1,p_2,\cdots,p_k$ 个位置上为干点，其他为湿点，则嵌入修改后应满足：

$$\begin{pmatrix} h_{11} & \cdots & h_{1p_1} & \cdots & h_{1p_k} & \cdots & h_{1n} \\ h_{21} & \cdots & h_{2p_1} & \cdots & h_{2p_k} & \cdots & h_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{q1} & \cdots & h_{qp_1} & \cdots & h_{qp_k} & \cdots & h_{qn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ y_{p_1} \\ \vdots \\ y_{p_k} \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} m_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ m_q \end{pmatrix}$$

消息长度= q，干点数= k

3-3 编码原理（湿纸码一般表示）2

- 可以发现以上等价于以下缩减的形式：

$$\begin{cases} h_{1p_1}y_{p_1} + \cdots + h_{1p_k}y_{p_k} = c_1 \\ \vdots \\ h_{qp_1}y_{p_1} + \cdots + h_{qp_k}y_{p_k} = c_q \end{cases}$$

- 以上 c_i 在给定 $\boldsymbol{x}$ 和 $\boldsymbol{m}$ 下是常数。以上性质可以等价表示为$H\boldsymbol{v} = H(\boldsymbol{y}-\boldsymbol{x}) = \boldsymbol{m} - H\boldsymbol{x}$，其中，$\boldsymbol{v} = \boldsymbol{y} - \boldsymbol{x}$，由于不是干点的位置 $v_i = 0$，上式也可以表示为缩减的形式：

$$\bar{H}\bar{\boldsymbol{v}} = \boldsymbol{m} - H\boldsymbol{x}$$

- 在上式中，$\bar{H}$ 是 $q \times k$ 矩阵，$\bar{\boldsymbol{v}}$ 中仅仅保留了$\boldsymbol{v}$ 中干点位置上的 k 个元素，$\bar{H}$保留了相应的列向量。显然，如果通过调节$\boldsymbol{y}$控制$\bar{\boldsymbol{v}}$使得以上方程有解，湿纸编码成功
- 由于在$\mathrm{rank}(\bar{H}) = q$ 下以上方程有解，因此，希望有一种方法能够尽可能确保这个性质；直观上，消息长度 q 与干点数 k 相当

3-4 编码容量（湿纸码方程一般有解依据）

- 设$P_{q,k}(s)$是任一随机$q \times k$矩阵的秩为s的概率，则根据相关文献的推导有

$$P_{q,k}(s) = 2^{s(q+k-s)-qk} \prod_{i=0}^{s-1} \frac{(1-2^{i-q})(1-2^{i-k})}{(1-2^{i-s})}$$

- 可以验证，对一个较大的干点数量（列数）$k > q$（行数），当减少消息长度q时，$P_{q,k}(q)$快速趋近1
- **据此，在湿纸编码中可以通过动态减小分块上消息的长度来使得可以成功嵌入，但这也意味着需要动态记录与传输消息长度**

- R. P. Brent, S. Gao, A. G. B. Lauder, “Random Krylov spaces over finite fields,” *SIAM J. Discrete Math.*, vol. 16, no. 2, pp. 276–287, 2003.

3-5 编码容量（湿纸码容量理论分析）1

- 设$p_{\geq k-r}$表示 k 个干点至少能传输 $k-r=q$比特消息的可能(消息数小于等于干点数)，包括两类情况：$\bar{H}_{q\times k}\bar{\boldsymbol{v}}=\boldsymbol{m}-H\boldsymbol{x}$ 中$\bar{H}$的秩为$k-r$，或者$\bar{H}$的秩为$k-r-i$ 但是其中 i 个线性相关行乘以 $\bar{\boldsymbol{v}}$ 后正好与方程右侧0/1对应，这个概率是2^{-i}。因此

$$p_{\geq k-r}=\sum_{i=0}^{k-r}\frac{1}{2^i}P_{k-r,k}(k-r-i)$$

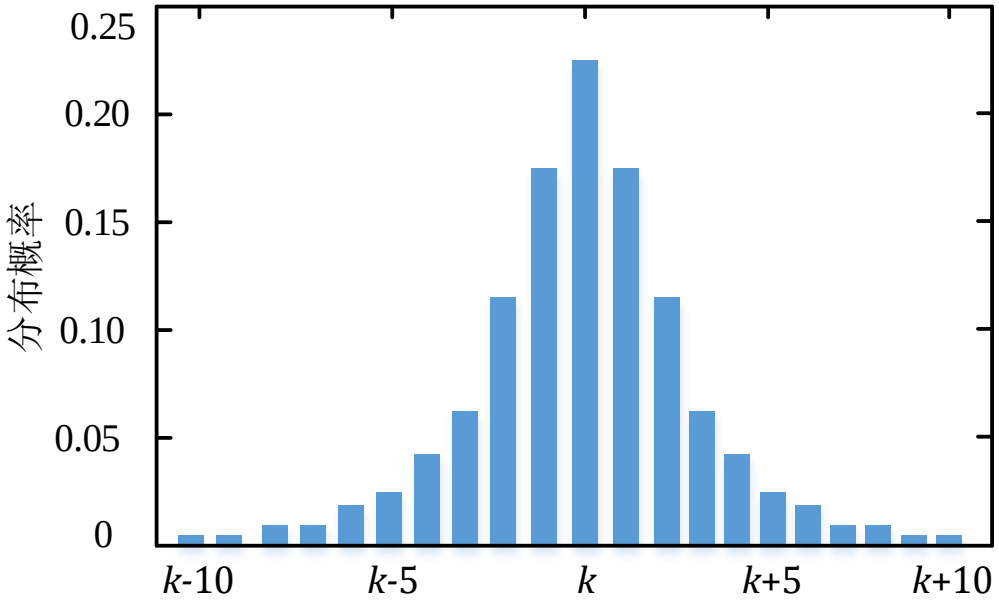
- 若至少能传输 $k+r=q$比特消息(消息数大于干点数)，则包括两类情况：当$\bar{H}$的秩为 k （不能超过列数），r个线性相关行乘以 $\bar{\boldsymbol{v}}$ 后正好与方程右侧对应；或者当$\bar{H}$的秩为$k-i$，但$r+i$ 个线性相关行乘以 $\bar{\boldsymbol{v}}$ 后正好与方程右侧0/1对应，即有

$$p_{\geq k+r}=\sum_{i=0}^{k}\frac{1}{2^{r+i}}P_{k+r,k}(k-i)$$

3-6 编码容量（湿纸码容量理论分析）1

- 基于上两式，可以得到基于k个干点，能够嵌入 i 个比特消息的概率$p_{=i} = p_{\geq i} - p_{\geq i+1}$
- 实验表明，以上$p_{=i}$的分布在$i = k$处取最大值，并且以此处左右对称（图），因此，基于k个干点能够嵌入的最大比特数量平均为$q_{\max}(k) = \sum_{i=0}^{\infty} i\, p_{=i} \approx k$

$p_{=i}$的分布在$i = k$处取最大值，并且以此处左右对称



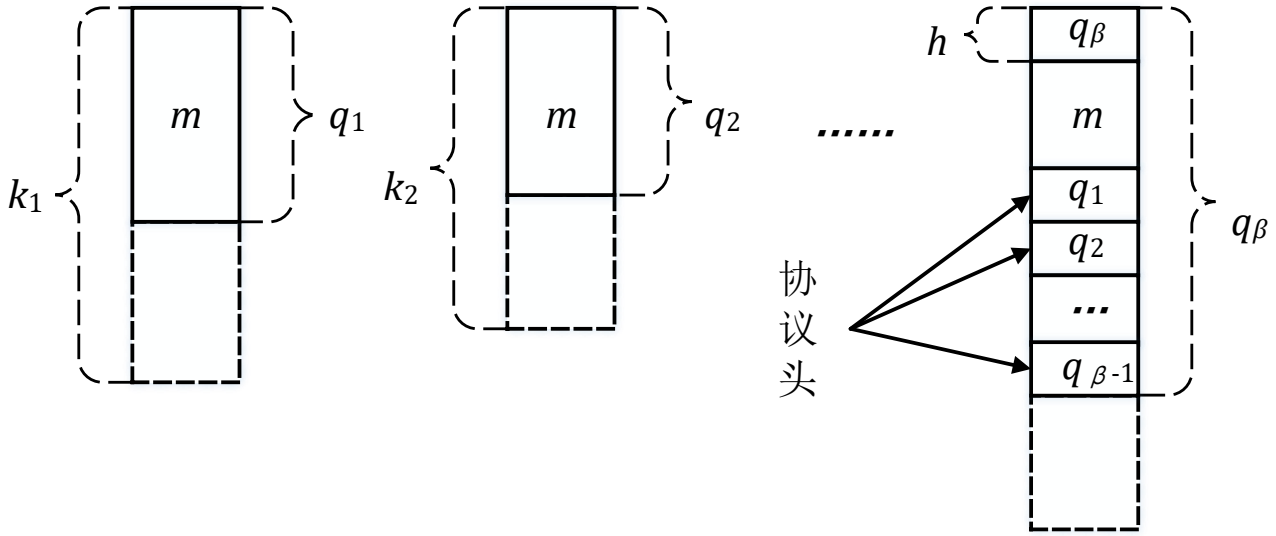
4-1 湿纸编码算法（分段原则）

- 湿纸编码隐写的嵌入等价于求解 k 个变元、q 个方程的方程组，一般 $q \approx k$，因此，计算复杂度等价于高斯消元法，是$O(k^3)$
- 由于 k 与可嵌入的样点数量或者需要嵌入消息的比特数量是相当的，例如 $k > 10^6$，将图像可嵌入点作为一个分块进行湿纸编码在计算上是困难的。因此，湿纸编码算法原则上需要将待嵌入的消息序列 $\boldsymbol{m}$ 进行分段，也意味着要对置乱后的载体分段

4-2 湿纸编码算法（分段方法）

- 湿纸编码需要一些先验知识确定分段数 β
- 编码方需要首先确定干点的比率 $r = k/n$，它受载体尺寸以及干点数量、通信需求等因素影响。设隐写收发双方根据经验可知道该比率范围为 $r_1 \le r \le r_2$，这里不妨确定为 $r = r_2$；当然，这个比率也可以由发送方单独确定并将信息与消息一并发送
- 假设k_{avg}表示适合在一个分段中利用的干点数量（约为消息分段数量，是与计算能力匹配的长度），则分段数可以表示为 $\beta = \lceil nr_2/k_{\text{avg}} \rceil$，实际上是按照干点数量的估计值确定段数
- 以下记每个载体分段为$\boldsymbol{x}^{(i)}$，其样点数为 $n_i \in \{\lfloor n/\beta \rfloor, \lceil n/\beta \rceil\}$，干点数量为 k_i，$n_1 + n_2 + \cdots + n_\beta = n$（$n_i$仅仅相差1）
- 由于载体数据在分段前需要位置置乱，因此，可以认为干点数量较为均匀地分布在各个分段中，即 k_i、n_i较为接近，这也保证了编码处理与特性在各个分段上基本相同

4-3 湿纸编码算法（消息与协议分段结构）



- 湿纸编码算法嵌入的消息与协议信息结构及其对干点的分段利用：每段嵌入的消息数量填入未嵌入消息末尾；对最后一段消息的处理是，剩余消息前加上剩余消息长度+h，一并作为最后一块的消息
- 算法需要对全部消息与协议数据的大小进行有效预估，并动态确定每一段的可嵌入消息量

4-4 一种湿纸编码算法（编码算法）1

0. 计算分块数$\beta = \lceil nr_2/k_{\text{avg}} \rceil$，利用PRNG生成GF(2)上的矩阵$H$，它有$n/\beta$列以及足够数量的行；//$\beta$在一定范围均可

1. 估算每个分段的协议头尺寸$h = \lceil \log_2(nr_2/\beta) \rceil + 1$,计算嵌入消息与协议头信息的总长度$q = |\boldsymbol{m}| + \beta h$；//分段的协议头用于存储每个分段嵌入的消息长，后者约等于存储干点数量需要的尺寸，以上h有一定余量

2. $\boldsymbol{x}' \leftarrow \boldsymbol{x}, i \leftarrow 1$;

3. 估计当前分段可隐藏的消息量$q_i = \lceil k_i(q+10)/k \rceil$，$q_i = \min\{q_i, 2^{h-1}, |\boldsymbol{m}|\}$,得到需要在本分段隐藏的消息$\boldsymbol{m}^{(i)} \leftarrow \boldsymbol{m}$下面$q_i$个比特；// q_i先按照干点数估计，$q+10$是希望前面尽量多隐藏一些，确保最后一个分段成功嵌入（最后一段不能通过减小q_i保证有解）

4. 取$H^{(i)} \leftarrow H$中的前n_i列与q_i行，求解$\bar{H}^{(i)}\bar{\boldsymbol{v}}^{(i)} = \boldsymbol{m}^{(i)} - H^{(i)}\boldsymbol{x}^{(i)}$，它有$q_i$个方程、$k_i$个变元，$\bar{H}^{(i)}$是$H^{(i)}$的$q_i \times k_i$子矩阵，子矩阵由干点的分布确定；如果无解，则减少$q_i$继续尝试直到成功求解；//根据前面的分析，减少$q_i$能够迅速使得有解

5. 根据求得的$\bar{\boldsymbol{v}}^{(i)}$修改$\boldsymbol{x}^{(i)}$，得到$\boldsymbol{y}^{(i)} = \boldsymbol{x}^{(i)} + \bar{\boldsymbol{v}}^{(i)}$;

4-5 一种湿纸编码算法（编码算法）2

6. 将q_i用h个比特的字段存储，追加到消息$\boldsymbol{m}$的后部；

7. 从$\boldsymbol{m}$中移除以上q_i比特；

8. $q \leftarrow q - q_i$，$k \leftarrow k - k_i$，$i \leftarrow i + 1$；

9. IF $i < \beta$，GOTO 3；//判断分块是否处理完

10. IF $i = \beta$，$q_\beta \leftarrow q$；//处理最后一个分块

11. 将q_β用h个比特的字段存储，提前追加到$\boldsymbol{m}$前，得到新的$\boldsymbol{m}$，确定最后一个分段需要隐藏的信息量$\boldsymbol{m}^{(\beta)} = \boldsymbol{m}$；//最后分段需要将余下的信息全部嵌入，是最后一次嵌入，因此事前需要将协议头准备好；之前的分段尽量多嵌入，一般保证了这里的q_β较小，有利于求解成功

12. 取 $H^{(\beta)} \leftarrow H$ 中的前n_β列与q_β行，求解 $\bar{H}^{(\beta)}\bar{\boldsymbol{v}}^{(\beta)} = \boldsymbol{m}^{(\beta)} - H^{(\beta)}\boldsymbol{x}^{(\beta)}$，它有$q_\beta$个方程、$k_\beta$个变元，$\bar{H}^{(\beta)}$是$H^{(\beta)}$的$q_\beta \times k_\beta$子矩阵，子矩阵由干点的位置确定；如果无解，则嵌入过程失败，需要调整嵌入参数或者更换载体 // 最后一次需要保证全部可嵌入，不可缩减消息尺寸了

13. 根据求得的$\bar{\boldsymbol{v}}^{(\beta)}$修改$\boldsymbol{x}^{(\beta)}$，得到$\boldsymbol{y}^{(\beta)} = \boldsymbol{x}^{(\beta)} + \bar{\boldsymbol{v}}^{(\beta)}$

4-6 一种湿纸码解码算法

0. 计算分块数$\beta = \lceil nr_2/k_{\text{avg}} \rceil$，利用PRNG生成GF(2)上的矩阵$H$，它有$n/\beta$列以及足够数量的行；//可假设计算$\beta$的参数是置乱域的开始字段，由发送者LSB传来

1. 估算每个分段的协议头尺寸$h = \lceil \log_2(nr_2/\beta) \rceil + 1$；

2. $i \leftarrow \beta$；//分段处理次序与编码相反

3. 取$H^{(\beta)} \leftarrow H$中的前n_β列与h行，求得$q_\beta = H^{(\beta)}\boldsymbol{y}^{(\beta)}$；//求得以上11中，在最后一段处理中前置于$\boldsymbol{m}$的$q_\beta$，它一共$h$个比特

4. 取$H^{(\beta)} \leftarrow H$中的前n_β列与第h行后$q_\beta - h$行，求得$\boldsymbol{m} = H^{(\beta)}\boldsymbol{y}^{(\beta)}$；

5. $i \leftarrow i - 1$

6. 从当前$\boldsymbol{m}$的最后h个比特中取出q_i，并将其从$\boldsymbol{m}$中移除；

7. 取$H^{(i)} \leftarrow H$中的前n_i列与q_i行，将$H^{(i)}\boldsymbol{y}^{(i)}$拼接于$\boldsymbol{m}$前：$\boldsymbol{m} \leftarrow H^{(i)}\boldsymbol{y}^{(i)}||\boldsymbol{m}$；

8. IF $i > 1$，GOTO 5；//判断分块是否处理完

9. ELSE $\boldsymbol{m}$即为提取的消息

5-1 湿纸编码隐写PQ

- 基于湿纸编码，结合一种对湿点与干点的划分方法，就可以构造一个湿纸编码隐写
- 最典型的这类隐写是PQ（Perturbed Quantization）。利用湿纸编码，使得隐写修改样点限定于干点集合$S = \{i \mid i \in \{1, \cdots, n\}, d_i - \lfloor d_i \rfloor \in [0.5 - \varepsilon, 0.5 + \varepsilon]\}$中，降低了隐写引入的噪声
- J. Fridrich等人提出，基于湿纸编码可以将基于纹理复杂度选择嵌入位置的方法改造为更好的自适应隐写
- 以后也将发现，基于加减1嵌入的隐写可以低代价地在次LSB中再嵌入一层信息，实现双层嵌入，这类嵌入就利用了湿纸编码的原理

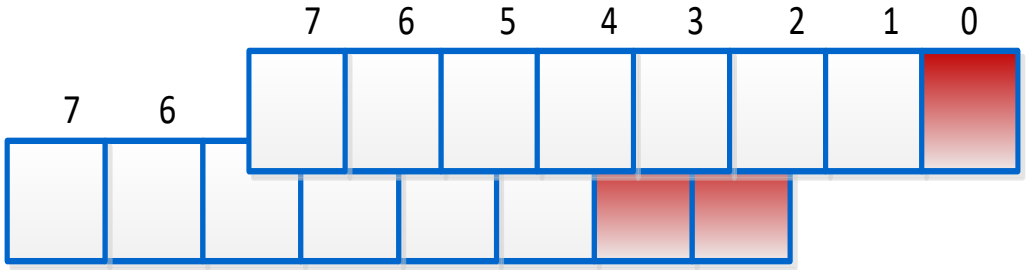
5-2 抗收缩JPEG隐写

- F5隐写算法首次采用了矩阵编码减少修改次数，有很重要的理论意义与应用价值，但是，它仍然存在分布上的“收缩”现象：
  - 当 -1 与 +1 被用于修改时将产生0，由于消息接收者无法区分是原有的0系数还是修改后的，因此隐写者必须继续嵌入直到产生一个非零系数，而接收者要逃过全部0系数
- 抗收缩的nsF5（No-Shrinkage F5）：用湿纸编码代替了矩阵编码，在将置乱后的JPEG系数进行分段以后，将0系数作为湿点、将非零系数作为干点进行湿纸编码，这样，接收者只需要用共享的编码矩阵提取信息，而无需关心0的可能产生情况
  - 实验结果说明，nsF5的抗检测能力明显超过了F5

5-3 ±1双层嵌入（第一层）

- 基于湿纸码可以巧妙地用2个位平面实现双层嵌入，进一步提高嵌入效率。设$L(x_i)$表示x_i的LSB，$S(x_i)$表示其第二LSB位平面；令F代表某矩阵编码方案（采用±1基本嵌入），它的消息嵌入率（亦称负载率）为α，嵌入效率为e，每LSB比特平均修改D次，则$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$能够承载消息$\boldsymbol{m}=(m_1,m_2,\cdots,m_{q=\alpha n})$，不妨设$q=\alpha n$为整数，该方案可表达为：

$$(m_1,m_2,\cdots,m_{\alpha n})=F(L(x_1),L(x_2),\cdots,L(x_n))$$

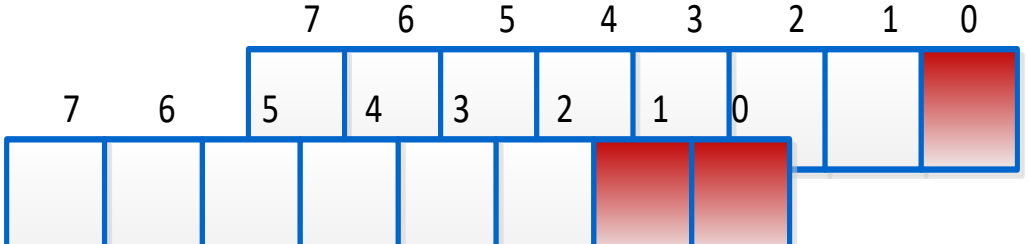


5-3 ±1双层嵌入（第二层）：“e + 1”算法

- 在嵌入中，平均需要修改Dn个比特，但如果采用±1嵌入，在平均Dn个位置上都有选择加1或者减1的两种选择，不同的选择对$S(x_i)$是0还是1影响不同，但对x_i的误差绝对值均是1。令这Dn个位置上的$S(x_i)$为干点，引入湿纸编码方案W，则可以通过调节平均Dn个位置上加1还是减1，使得湿纸编码方案W承载额外的消息，这可以表示为：

$$(m_{\alpha n+1}, m_{\alpha n+2}, \cdots, m_{(\alpha+D)n}) = W(S(x_1), S(x_2), \cdots, S(x_n))$$

- 由于嵌入率从α提高到$\alpha + D$，而每个原文比特平均修改的次数仍然为D，因此，嵌入效率e增加了1，$e = \frac{\alpha+D}{D} = \frac{\alpha}{D} + 1$
- 在样点的编码最大最小值处，以上第一层的修改不具备选择加1或者减1的条件，因此，在第二层嵌入中应该标记为湿点



6 文献阅读推荐

- [1] 教材第6章
- [2] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography, Multimedia Systems, 11(2): 98–107, 2005
- [3] J. Fridrich, M. Goljan, Petr Lisonek, and D. Soukal. Writing on wet paper, IEEE Transactions on Signal Processing, 53(10): 3923 - 3935, Oct. 2005
- [4] W. Zhang, X. Zhang, and S. Wang. A double layered “plus-minus one” data embedding scheme, IEEE Signal Processing Letters, 14(11): 848-851, 2007

谢谢！

中国科学院信息工程研究所
INSITUTE OF INFORMATION ENGNEERING CAS

SKLOIS
信息安全国家重点实验室