3 科研经历 —— 具有隐私保护的图像处理

① 项目任务 —— 加密图像的压缩

标准 JPEG 图像压缩

JPEG 压缩算法

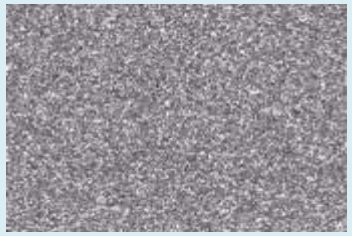
原始图像　　压缩后的图像

具体操作

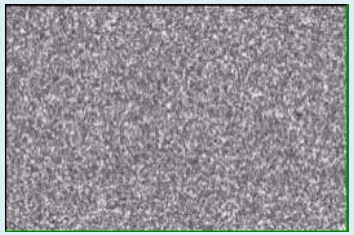
基于**标准 JPEG 压缩算法**，选用加密算法将图片**加密**。加密过后，对密文状态下的图像进行对应的**压缩**操作。最后对压缩的结果进行解密处理，得到对应压缩后的图像。

基于标准 JPEG 图像压缩的加密图像压缩

加密



压缩



解密

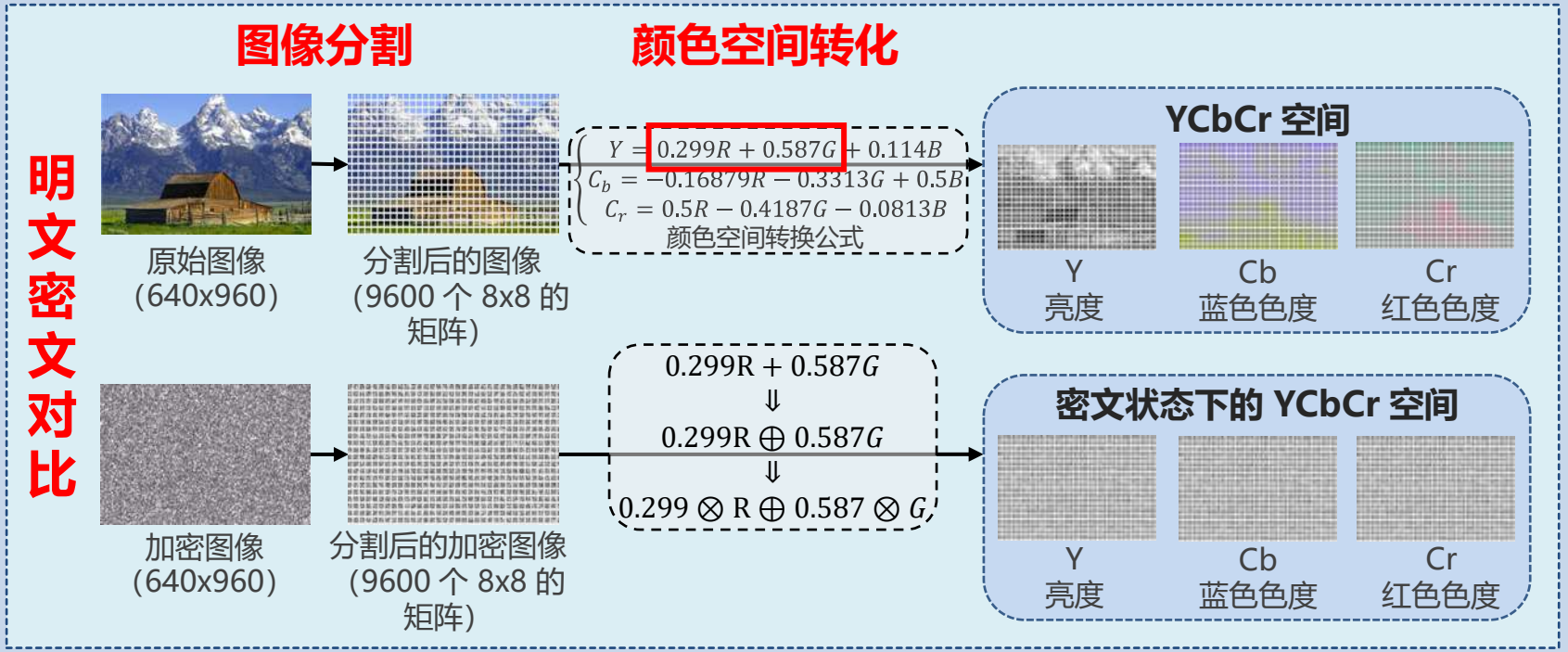
原始图像　　加密图像　　压缩后的加密图像　　压缩后的图像

3 科研经历 —— 具有隐私保护的图像处理

② JPEG压缩算法

① **图像分割**：将原始图像**分割**为 n 个 8x8 的 RGB 图像，后面的操作以**每个 8x8 的图像**为单位。在这一步中密文状态下的操作和明文相同；

② **颜色空间转化**：将每个 **RGB** 图像根据**颜色空间转换公式**转化为三个 **YCbCr** 的矩阵。这一步操作中涉及到**图像信息（m_x）**和**标准系数（k）相乘**，以及**不同数据（m_x）**的**相加**；



3 科研经历 —— 具有隐私保护的图像处理

② JPEG压缩算法

③ **离散余弦变换**：分别对三个小矩阵做**二维离散余弦变换**，将图像信号从时域转换到频域，分离出**高频信息**和**低频信息**。这一步操作中涉及到了**图像信息（m_x）**和**参数（k）**的**相乘**，以及**不同数据（m_x）**之间的**相加**；

④ **量化**：将经过离散余弦变换的**矩阵**和**标准量化表相除**并**取整**，对**高频信息**进行去除。这一步操作中涉及到了**图像信息（m_x）**和**标准量化表数据（k）**之间的**除法**的操作；

离散余弦变换

$$F(u,v)=\alpha(u)\cdot\alpha(v)\cdot\sum_{y=0}^{7}\sum_{x=0}^{7}\left(f(x,y)\cdot\cos\frac{(2x+1)u\pi}{16}\cdot\cos\frac{(2y+1)v\pi}{16}\right)$$

离散余弦变换公式
x, y 是输入像素坐标
u, v 是输出结果坐标

$$\alpha(u)=\begin{cases}\frac{1}{\sqrt{8}}, u=0\\ \frac{1}{2}, u\neq 0\end{cases}$$

量化

$$Q_Y=\begin{Bmatrix}16&11&10&16&24&40&51&61\\12&12&14&19&26&58&60&55\\14&13&16&24&40&57&69&56\\14&17&22&29&51&87&80&62\\18&22&37&56&68&109&103&62\\24&35&55&64&81&104&113&92\\49&64&78&87&103&121&120&101\\72&92&95&98&112&100&103&99\end{Bmatrix}$$

标准**亮度**量化表

密文下的离散余弦变换

$$\alpha(\cdot)\cdot\alpha(\cdot)\cdot f(x_0,y_0)\cdot\cos(\cdot)\cdot\cos(\cdot)+\cdots$$
$$\alpha(\cdot)\cdot\alpha(\cdot)\cdot f(x_0,y_0)\cdot\cos(\cdot)\cdot\cos(\cdot)$$
$$\Downarrow$$
$$k\cdot f(x_0,y_0)+\cdots$$
$$k\cdot f(x_0,y_0)$$
$$\Downarrow$$
$$k\otimes f(x_0,y_0)\oplus\cdots$$
$$k\otimes f(x_0,y_0)$$

$$\begin{Bmatrix}52&55&61&66&70&61&64&73\\63&59&55&90&109&85&69&72\\62&59&68&113&144&104&66&73\\63&58&71&122&106&106&70&69\\67&61&68&104&126&88&68&70\\79&65&60&70&77&68&58&75\\85&71&64&59&55&61&65&83\\87&79&69&68&65&76&76&94\end{Bmatrix}\rightarrow\begin{Bmatrix}-415.38&-30.19&-61.2&27.24&56.12&-20.1&-2.39&0.46\\4.47&-21.86&-60.76&10.25&13.15&-7.09&-8.54&4.88\\-46.83&7.37&77.13&-24.56&-28.91&9.93&5.42&-5.65\\-48.53&12.07&34.1&-47.73&-10.24&6.3&1.83&1.95\\12.12&-6.55&-13.2&-3.95&-1.87&1.75&-2.79&3.14\\-7.73&2.91&2.38&-5.94&-2.38&-0.61&4.3&1.85\\-1.03&0.18&0.42&-2.42&-0.88&-3.02&4.12&-0.66\\-0.17&0.14&-1.07&-4.19&-1.17&-0.1&0.5&1.68\end{Bmatrix}\rightarrow\begin{Bmatrix}-26&-3&-6&2&2&-1&0&0\\0&-2&-4&1&1&0&0&0\\-3&1&5&-1&-1&0&0&0\\-3&1&2&-1&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{Bmatrix}$$

某**亮度** 8x8 矩阵　　经过二维 DCT 后的某**亮度** 8x8 矩阵　　量化过后的某**亮度** 8x8 矩阵

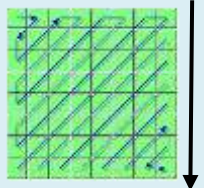
3 科研经历 —— 具有隐私保护的图像处理

② JPEG压缩算法

⑤ **哈夫曼编码**：首先将**二维矩阵**通过 **Zig-Zag 排列**生成**一维数组**。然后对一维数列进行 **RLE 编码**（把数据中非 0 的数据，以及数据前 0 的个数作为一个处理单元）。再根据数据中元素的使用频率，参照 JPEG 官方提供的**哈夫曼表**，调整元素的编码长度，以得到更高的压缩比。最后把二进制数据进行**序列化**，并写入 JPEG 文件。 主要的压缩步骤在 RLE 编码中，所以只需要在密文状态下实现**原始数据（m_x）和 0 之间的比较操作**，剩余对数据和码表之间进行映射的操作可直接在明文下进行。

$$\begin{Bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{Bmatrix}$$

解密后的某**亮度** 8x8 矩阵



Zig-Zag 排列

$-26, 0, -3, -6, -2, -3, \cdots, 0, 0, 0$

排列后得到的一维数列

哈夫曼编码

① 原始数据	$-26, 0, -3, -6, -2, -3, \cdots, 0, 0, 0$							
② RLE编码	-26		$0, -3$		-6		$\cdots$	$0,0, \ldots, 0$
	$(0, -26)$		$(1, -3)$		$(0, -6)$		$\cdots$	EOB
③ BIT编码	(0,5,00101)		(1,2,00)		(0,3,001)		$\cdots$	EOB
	$(0x05,00101)$		$(0x12,00)$		$(0x03,001)$		$\cdots$	EOB
④ 哈夫曼编码	001	00101	11011	00	100	001	$\cdots$	1010
⑤ 序列化	001 00101 11011 00 100 001 $\cdots$ 1010							
	25 $D9$ $\cdots$							

3 科研经历 —— 具有隐私保护的图像处理

③ 同态加密的应用

如何实现在密文状态下进行压缩操作又不影响最后的压缩结果?

普通加密只能保护数据的隐私，但在对加密的数据进行操作后不能得出正确的结果；**同态加密**可以在加密的数据中进行诸如检索、比较等操作，得出正确的结果，而在整个处理过程中无需对数据进行解密。**同态加密**主要分为：

① **全同态加密（慢）**：支持密文域任意次数的**加法和乘法**同态运算；

② **部分同态加密（快）**：支持密文域任意次数的**加法或乘法**同态运算。

乘法同态：$E(m_1) \otimes E(m_2) = E(m_1 \cdot m_2)$

加法同态：$E(m_1) \oplus E(m_2) = E(m_1 + m_2)$

以 **RSA** 加密算法为例：

乘法同态：$E(\boldsymbol{m_1} \cdot \boldsymbol{m_2}) = (m_1 \cdot m_2)^e = {m_1}^e \cdot {m_2}^e = E(m_1) \cdot E(m_2)$

以 **Paillier** 加密算法为例：

加法同态：$E(\boldsymbol{m_1} + \boldsymbol{m_2}) = g^{m_1+m_2} \cdot r^n = g^{m_1} \cdot r^n \cdot g^{m_2} \cdot r^n = E(m_1) \cdot E(m_2)$

数乘同态：$E(\boldsymbol{k} \cdot \boldsymbol{m_1}) = g^{(k \cdot m_1)} \cdot r^n = g^{k \cdot m_1} \cdot r^n = E(m_1)^k$

3 科研经历 —— 具有隐私保护的图像处理

③ 同态加密的应用

密文状态下的除法操作如何实现?

① 将**除法**对应的公式做一定的变换，通过**加法同态**和**数乘同态**，利用**随机数** r_1、r_2构造公式，使得公式的值不变且**分母**和**分子**都无法被解出。解密后，直接进行明文下的除法运算；

② 在除法运算后，得到的**整数部分**为**原始除法得到的整数部分加上随机数** r_2；**小数部分**即为**原始除法得到的小数部分**，小数部分可以直接进行四舍五入。分别将计算得到的**整数部分**和**小数部分**加密后再进行计算；

③ 最后通过**加法同态**，减去**随机数** r_2，计算得到**商对应的密文**。

密文状态下的比较操作如何实现?

① 通过**数乘同态**将**随机数** r_x 和**数据** m_x 相乘，进行解密后将 $r_x m_x$ **与 0 比较大小**；

② 根据判断的结果进行 **RLE 编码**，如果是 0，则给当前处理单元中 0 的个数加 1；如果不是 0，则把前面 0 的个数和当前数据记为一个处理单元。

密文下的除法操作

$$E(a)\ E(b) \longrightarrow E(br_1)\ \ E(ar_1 + br_1r_2)$$

$$\frac{a}{b} = \frac{\Delta}{\Delta} + \Delta = \frac{ar_1 + br_1r_2}{br_1} - r_2$$

$$\frac{ar_1 + br_1r_2}{br_1} = \frac{a}{b} + r_2 = Int' + Float$$

$$= (Int + r_2) + Float$$

$$E(Result) = E(Int' - r_2 + Float)$$

$$= E(Int') \oplus E(-r_2) \oplus E(Float)$$

$$E(Float) = \begin{cases} E(0), 0 \le Float < 0.5 \\ E(1), 0.5 \le Float < 1 \end{cases}$$

密文下的 RLE 编码

$$E(x_0), E(x_1), E(x_2), \cdots, E(x_{63})$$

$$\Downarrow$$

$$E(x_0 r_0) = E(x_0) \otimes E(r_0),$$

$$\cdots$$

$$E(x_{63} r_{63}) = E(x_{63}) \otimes E(r_{63})$$

$$\Downarrow$$

$$x_0 r_0 \neq 0 \ or \ x_0 r_0 = 0$$

$$\Downarrow$$

$$(0, E(x_0)), (5, E(x_2)), \cdots, EOB$$