A JPEG-tolerant Image Authentication Scheme

[1] Takeyuki Uehara, [1] Rei Safavi-Naini, [2] Philip Ogunbona

[1] Centre for Information Security, University of Wollongong

[2] Motorola Australian Research Centre

AUSTRALIA

To be published *ACM Journal of Multimedia Communication*

Image authentication

- Cryptographic authentication detects single bit change.
- Image authentication need not provide bit accuracy:
  - *redundancy* and *irrelevancy*
- Image files are mainly in compressed form.
  - Lossy compression systems : JPEG, JPEG2000, MPEG
- Changes that are caused by compression are acceptable:
  - within a predefined compression rate

High compression rate	$\rightarrow$	large changes (data loss)
Low compression rate	$\rightarrow$	small changes (small loss)

A JPEG tolerant authentication system

- JPEG compression :
  1. Transform :
     - divide the image into 8×8 pixel blocks
     - use two dimensional DCT transform to generate 64 coefficients (64 frequencies)
  2. Quantization : Remove irrelevant information.
  3. Entropy coding : *Remove redundancy*: run-length coding and Huffman/arithmetic coding.

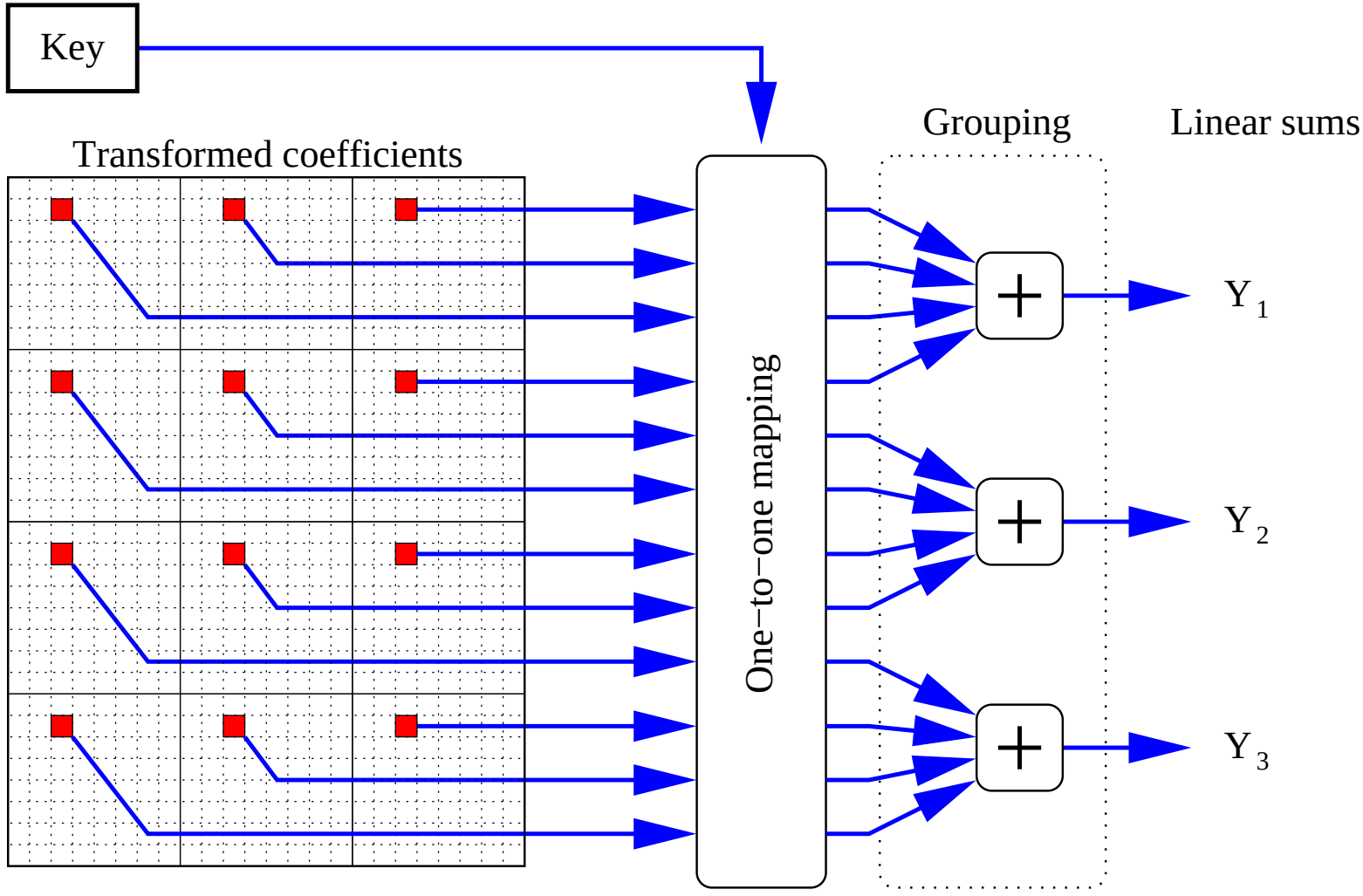
Calculating the authenticator

Authentication :

- Divide the image into 8×8 pixel blocks.
- Transform each block using the DCT.
- Partition the blocks into g group, $\mathcal{G}_{x_1}, \mathcal{G}_{x_2}, ...$
- Select a subset of frequencies. For each selected frequency u, and for each group j form a *feature code* $Y_j^{(u)}$.
- The authenticator is the sequence of feature codes, encoded in binary, $Y_1^{(1)} Y_1^{(2)} Y_1^{(3)} .. Y_2^{(1)} Y_2^{(2)} ..$

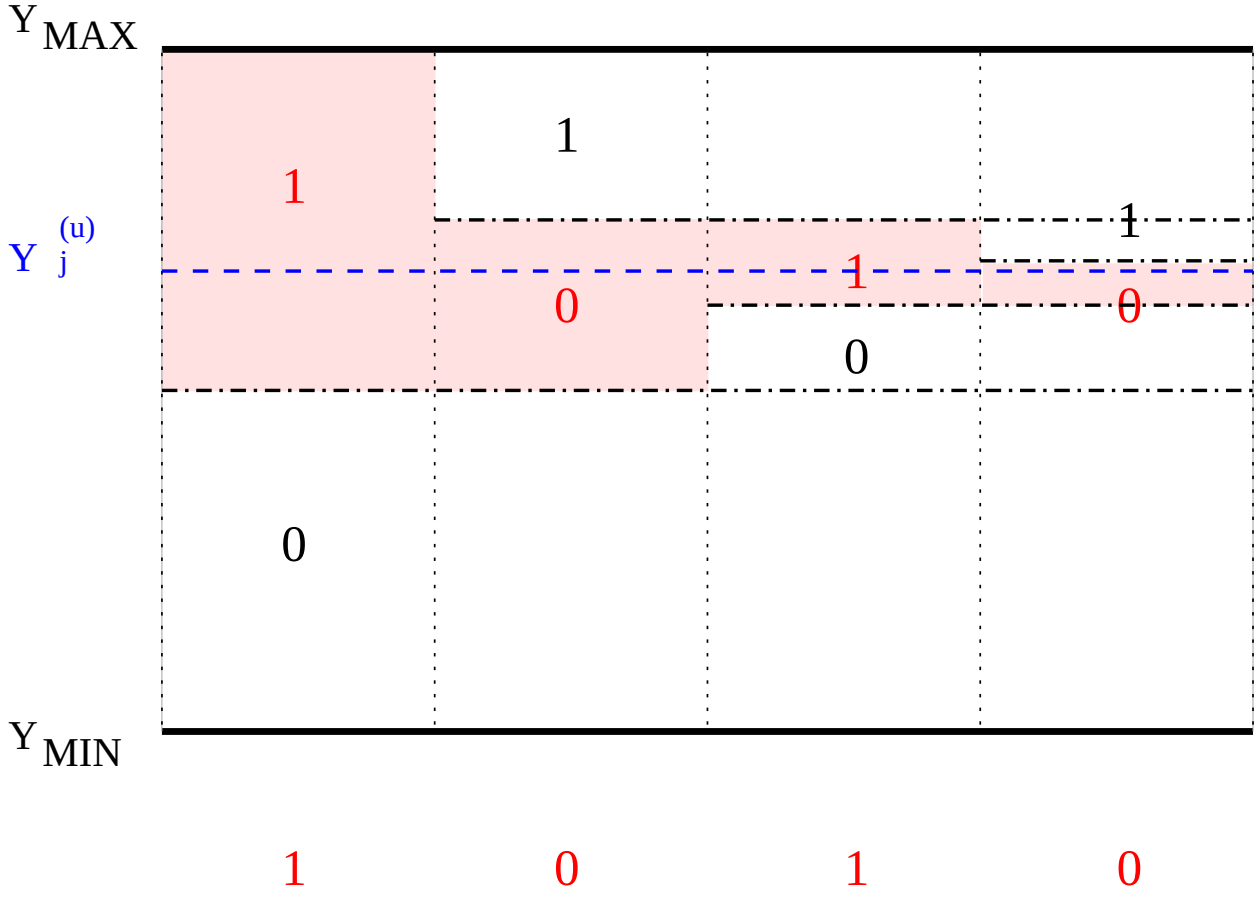
Calculating a feature code

- A feature code is a linear sum of the coeffcients.



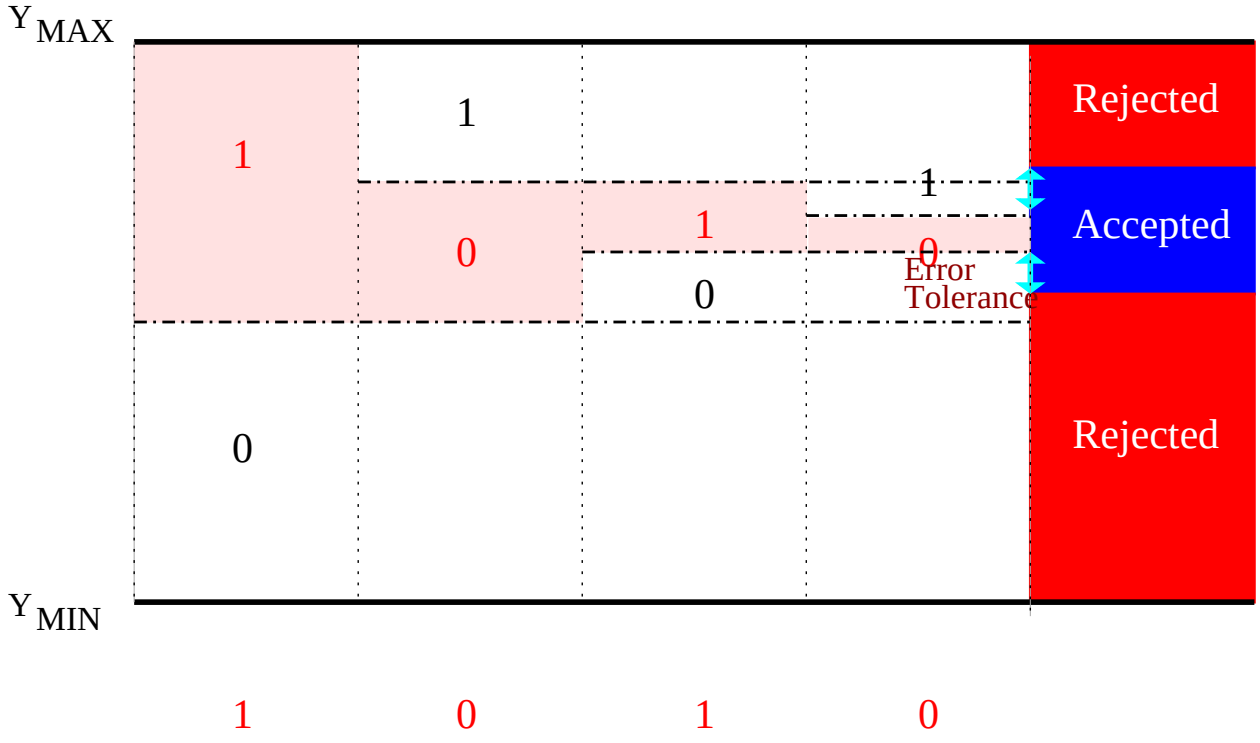
Encoding a feature code

- Use successive halving of the interval.
- The length depends on the required compression tolerance.



Verification

- Compute $\tilde{Y}_j^{(u)}$ from the reconstructed coefficients.
- Decode the feature codes.
- The two part must be *almost* equal.



Performance