

DOUBLE EMBEDDING IN THE QUANTIZATION INDEX MODULATION FRAMEWORK

A. Sarkar and B. S. Manjunath

Department of Electrical and Computer Engineering,
University of California at Santa Barbara
Santa Barbara, CA 93106

ABSTRACT

Quantization index modulation (QIM) is a commonly used data hiding technique where a single bit is embedded per coefficient. Here, we propose the use of double embedding in the QIM framework where a single coefficient is modified twice, using two quantizers, to embed two bits. The motivation behind substituting single embedding with double embedding in the QIM framework for a certain steganographic scheme is to increase its hiding rate without significantly increasing the embedding distortion and the stego scheme's detectability against steganalysis. We empirically determine the best way to couple the double embedding framework with a repeat-accumulate code based error correction scheme. For moderate noise levels, the use of double embedding is seen to be significantly advantageous over single embedding.

Index Terms— double embedding, quantization index modulation, repeat-accumulate code, embedding distortion

1. INTRODUCTION

A common embedding technique used for data hiding is quantization index modulation (QIM) [1, 2]. In QIM, data embedding is performed through a choice of the quantizer and an element is quantized once to embed a certain bit. Here, we look at incorporating “double embedding” in QIM-based steganographic schemes, where the same element is modified twice to embed 2 bits. The aim is to increase the hiding rate without compromising on the embedding distortion introduced as well as the robustness of the steganographic scheme against the same steganalysis methods. As a secure hiding framework in which to compare the detectability of single and double embedding, we have used our recently proposed secure steganographic scheme, Yet Another Steganographic Scheme (YASS) [3], which achieves security based on hiding in randomized blocks. In YASS, hiding occurs in a selected band of quantized discrete cosine transform (DCT) elements computed per 8×8 block, chosen randomly out of a $B \times B$ ($B > 8$) big-block. To compensate for the errors introduced by the JPEG compression step in YASS, it is coupled with a repeat-accumulate (RA) [4] coding framework.

This research is supported by a grant from ONR # N00014-05-1-0816.

1.1. Single and Double Embedding Using QIM

We compare the single and double embedding schemes, within the QIM framework, using Fig. 1.

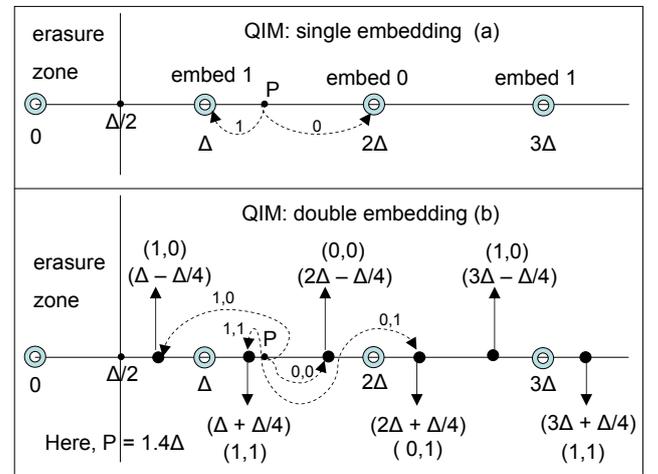


Fig. 1. QIM-based hiding is shown for single and double embedding schemes.

For the single embedding case (Fig. 1(a)), let the QIM embedding logic be converting an element to the nearest even/odd multiple of the quantization interval, Δ , to embed 0/1, respectively. For hiding, we use quantized discrete cosine transform (DCT) coefficients. For perceptual transparency, we do not modify coefficients that are too close to zero; hence, all coefficients in the range $[-0.5, 0.5]$ are mapped to zero and are regarded as erasures.

The two quantizers used for double embedding (Fig. 1(b)) have quantization intervals of Δ and $\Delta/2$, respectively. In the example (Fig. 1(b)), $\Delta = 1$ and the DCT coefficient (P) equals 1.4. Let the first bit to be embedded be 1 (using the coarser quantizer) and the second bit be 0 (using the finer quantizer). To embed 1, the coefficient (1.4) is changed to the nearest odd multiple of Δ (1). For the second bit, the coefficient is decreased/increased by $\Delta/4$ to embed 0/1 respectively. To embed 0, the coefficient is changed from 1 to 0.75. The double embedding scheme can also be thought of as *modifying the coefficient once to one of 4 nearest codewords to embed 2 bits*. E.g. a coefficient of 1.4 can be changed to

0.75/1.25/1.75/2.25 to embed 10/11/00/01, respectively.

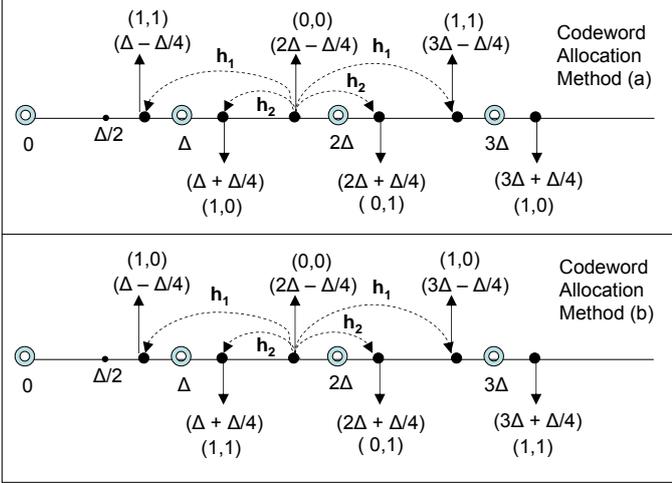


Fig. 2. Double embedding is shown for two codeword allocation schemes: in (a) (or (b)), the quantization level (1.25Δ) (or (0.75Δ)) corresponds to 1 and 0 being embedded for the coarser and finer quantizers, respectively. Here, h_1 (or h_2) is the probability of mistaking a coefficient, originally mapped to 1.75Δ after embedding, as being mapped to 2.75Δ (or 2.25Δ) due to JPEG compression induced noise.

2. DISTORTION AND ERROR ANALYSIS FOR THE QIM EMBEDDING SCHEMES

Better Allocation of Codewords for Double Embedding

In Fig. 2(a) and (b), two different methods of allocation of quantizer levels are shown for double embedding. For the former, there is always a change of 1 bit between successive quantization levels. For the latter, a maximum change of 2 bits is involved for successive quantization levels. Let $Q(x)$ denote the quantized value of the DCT coefficient x , while \hat{x} is the nearest quantizer level to which $Q(x)$ gets mapped due to JPEG compression noise. The probability terms h_1 and h_2 (shown in Fig. 2) are expressed as:

$$\begin{aligned} h_1 &= Pr(\hat{x} = 2.75\Delta | Q(x) = 1.75\Delta) \\ &= Pr(n > 0.75\Delta, n < 1.25\Delta) \\ h_2 &= Pr(\hat{x} = 2.25\Delta | Q(x) = 1.75\Delta) \\ &= Pr(n > 0.25\Delta, n < 0.75\Delta) \end{aligned}$$

where $n = \hat{x} - Q(x)$ is the JPEG compression induced noise.

Based on the level allocation in Fig. 2(a) and (b), the probability of error $p_{e,a}$ and $p_{e,b}$ are computed, assuming that the original coefficient, after QIM-based embedding, $Q(x) = 1.75\Delta$. When $Q(x)$ gets mapped to 0.75Δ (or 1.25Δ) in method (a), the number of bit errors introduced as ‘00’ is wrongly decoded as ‘11’ (or ‘10’) is 2 (or 1).

$$\begin{aligned} p_{e,a} &= 2Pr(\hat{x} = 0.75\Delta) + Pr(\hat{x} = 1.25\Delta) + \\ &Pr(\hat{x} = 2.25\Delta) + 2Pr(\hat{x} = 2.75\Delta) \\ &= 2h_1 + h_2 + h_2 + 2h_1 = 4h_1 + 2h_2 \end{aligned}$$

$$\text{Similarly } p_{e,b} = h_1 + 2h_2 + h_2 + h_1 = 2h_1 + 3h_2$$

All the probability terms are conditioned on $Q(x) = 1.75\Delta$.

Here, we consider the 4 quantization levels nearest to $Q(x)$ as possible quantization levels where $Q(x)$ can get mapped to due to noise. The noise distribution is symmetrical and falls off sharply on either side of 0 - this assumption has been experimentally verified. Therefore, the probability of $Q(x)$ getting mapped to levels that are more than Δ away from $Q(x)$ ($Pr(|n| > \Delta)$) will be very small. Due to the highly tapered distribution of the noise, $h_1 \ll h_2$, and hence, $p_{e,a} < p_{e,b}$. Due to the lower error probability, the quantizer level allocation in Fig. 2(a) is preferred.

Embedding Distortion for Double Embedding

For double embedding, the embedding and decoding can occur on a ‘bit-by-bit’ basis, or on a ‘2-bit tuple’ basis. E.g. let us embed ‘00’ using an image coefficient of 2.9 ($\Delta = 1$). In the ‘bit-by-bit’ approach using codeword allocation (a), we embed the first ‘0’ by converting it to the nearest even integer, 2. Then we embed the second ‘0’ by changing the term to $(7/4)$. If ‘00’ was considered as a 2-tuple and we wanted to modify it to the nearest element in $\{(7/4)\Delta \pm 2n\Delta, n \in \mathbb{Z}\}$, that element would have been $(15/4)$. The effective distortion is 1.15 ($2.90 - 1.75$) and 0.85 ($3.75 - 2.90$), using 1-bit and 2-bit tuples, respectively. Embedding the bits as 2-bit tuples lowers the squared distortion, as shown later using (1) and (2).

The distortion for ‘bit-by-bit’ embedding is expressed in (1). Here, p_{01} is the probability that the bit to be embedded using the coarser and finer quantizer equals 0 and 1, respectively. By symmetry, p_{00}, p_{01}, p_{10} and p_{11} all equal 0.25. Though the distribution of the DCT coefficients is not uniform (the distribution of the AC DCT coefficients has been approximated as Laplacian [5]), we assume a uniform distribution $f(x)$ in (1) and (2) to simplify computations. The average squared distortion \mathcal{D}_1 for a non-erasure zone of length 2Δ (the effective distortion is the same over any non-erasure zone of the same length under the uniform distribution assumption) using codeword allocation method (a) is:

$$\mathcal{D}_1 = \int_{2.25\Delta}^{4.25\Delta} \{x - Q(x)\}^2 f(x) dx \quad (1)$$

$$\begin{aligned} &\text{where we assume } f(x) = (1/2\Delta), x \in [2.25\Delta, 4.25\Delta] \\ &= \frac{p_{00}}{2\Delta} \int_{2.25\Delta}^{3\Delta} \{x - 1.75\Delta\}^2 dx + \frac{p_{00}}{2\Delta} \int_{3\Delta}^{4.25\Delta} \{x - 3.75\Delta\}^2 dx \\ &+ \frac{p_{01}}{2\Delta} \int_{2.25\Delta}^{3\Delta} \{x - 2.25\Delta\}^2 dx + \frac{p_{01}}{2\Delta} \int_{3\Delta}^{4.25\Delta} \{x - 4.25\Delta\}^2 dx \\ &+ \frac{p_{11}}{2\Delta} \int_{2.25\Delta}^{4\Delta} \{x - 2.75\Delta\}^2 dx + \frac{p_{11}}{2\Delta} \int_{4\Delta}^{4.25\Delta} \{x - 4.75\Delta\}^2 dx \\ &+ \frac{p_{10}}{2\Delta} \int_{2.25\Delta}^{4\Delta} \{x - 3.25\Delta\}^2 dx + \frac{p_{10}}{2\Delta} \int_{4\Delta}^{4.25\Delta} \{x - 5.25\Delta\}^2 dx \\ &= (19/48)\Delta^2 \end{aligned}$$

The distortion for the ‘2-bit tuple’ based embedding is expressed in (2). Considering the zone $[2.25\Delta, 4.25\Delta]$ and us-

Table 1. The average bpsc and PSNR values are reported for 500 images, obtained using single embedding and the variants of double embedding, in the QIM-based YASS framework. We use $B = 9$, λ (size of hiding band) = 5 and $QF_a = 75$.

| Single Embedding | | | | Method 1 | | Method 2 | | Method 3 | | Method 4 | |
|------------------|----------|---------------|-------|----------|-------|----------|-------|----------|-------|---------------|-------|
| QF_h | Δ | bpsc | PSNR | bpsc | PSNR | bpsc | PSNR | bpsc | PSNR | bpsc | PSNR |
| 50 | 1 | 0.0998 | 45.45 | 0.0928 | 45.16 | 0.1256 | 44.94 | 0.1236 | 44.94 | 0.1298 | 45.22 |
| 60 | 1 | 0.0984 | 47.08 | 0.0661 | 46.75 | 0.0899 | 46.55 | 0.0942 | 46.55 | 0.0987 | 46.85 |
| 70 | 1 | 0.0743 | 48.71 | 0.0382 | 48.38 | 0.0415 | 48.17 | 0.0368 | 48.17 | 0.0425 | 48.49 |
| 75 | 1 | 0.0475 | 49.61 | 0.0230 | 49.29 | 0.0225 | 49.07 | 0.0146 | 49.07 | 0.0219 | 49.41 |
| 50 | 1.5 | 0.0967 | 42.67 | 0.1501 | 41.99 | 0.1551 | 42.14 | 0.1431 | 42.14 | 0.1626 | 42.37 |
| 60 | 1.5 | 0.1094 | 44.23 | 0.1208 | 43.55 | 0.1423 | 43.68 | 0.1344 | 43.68 | 0.1586 | 43.93 |
| 70 | 1.5 | 0.1171 | 45.83 | 0.0797 | 45.18 | 0.1060 | 45.28 | 0.1043 | 45.28 | 0.1327 | 45.56 |
| 75 | 1.5 | 0.1116 | 46.74 | 0.0598 | 46.11 | 0.0808 | 46.18 | 0.0802 | 46.18 | 0.1131 | 46.47 |
| 50 | 2 | 0.0926 | 40.47 | 0.1577 | 39.48 | 0.1593 | 39.94 | 0.1451 | 39.94 | 0.1698 | 40.13 |
| 60 | 2 | 0.1071 | 41.98 | 0.1531 | 41.02 | 0.1571 | 41.45 | 0.1417 | 41.45 | 0.1758 | 41.66 |
| 70 | 2 | 0.1206 | 43.58 | 0.1306 | 42.66 | 0.1344 | 43.03 | 0.1224 | 43.03 | 0.1644 | 43.27 |
| 75 | 2 | 0.1265 | 44.50 | 0.1078 | 43.60 | 0.1135 | 43.94 | 0.1046 | 43.94 | 0.1509 | 44.19 |

ing codeword allocation method (a), the average squared distortion \mathcal{D}_2 is computed as:

$$\begin{aligned} \mathcal{D}_2 &= \frac{p_{10}}{2\Delta} \int_{2.25\Delta}^{4.25\Delta} \{x - 3.25\Delta\}^2 dx \\ &+ \frac{p_{00}}{2\Delta} \int_{2.25\Delta}^{2.75\Delta} \{x - 1.75\Delta\}^2 dx + \frac{p_{00}}{2\Delta} \int_{2.75\Delta}^{4.25\Delta} \{x - 3.75\Delta\}^2 dx \\ &+ \frac{p_{01}}{2\Delta} \int_{2.25\Delta}^{3.25\Delta} \{x - 2.25\Delta\}^2 dx + \frac{p_{01}}{2\Delta} \int_{3.25\Delta}^{4.25\Delta} \{x - 4.25\Delta\}^2 dx \\ &+ \frac{p_{11}}{2\Delta} \int_{2.25\Delta}^{3.75\Delta} \{x - 2.75\Delta\}^2 dx + \frac{p_{11}}{2\Delta} \int_{3.75\Delta}^{4.25\Delta} \{x - 4.75\Delta\}^2 dx \\ &= \Delta^2/3 = (16/48)\Delta^2, \end{aligned} \quad (2)$$

which equals the average squared distortion over a non-erasure zone of length 2Δ for the single embedding case.

It can be further shown that the average distortion over the erasure region is $(40\Delta^2/192)$ and $(43\Delta^2/192)$ for single and double embedding cases, respectively. There is a high distribution of DCT coefficients in the erasure zone. Hence, for the same Δ and similar hiding conditions, the embedding distortion is slightly higher when double embedding is used.

3. USING QIM-BASED DOUBLE EMBEDDING WITH RA CODING FRAMEWORK

Assume that there are N embeddable coefficients. For the single embedding case, the RA-encoded sequence has N bits and assuming that the minimum redundancy factor which allows perfect data recovery is q_{opt} , the number of databits equals $\lfloor N/q_{opt} \rfloor$. For the double embedding case, the RA-encoded sequence accommodates $2N$ bits.

From Fig. 1(a) and (b), it is seen that the nearest quantization levels are Δ and $\Delta/2$ apart, for single and double embedding, respectively. Hence, though a coefficient can embed 2 bits, the noise robustness will always be lower for double

embedding. Let q_{opt} equal q_1 and q_2 for single and double embedding, respectively, where $q_2 \geq q_1$. Hence, the effective number of databits equals $\lfloor N/q_1 \rfloor$ and $\lfloor 2N/q_2 \rfloor$, respectively. Double embedding achieves a higher data-rate than single embedding if $\lfloor 2N/q_2 \rfloor > \lfloor N/q_1 \rfloor$.

The following variants of double embedding have been experimented with:

- **Method 1 (M1):** It embeds an RA-encoded sequence of N bits using each of the two quantizers. Let the optimum q factor for the two quantizers be q_a and q_b , respectively. The total number of databits equals $(\lfloor N/q_a \rfloor + \lfloor N/q_b \rfloor)$.

- **Method 2 (M2):** Let $\{b_1, b_2, \dots, b_{2N}\}$ be the sequence of $2N$ bits. The first half ($\{b_1, \dots, b_N\}$) is embedded using the coarser quantizer and $\{b_{N+1}, \dots, b_{2N}\}$ is embedded using the finer quantizer.

- **Method 3 (M3):** It embeds every odd-numbered bit ($\{b_1, b_3, \dots, b_{2N-1}\}$) using the coarser quantizer and every even-numbered bit using the finer quantizer.

For hiding, we use a big-block size B of 9 in the YASS framework and the hiding band consist of the top 5 ($\lambda = 5$) AC DCT coefficients per 8×8 sub-block. The average hiding rate is expressed in terms of the ‘‘bpsc’’ (bits per non-zero DCT coefficient) and the embedding distortion is quantified in terms of the Peak Signal To Noise Ratio (PSNR) - the results are averaged over 500 images. The design quality factor for hiding, QF_h is varied from 50-75. After hiding, the images are JPEG compressed at $QF_a = 75$. In Table 1, ‘‘bit-by-bit’’ embedding is used for M1-M3 and out of them, *M2 is seen to have the highest embedding rate at the same PSNR*. We then replace the single bit based embedding in M2 by ‘‘2-bit tuple’’ based embedding and refer to it as **Method 4 (M4)** in Table 1.

3.1. Steganalysis Experiments

We conduct the steganalysis experiments on 4500 JPEG images, from the MM270K database ¹, compressed at a quality

¹downloaded from <http://www-2.cs.cmu.edu/yke/retrieval>

Table 2. Comparison of detection accuracies using PF-274 and Chen-324, for single (denoted by (1)) and double (denoted by (2)) embedding in QIM-based YASS, with $B = 9$.

| QF_h | Δ | λ | $P_{d,PF}(2)$ | $P_{d,PF}(1)$ | $P_{d,Ch}(2)$ | $P_{d,Ch}(1)$ |
|--------|----------|-----------|---------------|---------------|---------------|---------------|
| 50 | 1 | 5 | 0.6212 | 0.6062 | 0.5555 | 0.5554 |
| 60 | 1 | 5 | 0.5582 | 0.5541 | 0.5414 | 0.5356 |
| 70 | 1 | 5 | 0.5447 | 0.5238 | 0.5284 | 0.5252 |
| 50 | 2 | 5 | 0.6514 | 0.6464 | 0.6216 | 0.6165 |
| 60 | 2 | 5 | 0.6405 | 0.6382 | 0.5788 | 0.5746 |
| 70 | 2 | 5 | 0.6377 | 0.6311 | 0.5736 | 0.5652 |

factor of 75. Half of the images are used for training and the other half for testing. We use a support vector machine (SVM) based classifier for steganalysis, where the SVM is trained using the following features.

- Pevny and Fridrich’s 274-dimensional feature (**PF-274**) that merges Markov and DCT features [6] is used.
- Chen et al’s 486-dimensional feature (**Chen-486**) [7], that accounts for both intra and inter-block correlation among JPEG DCT coefficients, has also been experimented with.

The probability of classifying a test image correctly as cover or stego - the detection accuracy P_d ($P_d \approx 0.5$ implies undetectable hiding, and as the detectability improves, P_d increases towards 1) is obtained for the different steganalysis methods. We use M4 for double embedding. In Table 2, $P_{d,PF}$ and $P_{d,Ch}$ refer to the detection accuracy obtained using PF-274 and Chen-486, respectively. $P_{d,PF}(1)$ and $P_{d,PF}(2)$ refer to the P_d values for PF-274 based steganalysis, when single and double embedding are used, respectively. *The steganalysis results show that for similar hiding conditions, the detection accuracy (P_d) for double embedding is slightly more than when single embedding is used.* Thus, though double embedding results in a higher hiding rate, the effect on steganalysis performance is quite small.

3.2. Discussion

Based on the embedding distortion (quantified by PSNR) and the steganalysis (quantified by P_d using PF-274 and Chen-486 features) results, the following observations can be made.

- We obtain a much higher embedding rate using double embedding as compared to single embedding, the margin being higher when the system is more robust to JPEG compression (higher Δ and lower QF_h). Single embedding has higher bpnc for the less robust cases ($\Delta = 1$ and $QF_h = 70$ or 75).
- The “2-bit tuple” based approach (M4) generally results in higher bpnc than the “bit-by-bit” based method (M2) - the PSNR is also slightly higher for M4.
- Regarding double embedding, we see that *it is generally better to combine the two N -bit sequences (M2) instead of using them separately (M1)*. Also, for a sequence of $2N$ bits, we find that placing the two sequences consecutively (N bits of the first followed by N bits of the second) is generally better than embedding successive bits using the coarser and finer

quantizers, i.e. M2 is better than M3.

We can explain the above observation as follows - RA-coding generally performs better on longer sequences. Hence, combining the two N -bit RA-coded sequences into one sequence of $2N$ bits gives improved results (M2 over M1). Since there are more errors using the finer quantizer, the decoding errors tend to be more concentrated in M2 and more spread out in M3. *For RA-encoded sequences, we have experimentally observed that when the errors are more concentrated, it generally converges at a lower q as compared to when the errors are well spaced - specially for $q \geq 15$.*

Double embedding can replace single embedding in a steganographic scheme where we need significant increase in bpnc, the penalty being a slight increase in P_d .

4. CONCLUSION

We have increased the hiding rate by using double embedding in the QIM framework. It has been shown that using a single encoded sequence encompassing the two quantizers, where the two successive halves of the sequence are embedded using the coarser and finer quantizers, generally results in a higher embedding rate as compared to single embedding, for moderately noisy channels. Embedding two bits as a pair significantly increases the hiding rate as compared to individual embedding. We will focus on improving the noise robustness of the double embedding scheme in future.

References

- [1] O. Altun, G. Sharma, and M. Bocko, “Set theoretic quantization index modulation watermarking,” in *Proc. of ICASSP*, 2006, vol. 2, pp. 229–232.
- [2] B. Chen and G. W. Wornell, “Quantization Index Modulation: A class of provably good methods for digital watermarking and information embedding,” *IEEE Trans. on Info. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.
- [3] K. Solanki, A. Sarkar, and B. S. Manjunath, “YASS: Yet Another Steganographic Scheme that resists blind steganalysis,” in *9th International Workshop on Information Hiding*, Jun 2007, pp. 16–31.
- [4] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for turbo-like codes,” in *36th Allerton Conf. on Communications, Control, and Computing*, Sept. 1998, pp. 201–210.
- [5] R. Reininger and J. Gibson, “Distributions of the two-dimensional DCT coefficients for images,” *Comm., IEEE Trans. on*, vol. 31, no. 6, pp. 835–839, Jun 1983.
- [6] T. Pevny and J. Fridrich, “Merging Markov and DCT features for multi-class JPEG steganalysis,” in *Proc. of SPIE*, San Jose, CA, 2007, pp. 3 1 – 3 14.
- [7] C. Chen and Y. Q. Shi, “JPEG image steganalysis utilizing both intrablock and interblock correlations,” in *Proc. of International Symposium on Circuits and Systems (IS-CAS)*, May 2008, pp. 3029–3032.