Journal of Imaging Science and Technology® 55(5): 050601-1–050601-9, 2011.

Impact of Scrambling on Barcode Entropy

Marie Vans and Steven J. Simske▲
Hewlett-Packard Laboratories, 3404 East Harmony Road, Fort Collins, Colorado 80528
E-mail: Marie.Vans@hp.com

Abstract. *Security barcodes and other barcodes linked to on-line databases have become commonplace due to the increased availability of mobile phones equipped with high-quality cameras. In this article, the authors provide methods for quantifying the entropy of the embedded barcode data, assuming methods other than the standards-specified error-correcting code (ECC) approaches can be adopted. Higher entropy, which reduces the likelihood of a fraudulent agent being able to "guess" correct barcodes, is measured directly using a variety of novel algorithms and applied to large sets of barcodes. The authors data, however, show that removing ECC provides the additional advantage of increasing the entropy. Thus, all other settings (data payload size, printing technology, substrate used, etc.) being equal, eliminating the ECC increases the security of the information content for the barcodes (DataMatrix) tested. © 2011 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2011.55.5.050601]

INTRODUCTION

The use of barcodes has gone beyond their original purpose for point of sale (ringing up sales). Many organizations (ScanBuy, Microsoft, etc.), standards bodies [open mobile alliance (OMA), GS1 Mobile Commerce, etc.], and consortia have specified the data models for barcodes to be captured at point of sale, as a means to connect to a website, or for consumer capture of salient content about products and/or their surroundings.[1–4]

Two-dimensional (2D) and three-dimensional (3D) color barcodes are increasingly being used to augment traditional one-dimensional (1D) barcodes [e.g., Code 39 and Universal Product Code (UPC)]. Two-dimensional and three-dimensional barcodes are high density and can be used for carrying additional data (e.g., mass serialization) or for referencing (e.g., URL pointing) purposes. Barcodes are normally encoded with error-correcting code (ECC), which is added to allow for recovery from certain types of distortion and damage. While there are many different types of error-correcting codes, the objective for using them is to detect and correct errors introduced during transmission of data over some communications channel. For barcode recognition, however, ECC is used mainly to reconstruct data that may have been physically damaged due to handling in the supply chain. The major disadvantage of using these methods is that the number of errors detectable is limited. Additionally, the ECC added is derived from assumptions, such as Shannon entropy theory,[5] which is more relevant to one-dimensional barcodes or general information theory. Therefore, the use of ECC itself can be questioned, which means that using barcodes as information carriers outside of the current barcode standards may be possible.

Barcodes are basically information-carrying images that are readable by barcode scanning devices. There are a variety of different types of barcodes which are typically designed to meet the needs of a specific industry or application. For example, older, one-dimensional UPC barcodes were designed for tracking retail items in stores (see Figure 1(a)). A one-dimensional barcode has a single dimension (vertical bars) in which to carry information, while two-dimensional barcodes were designed to carry much more information in both horizontal and vertical directions (modules) (see Figs. 1(a)–1(c)). The QRCode is probably the most recognizable of all the 2D barcodes due to its popularity in the consumer space; however, the DataMatrix[6] barcode is increasingly being used by industry to directly mark manufactured items and on documents and packaging. This is due to the fact that DataMatrix barcodes can be printed at and read from a very small area and carry up to 2335 alphanumeric characters.

Our experience has shown that the most critical barcode distortions to address are: (a) Effect of the print-scan cycle, or "copying" cycle, (b) localized damage such as water damage and/or puncturing, and (c) blurring.[7] We have shown in the previous work that it is advantageous to either (a) increase the size of the barcode modules themselves or (b) duplicate the barcode data itself within the barcode, in place of ECC.[7,8] The latter typically results in a barcode "unreadable" to the current standard for the barcode symbology, and allows custom interpretation of familiar barcode representations.

In this article, we describe research meant to highlight the differential effects of scrambling methods on entropy. In information theory, entropy is defined to be a measure of the amount of information contained in a message or the expected value of the information content of a message.[5] In our work, entropy is used as a measure for determining the randomness of encoded barcode data. We apply encryption methods to randomly generated strings, both with and without ECC. This allows us to measure whether

▲IS&T member.

Received Dec. 13, 2010; accepted for publication Apr. 17, 2011; published online Nov. 2, 2011
1062-3701/2011/55(5)/050601/9/$20.00.

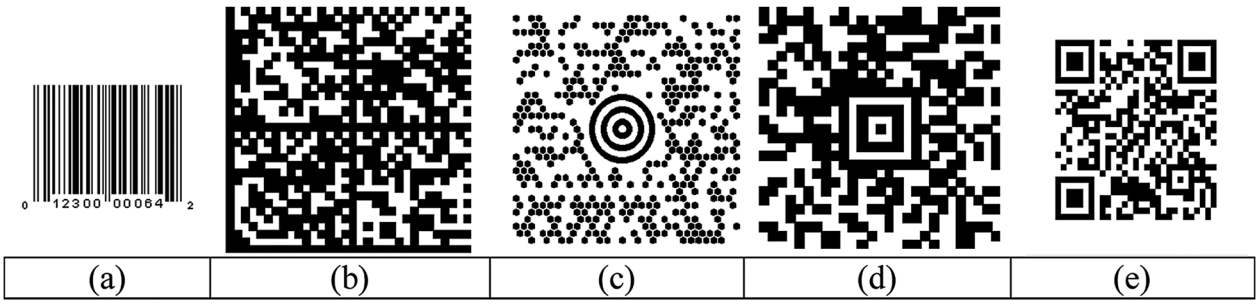

Figure 1. Examples of barcodes: (a) UPC, 1-D, (b) DataMatrix, 2-D, (c) Maxicode, 2-D, (d) Aztec, 2-D, and (e) QR Code, 2-D.

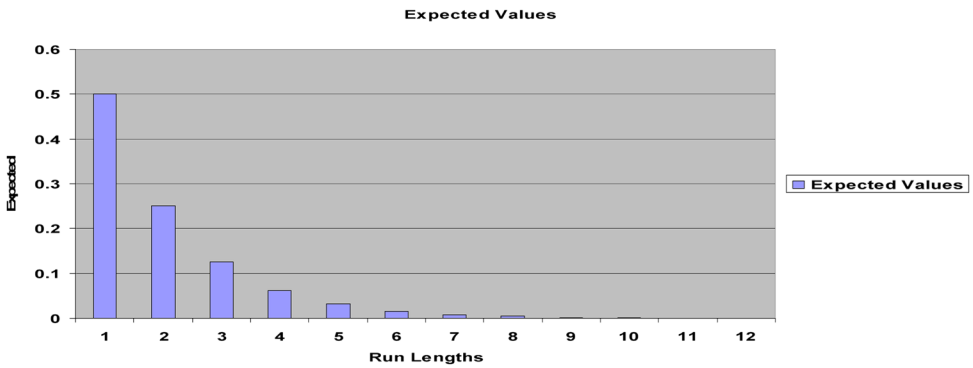

Figure 2. Expected bin percent for maximum entropy.

data containing ECC is more or less random than completely random data, as well as the effect of encryption on both types of data. Increased entropy, which reduces the likelihood of a fraudulent agent being able to "guess" correct barcodes, is measured directly using a variety of novel entropy algorithms and applied to large sets of barcodes. Standard statistical methods are applied to results in an effort to determine whether it is possible to detect differences between strings containing ECC and those that are purely random.

We discuss the implications of these findings on overall security printing and forensic printing ecosystems. Security is an important consideration, given the loss in counterfeiting of barcodes. For example, a barcode scam in which legitimate barcodes were copied and scanned as normal led to a $\$1.5 \times 10^6$ loss.[9] Since large criminal organizations are responsible for much of the fraud—counterfeiting, factory overruns, diversion, smuggling, rebate fraud, etc.—that currently exceeds 8% of world trade,[10] an effective security ecosystem is designed to decrease the initial time to discover and enable efficient and accurate assessment of the size of the counterfeiters involved. The barcode scrambling approaches outlined, herein, are an important part of that ecosystem of combined security printing, investigation and evidence gathering, and prosecution.

METHODS AND MATERIALS

For our experiments, we used the DataMatrix standard[6] for barcodes containing ECC and created DataMatrix-like barcodes that contained no ECC. We used three entropy measures on a dataset of 672,000 two-dimensional barcodes, half of them incorporating Reed-Solomon ECC,[11] with the other half having no ECC. Reed-Solomon ECC is the error-correcting code used in the DataMatrix standard. DataMatrix was ideal for our purposes because data is packed into the array in a way that makes ECC modules simple to extract. Therefore, we can easily construct a barcode out of a set of data along with the appropriate ECC modules, measure the entropy of the entire set of barcode bits and then create another barcode consisting of the same data-bearing modules but with the ECC modules replaced with an identical number of random bits. We used entropy as a measure for the effect of ECC and scrambling on these two sets of 2D barcodes. Here, entropy represents signal randomness, i.e., how the bits are distributed in a signal.

ENTROPY MEASURES

For example, Eq. (1) presents what we call "normalized entropy," where N is the maximum number of run lengths (substrings of all 0's or all 1's) and $E(X)$ is $M * (1/2)i$, which is the expected number of run lengths of each length i. x is the actual number found in each bin i, and M is the total number of all run lengths.

$$e_1 = \sum_{i=1}^{N} \log\left(\frac{E(X) + |E(X) - x|}{E(X)}\right) * E(X) \qquad (1)$$

Figure 2 shows a graph of the expected bin percentages for maximum entropy of a signal. For example, if we have a

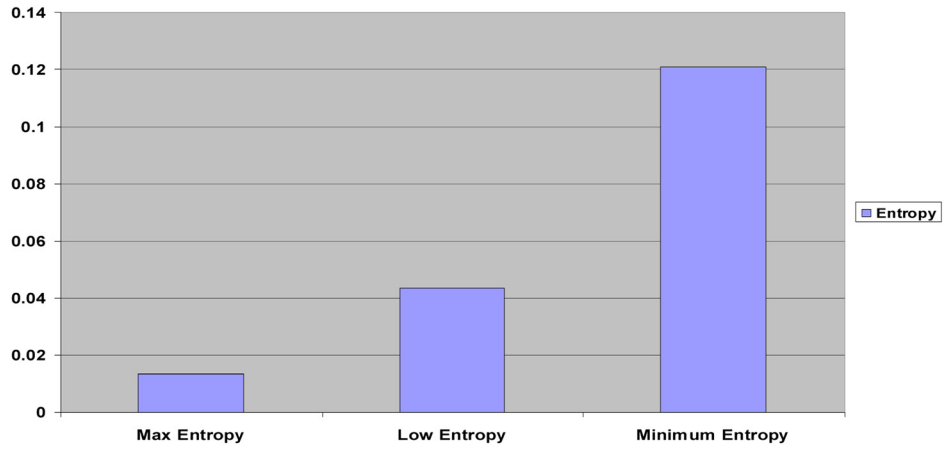

Figure 3. Sample e_1 values for maximum, low, and minimum entropy.

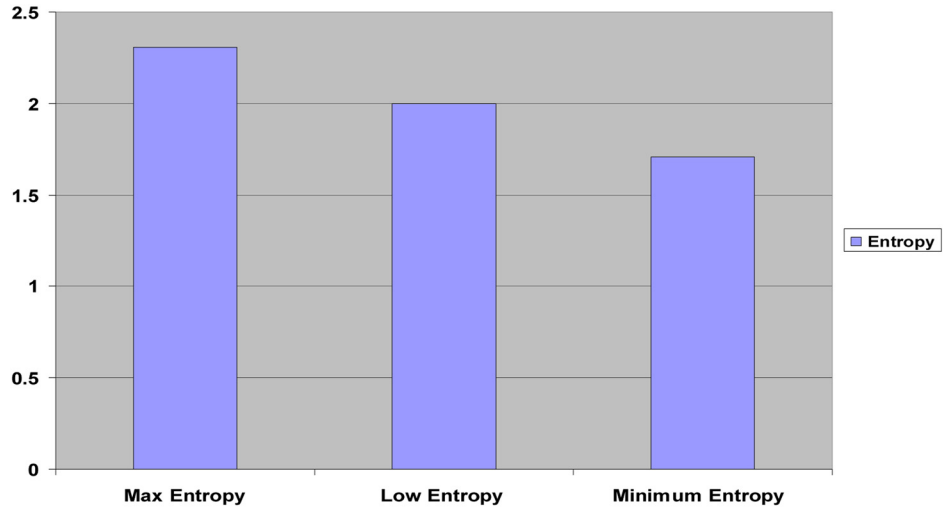

Figure 4. HD entropy: sample e_2 values for maximum, low, and minimum entropy.

string with 32 different runs of 1's and 0's, there should be 16 run lengths of size 1, eight run lengths of size 2, four run lengths of size 3, and so on. Figure 3 shows three characterizing e_1 values using the normalized entropy algorithm: the maximum entropy, low entropy wherein all the run lengths are approximately equal, and a minimum or no entropy when the string is essentially all 1's or 0's. As can be seen, higher entropy results in a lower value for e_1. It should be noted that by using $1/e_1$ this entropy metric would be more intuitive in that minimum entropy would always be less than maximum entropy.

We used another entropy measure, based on Hamming distance (HD) (number of bits that are different between two strings), shown in Eq. (2) (e_2). In this case, N refers to the maximum Hamming distance between two bytes and x refers to the normalized i HD of the actual strings. This HD is calculated on a moving window along a string in a forward direction. Figure 4 shows the general trend for this measure. For example, if the string contains a pattern of 1's and 0's such that the Hamming distance is always the same (1100110011001100), entropy would be low. An additional HD measure (e_3) was also used, which is similar to Eq. (2), except that the HD is calculated by moving the window in any direction.

$$e_2 = \frac{\sum_{i=1}^{N} \log_2\left(\frac{1+|x-1.0|}{1.0}\right) * 1.0}{N-1} \tag{2}$$

We used these measures because they are insensitive to string length, the expected values are easy to compute, and they converge quickly.

Scrambling Techniques

In order to test encryption methods, we ran several experiments using the three entropy measures each in combination with four scrambling algorithms. A test run consisted of 500 randomly generated strings, with an average length of 310 bits for each of the typical single block barcode symbol sizes of 12×12 up to 26×26. Each of these symbol sizes was tested with module sizes from 12 to 18 pixels. This generates 28,000 individual barcodes. Each test also has an

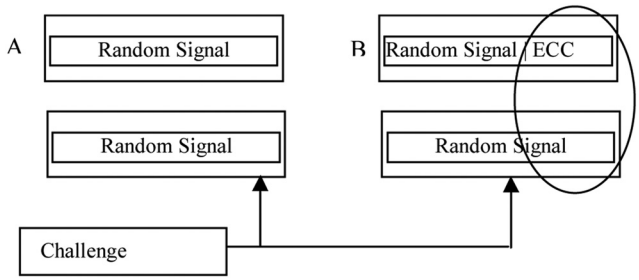

Figure 5. Effect of ECC on challenge result.

associated scrambling algorithm and entropy measure. Each test was run twice; once using the maximum number of ECC bits allowable for the size and once using additional randomly generated data where the ECC bits would normally be inserted. A total of 672,000 barcodes were tested with half containing ECC bits and half without ECC bits.

The four scrambling algorithms consisted of the following:

1. XOR (Exclusive OR): A randomly generated string of the same size as the entire string (message + ECC bits) is operated on using an exclusive OR function with the input string.

2. Structural scramble: Divide the string matrix into equal sized structures (squares, rectangles, etc.). Swap bits within each structure so that the new structure is a mirror image of the original.

3. Even check bits: Add a check bit at the end of each row and column so that the total number of black modules is even.

4. Odd check bits: Add a check bit at the end of each row and column so that the total number of black modules is odd.

Tests

"Challenging" the entropy of the string set with another random string should result in different responses if the string is not as entropic as the challenge string. For example, Figure 5 demonstrates that when the completely random number is challenged there should be no difference in the entropy between the two randomly generated strings. However, when the string contains ECC, there may well be a detectable difference in the entropy between the string with ECC and the randomly generated challenge string. This is indicated by the oval surrounding the place within the string that contains ECC. One of the main objectives of these experiments is to determine whether that difference is detectable. If so, finding the best scrambling algorithm along with the most sensitive entropy measure to highlight differential effects leads us to a recommendation for adding security mechanisms to two-dimensional barcodes.

RESULTS

One of the goals of the experiment was to determine whether the entropy measures we used, if any, were able to distinguish ECC strings from non-ECC strings. Figure 6 compares the three measures on ECC strings, while Figure 7 compares them on the non-ECC strings. All four scrambling algorithms are represented, and the results are given in percent change of entropy between the input and output strings. These graphs are helpful in that they confirm that the measures perform as expected, especially for the ECC strings. As the size of the symbol increases, the number of possible run lengths also increases, and entropy should increase when scrambling structured strings, in this case ECC strings. In the case of non-ECC strings, the results are more interesting. The normalized entropy measure changes very little, indicating that scrambling non-ECC strings results in little change to entropy and may not be helpful for detecting differences in scrambled non-ECC strings. However, the other two entropy measures show more promise in being able to detect changes in non-ECC strings, as they are significantly different than normalized entropy results and are not as close to the value of 1.0 expected of completely random strings.

To see the difference between ECC and non-ECC strings, Figure 8 presents the data for normalized entropy a little differently using all the scrambling techniques. For each symbol size, the result is again the percent change of entropy between the input and output strings. For example, results for the 12×12 symbol shows that the change in mean entropy for the string containing no ECC was very small. This makes sense because scrambling a fully random string should result in another random string. The entropy of 12×12 symbols with ECC, however, increased (by more than 5%) after scrambling. This is also logical as scrambling a string containing nonrandom bits should result in a more random string. In general, the change in entropy results is affected by symbol size, as the number of possible run lengths increase.

The next thing to look at is the population statistics. Figures 9 and 10 are the input and output mean e_1 values for ECC and non-ECC along with the standard error for the XOR scrambling algorithm. Unfortunately, these results are representative of the results for each of the scrambling algorithms. The main conclusion to be drawn from these figures is that there is no way to distinguish between ECC and non-ECC strings by looking at difference in input or output means only. Thus, "reverse engineering" the ECC method used would not be possible using these metrics, as the mere existence of ECC in the original strings cannot be elucidated by these methods.

In Figure 11, the data for Figs. 9 and 10 are combined to show the input and output normalized entropy (e_1) means for ECC and non-ECC using the XOR scrambling algorithm. This figure highlights the difficulty in finding differences using population statistics.

Results are pronouncedly different in comparing ECC and non-ECC using the Hamming distance measures (e_2 and e_3) than they are for the normalized entropy measure. Figure 12 shows the first HD measure for ECC and non-ECC signals. Here, there is a detectable difference between the signals that contain ECC and those that do not. Recall from Fig. 4 that as a signal becomes less random, this

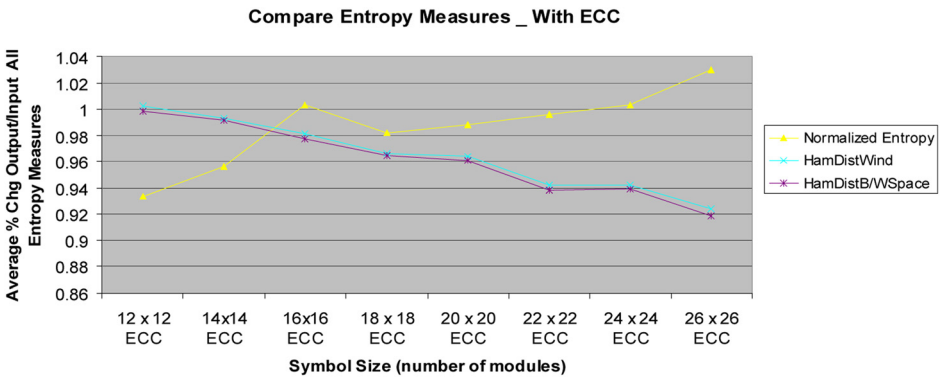

Figure 6. Comparison of entropy measures—strings with ECC.

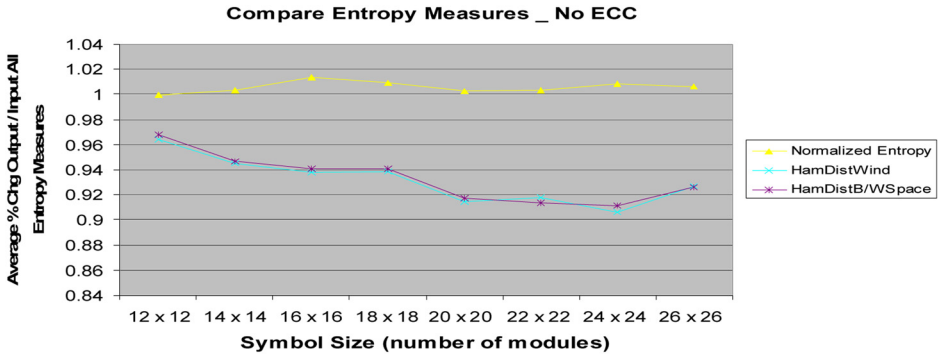

Figure 7. Comparison of entropy measures—strings without ECC.

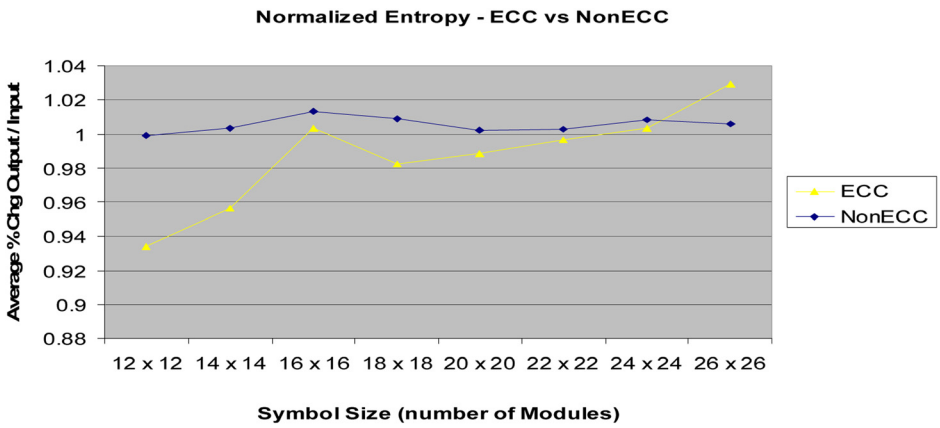

Figure 8. Normalized entropy results (e_1) —ECC vs non-ECC.

entropy measure decreases. Fig. 12 demonstrates that the change in entropy after scrambling results in a lower entropy value (meaning less randomness) for both the ECC and the non-ECC strings. For the majority of the symbol sizes, e_2 output values are lower than input values. In other words, ECC strings start out with more structure than the non-ECC strings and become more random after scrambling.

All of the scrambling algorithms, along with the Hamming Distance measures, give us similar results in terms of the separation between ECC and non-ECC strings. As an example, Figures 13-15 show the population statistics for the input and output means when using the XOR scrambling algorithm. The other scrambling algorithms show similar results. The data points contain only half the error bar in order to show the magnitude of the standard error. Obviously, these two populations overlap and cannot be distinguished with any reasonable level of statistical confidence. In general, while the change in entropy after scrambling of the non-ECC strings is detectable, the population statistics (Figs. 14 and 15) show that detecting the difference between ECC and non-ECC signals using population means is not easy, and is probably impossible (certainly impractical) using these methods.

We also looked at the correlation between the mean input and output values for each scrambling algorithm.

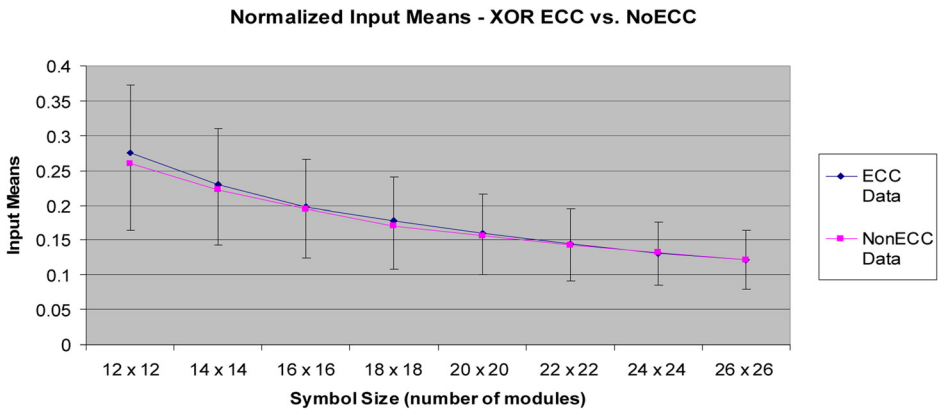

Figure 9. Input means comparison—ECC vs non-ECC.

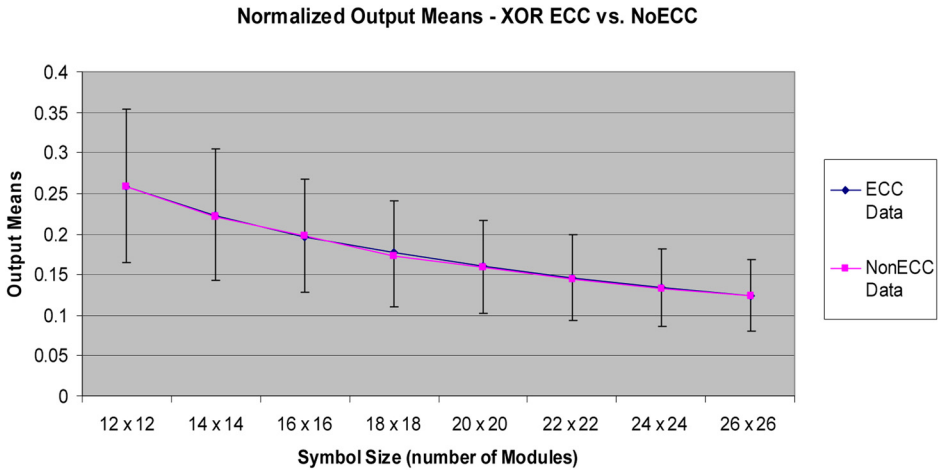

Figure 10. Output means comparison (e_1)—ECC vs non-ECC.

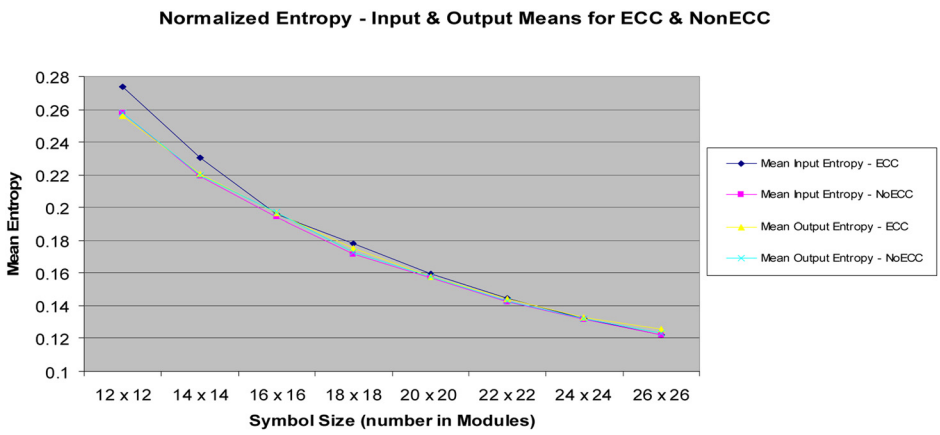

Figure 11. Mean entropy comparison (e_1)—ECC vs non-ECC.

Figures 16 and 17, which show the results for XOR and even checkbit scrambling, are representative of the results for all the algorithms. Here the results are more promising for distinguishing ECC and non-ECC strings. Interestingly, there appears to be less correlation between input and output values for ECC strings than non-ECC strings. This could be indicative of the amount of randomness introduced into ECC strings that are scrambled. Also, the differences are more pronounced when the even and odd checkbit scrambling algorithms are used.

DISCUSSION AND CONCLUSIONS

We have presented three entropy-based methods for determining the degree of randomness in a signal and the effect

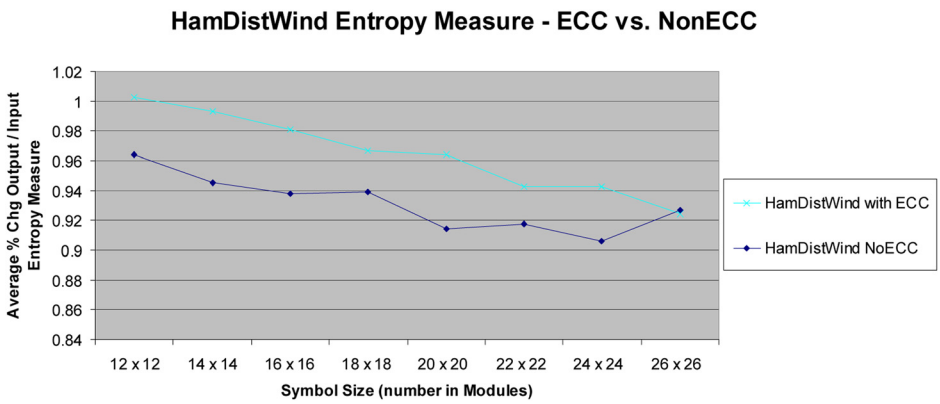

Figure 12. HD entropy measures (e_2)—ECC vs non-ECC.

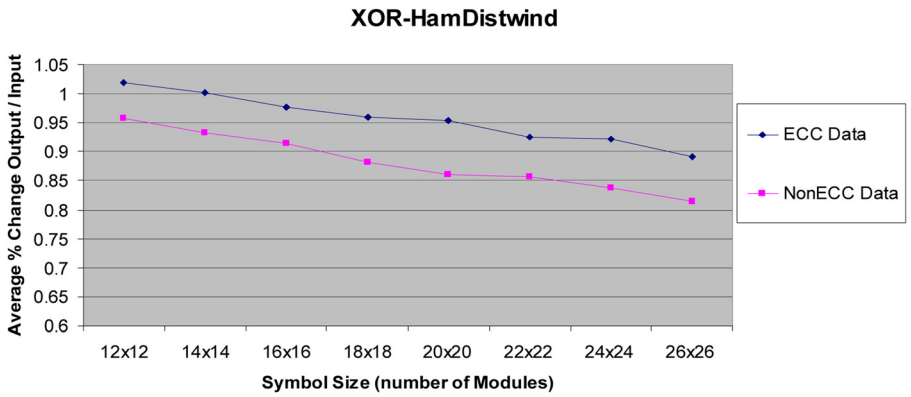

Figure 13. XOR scrambling with HD % change.

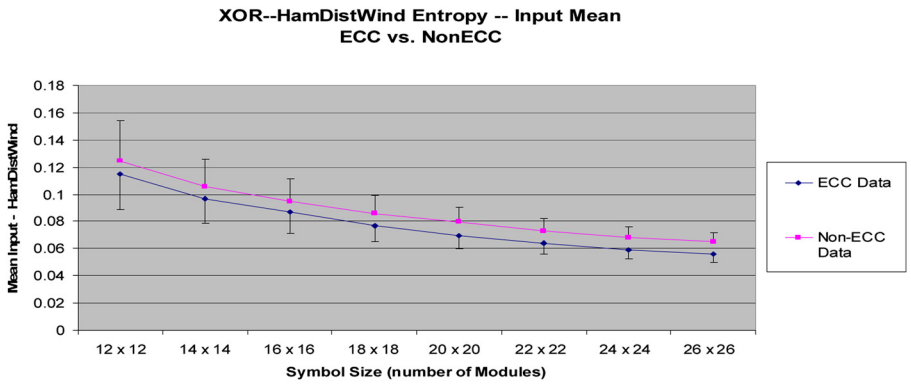

Figure 14. XOR scrambling with HD—input mean.

of scrambling on the outcome of these methods. The results point to the possibility of using the Hamming distance measures for any of the scrambling algorithms. While the scrambling algorithms had no measurable effect on detectable changes, the HD metrics were able to detect differences for structured strings more consistently than the normalized entropy measure. Removing ECC introduces the possibility of making these barcodes less robust; however, it is possible to address this issue by duplicating the data within the barcode or increasing the size of the modules.[7,8] Obviously, eliminating the ECC renders the barcode out of standard and would require specialized reading software to interpret the data, but this may be acceptable given the increase in security.

In our case, we used two-dimensional barcodes because of the ubiquity of these symbols in the supply chains of virtually every manufacturing sector. As the incidents of counterfeiting continue to rise, security at each node within the supply chain becomes more critical. The DataMatrix standard[6] does not take this type of security

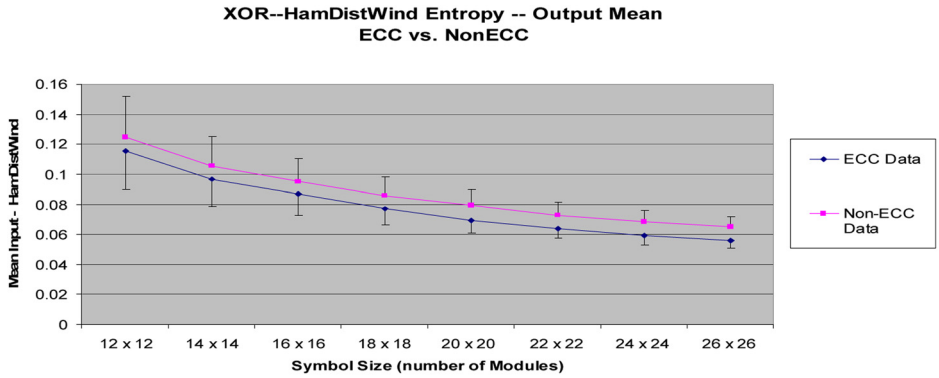

Figure 15. XOR scrambling—output mean.

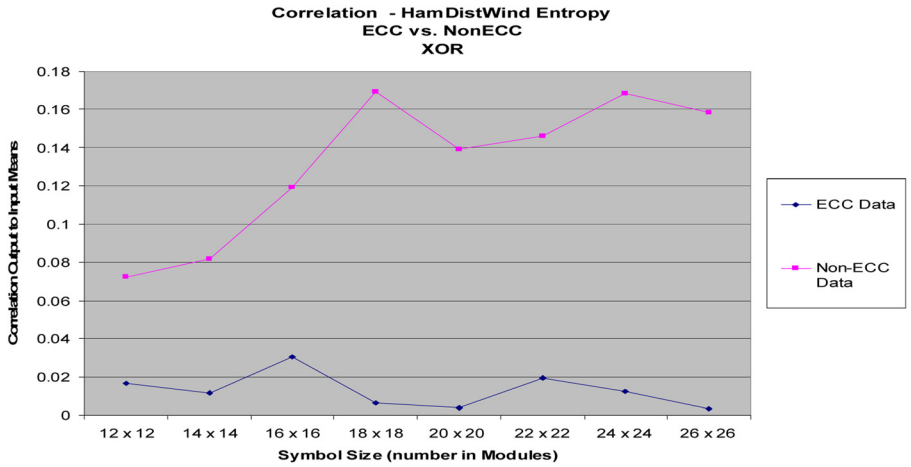

Figure 16. XOR scrambling—correlation of output to input values.

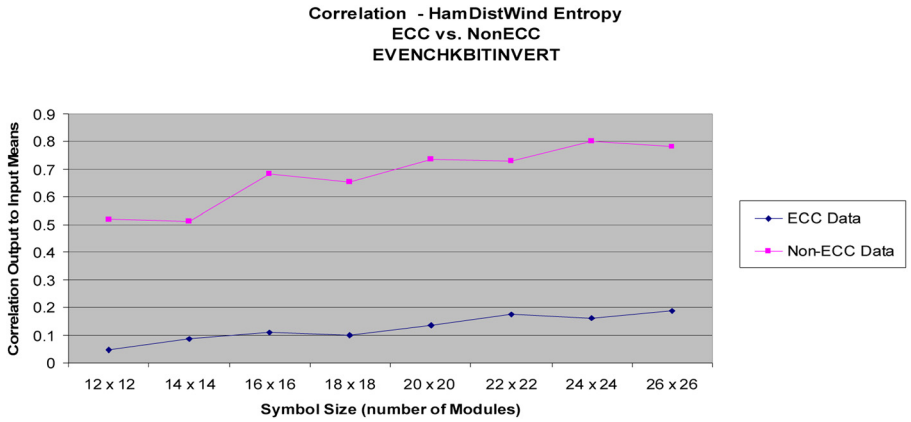

Figure 17. Square block scrambling—correlation of output to input values.

into consideration, as the ECC within the string has structure and is therefore vulnerable to attacks. We have shown that our entropy measures and the appropriate "attack" can detect the difference between a truly random signal and a signal that contains predictable structure. This feature can be used to discover whether ECC has been used on a set of materials, and, if so, potential vulnerabilities of the security data. The methods described here can also be implemented to determine whether data is encrypted, since proper encryption should also work to maximize entropy.

As shown, it is possible to interrogate the entropy of the comprised signal and compare it to the original entropy values. Additionally, it may be used to find which ECC is used in systems with two or more ECC choices. For example, Bose, Chaudhuri, and Hocquenghem (BCH) codes, Gallagher codes, Hamming codes, etc., also depend on the structural arrangement of the tiles in the barcode. The

results of applying our metrics on strings using these ECC could be different from those we have reported using Reed-Solomon ECC, but may be similarly predictable.

REFERENCES

[1] GS1, MobileCom Website, http://www.gs1.org/productssolutions/mobile/, accessed March, 2011.
[2] GS1, EPCglobal Homepage, http://www.epcglobalinc.org/home, accessed March 2011.
[3] Microsoft Research, High Capacity Color Barcodes (HCCB) Website (2011), http://research.microsoft.com/en-us/projects/hccb/default.aspx, accessed March 2011.
[4] OMA, Open Mobile Alliance Website (2010), http://www.openmobilealliance.org/, accessed March 2011.
[5] C. E. Shannon, "Prediction and entropy of printed English," Bell Syst. Tech. J. **30**, 50–64 (1951).
[6] International Standard ISO/IEC 16022, "*Information Technology—Automatic Identification and Data Capture Techniques—Data Matrix Bar Code Symbology Specification*," 2nd ed., ISO, 15 September, 2006.
[7] S. J. Simske, M. Sturgill, and J. S. Aronoff, "Effect of Copying and Restoration on Color Barcode Payload Density," *DocEng '09: Proc. 9th ACM Symposium on Document Engineering* (ACM, New York, 2009) pp. 127–130.
[8] M. Vans, S. J. Simske, and J. S. Aronoff, "Barcode Structural Pre-Compensation Optimization," *Proc. IS&T's* NIP25: International Conference on Digital Printing Technologies (IS&T, Springfield, VA, 2009), pp. 167–169.
[9] E. Schuman, "Wal-Mart Stung in $1.5 Million Bar-Code Scam," *EWeek.com, Enterprise Application News*, January 5, 2005. http://www.eweek.com/c/a/Enterprise-Applications/WalMart-Stung-in-15-Million-BarCode-Scam/, accessed April 12, 2011.
[10] World Economic Forum, January 2009, http://www.weforum.org/s?s=-counterfeiting (2009), March 2011.
[11] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," J. Soc. Ind. Appl. Math. 8(2), 300–304 (1960).