Factorial Number Systems and their Application to Steganography

Brian Mearns *Graudate Student Member, IEEE*

Abstract

This document describes a novel set of number systems based not on powers of a given base, but on factorials of the system's radix. The construction of these number systems is based on the process of ordering a set of sortable elements and provides a unique one-to-one mapping between such an ordering and the integers in the range $[0, R!)$ for a set of R elements. In this way, the particular order of any otherwise arbitrarily arranged sortable elements can be used to convey a numerical value and therefore any arbitrary message restricted only by the number of elements used. As a steganographic tool, this allows a message to be hidden in an inconspicuous list such as deck of cards, shopping list, or the palette of an image file.

I. Review of Digital Number Systems

A number system is a map between numerical values and representations of those values. Generically, we might say that A represents the value X in some number system $\mathcal{S}$. We denote this as $A \overset{\mathcal{S}}{\Rightarrow} X$, or simply $A \Rightarrow X$ if the particular number system, $\mathcal{S}$, is obvious from the context.

The most commonly used number systems are *digital number systems*, in which values are represented by a string of digits:

$$a_{N-1}a_{N-2}\cdots a_2a_1a_0$$

For clarity, we will represent digit strings as ordered lists, such as

$$(a_{N-1}, a_{N-2}, \ldots, a_2, a_1, a_0)$$

When appropriate, the vector will be subscripted to indicate the number system.

Every digit in such a string makes a certain contribution to the value represented. Specifically, this contribution is the product of the digit itself (a_i for $i = 0, \ldots, N-1$) and the weight, W_i, associated with the digit's position. The total value is the sum of these contributed values:

$$(a_{N-1}, \ldots, a_0)_{\mathcal{S}} \Rightarrow \sum_{i=0}^{N-1} W_i a_i;\ W_i = f_{\mathcal{S}}\left(i\right) \tag{1}$$

The definition of W_i at the end of (1) indicates that the weight of each digit position is a function defined according to number system whose only argument is the digit position, i. Specifically, the digit weight, W_i, and digit value, a_i, are independent.

A. Positional Notation Number Systems

The familiar decimal, hexadecimal, and binary number systems belong to a subclass of the digit number systems called *positional notation number systems*. A positional notation system is defined by an integer value, referred to as the *base* of the system. All of the digits in a positional notation system with base B are integers in the closed range $[0, B-1]$, and the weight of each digit position is an integer power of the system's base. Specifically,

$$W_i = B^i$$

Deriving from equation 1, we define a positional notation number system with base B as follows:

$$(a_{N-1}, a_{N-2}, \ldots, a_2, a_1, a_0)_{\mathcal{P}_B} \Rightarrow \textstyle\sum_{i=0}^{N-1} B^i a_i$$
$$a_i \in \{0, 1, \ldots, B-1\} \text{ for } i = 0, \ldots, N-1$$

Note that the $\mathcal{P}_B$ subscript on the digit string indicates a positional notation representation with base B.

II. One-to-One Number Systems

Orthogonal to the digital number systems, we define the classification of *one-to-one integer number systems* as those number systems which have exactly one way to unambiguously represent every integer in a particular closed range. To define this classification more rigorously, let L and D be two integers, with $D \geq 0$. A one-to-one integer number system with range $[L, L+D]$, must satisfy the following three requirements:

1) $(X \in \mathbb{Z}) \wedge (L \leq X \leq L+D) \rightarrow \exists A : A \Rightarrow X$
2) Given $A \Rightarrow X$ and $B \Rightarrow Y$, $X \neq Y \rightarrow A \neq B$
3) Given $A \Rightarrow X$ and $B \Rightarrow Y$, $A \neq B \rightarrow X \neq Y$

Requirement 1 is called *full-coverage*, and it requires a one-to-one integer number system to be able to represent every integer in the specified range $[L, L+D]$. Requirement 2 is called *uniqueness* and it requires that different values have different representations. Finally, requirement 3 is called *non-redudancy*, and it requires that there be no more than one way to represent a given value.

There are certain applications that benefit from having multiple ways to represent certain values (such as redundant binary representation), and there are certain applications that benefit from trading off full-coverage in exchange for partially covering a larger range more compactly (such as floating point numbers). However, the steganographic application discussed in section V requires full-coverage so that the represented values can be used as arbitrary binary objects. In addition, non-redundancy provides greater capacity for this application because it covers a larger number of unique values than a corresponding number system that has multiple ways of representing some values. Finally, uniqueness is necessary to communicate unambigously; without it a given representation could correspond to multiple values and therefore multiple messages.

The following sections will address each of these requirements in turn.

III. Uniqueness of Digital Number Systems

All digital number systems provide uniqueness (requirement 2). This can be seen by noting that equation 1 can be rewritten so that the number system, $\mathcal{S}$, is a function with the digit string as the argument and the value represented by the string the result:

$$\mathcal{S}\left(\,(a_{N-1},\ldots,a_0)\,\right) = \sum_{i=0}^{N-1} W_i a_i$$

In the terms of requirement 2, this looks like:

$$\mathcal{S}\left(A\right) = X$$

$$\mathcal{S}\left(B\right) = Y$$

To violate uniqueness would require $X \neq Y$ and $A = B$:

$$\begin{aligned}\mathcal{S}\left(A\right) = X \quad &\neq \quad Y = \mathcal{S}\left(B\right) = \mathcal{S}\left(A\right)\\ \mathcal{S}\left(A\right) \quad &\neq \quad \mathcal{S}\left(A\right)\end{aligned}$$

Since this is a contradiction, uniqueness cannot be violated.

IV. Introducing Factorial Number Systems

This section will present the defining characteristics of factorial number systems and show that these characteristics are sufficient for making these systems finite, one-to-one integer number systems. Rationalization for the use of these number systems in light of other sufficient number system will be provided later, in section V.

Factorial number systems are a family of integer digital number systems where each number system is defined by an integer value called its *radix*. The weight of each digit position for a factorial number system is given by

$$W_i = \frac{R!}{(R-i)!} \text{ for } i = 0,\ldots,R-1 \tag{2}$$

where R is the radix that defines the number system.

The other defining characteristic of factorial number systems is that the digits are restricted as follows:

$$a_i \in \{0,1,\ldots,R-1-i\} \text{ for } i = 0,\ldots,R-1 \tag{3}$$

This is unusual compared to more familiar number systems where the possible digit values are the same for every position. In a factorial number system, the least significant digit position has the most digits to choose from (every integer from 0 to $R-1$, inclusive), with each subsequent position loosing one possible digit off the high end of the range. Shortly, it will be shown that this construction provides the non-redundancy requirement for a one-to-one number system.

Notice that in both (2) and (3), the counter i only runs up to $R-1$, instead of the more generic $N-1$ used previously. This is an important aspect of factorial number systems in that it causes these systems to be finite. As will be shown, the factorial number system with radix R can only represent integer values in $[0, R!-1]$. This is sufficient and appropriate for the steganographic application which will be discussed in section V, and is necessary due to the definitions of W_i and a_i in (2) and (3), respectively.

Before detailing the implications of these characteristics, we will summarize the above in the following description of a factorial number system with radix R:

$$(a_{R-1}, a_{R-2}, \ldots, a_2, a_1, a_0)_{\mathcal{F}_R} = \textstyle\sum_{i=0}^{R-1} a_i \frac{R!}{(R-i)!}$$

$$a_i \in \mathbb{Z};\ a_i \in [0, R-i-1] \text{ for } i = 0, 1, \ldots, R-1$$

$$R \in \mathcal{Z};\ R > 0$$

A. Factorial Number Systems are One-to-One

Factorial number systems are a subclass of digital number systems and therefore satisfy the uniqueness requirement of one-to-one number systems, as shown above in section III. It will now be shown that factorial number systems also satisfy the two remaining requirements of a one-to-one number system: full overage and non-redundancy.

Let C_i be the set of all possible values that digit i can contribute to the total value:

$$\begin{aligned} C_i &= \{a_i W_i | a_i \in \{0, 1, 2, \ldots, R-i-1\}\} \\ &= \{0, W_i, 2W_i, \ldots, (R-i-1) W_i\} \end{aligned}$$

C_i represents the set of all values that can be represented with all digits except digit i locked in at 0. Notice that if the set were extended by one more element, that element would be

$$(R-i) W_i = (R-i) \frac{R!}{(R-i)!} = \frac{R!}{(R-i-1)!} = W_{i+1}$$

We can thus say that this set spans the range $[0, W_{i+1})$, but only with a precision of W_i (in other words, each element in the set is W_i away from its nearest neighbors).

We can likewise find the set $C_{i-1} = \{0, W_{i-1}, \ldots, (R-i) W_{i-1}\}$, which covers the range $[0, W_i)$ with a precision of W_{i-1}. We now add each element of C_{i-1} to the first element of C_i to construct the following:

$$\{0+0, 0+W_{i-1}, 0+2W_{i-1}, \ldots, 0+(R-i) W_{i-1}\}$$

Notice that the final element in this constructed set is a distance W_{i-1} away from the next element of C_i, which has a value of W_i:

$$\begin{aligned} (R-i) W_{i-1} + W_{i-1} &= (R-i+1) W_{i-1} \\ &= (R-i+1) \frac{R!}{(R-(i-1))!} \\ &= (R-i+1) \frac{R!}{(R-i+1)!} \\ &= \frac{R!}{(R-i)!} \\ &= W_i \end{aligned}$$

We have therefore filled in the W_i distance between the two lowest elements of C_i to a precision of W_{i-1}. We can likewise do this with each element of C_i to span the entire range $[0, W_{i+1})$ with a precision of W_{i-1}. This represents all the values that can be constructed using only digits i and $i-1$, with all other digits set to 0.

By induction, we can use the next digit down to fill in these gaps to a precision of W_{i-2}, and so on all the way down to a precision of W_0 with the least significant digit. By (2), $W_0 = \frac{R!}{(R-0)!} = \frac{R!}{R!} = 1$, so we have therefore covered every integer in $[0, W_{i+1})$ using i digits.

By letting $i = R-2$ (the second to last digit), we can cover all the integers in $[0, W_{R-1}) = \left[0, \frac{R!}{(R-R+1)!}\right) = [0, R!)$. This still leaves the last digit, but (3) dictates that this will always have a digit value of 0, and so contributes nothing to the value. The factorial number system of radix R therefore satisfies the full-coverage requirement for the range $[0, R!-1]$.

Finally, notice that the sets constructed above by adding each element of C_i to C_{i-1} do not overlap, because the largest value in C_{i-1} is less than the distance between adjacent members of C_i. Factorial number systems therefore also satisfy the non-redudancy requirement.

V. Ordered-set steganography

Steganography is the science of hiding data within an inconspicuous *cover object*. It is a branch of information theory and closely related to cryptography. Cryptography, however, intends to conceal the meaning of the data, where as steganography seeks to conceal the very existance of the data.

A basic tenet of steganography is that any arbitrary choice available to a message generator is an opportunity to encode information. A classic example from history is Paul Revere's friend in the Old North Church. His choice was to hang one lantern or two in the bell tower, and the decision he made conveyed information regarding the movements of the British troops.

The factorial number system constructions described above are based on the choice of how to order a set of items. Specifically, consider a set of unique items with some method for presenting the set in an ordered list. For the sake of steganography, each possible ordering of the presented list could represent a different message. Generically, we will assume the message is an arbitrary binary objecy: this could be ASCII text, or a compressed image file, or simply an index into a codebook of pre-arranged messages. Such a set containing R elements has $R!$ ways it can be ordered and can therefore encode a message up to $\log_2 R!$ bits.

In constructing this list, the message generator will begin by choosing the first element from the set of all R possibilities. This corresponds to the least signficant digit in the radix-R factorial number system. This leaves only $R-1$ items to choose the second item from, which corresponds to the second-least significant digit, and so on. Notice that for the final element, there is only one option left, which means it is not a choice for the message generator and therefore cannot convey any additional information. This corresponds to the fact that the digit $R-1$ in the factorial number system always has a digit value of 0 and therefore does not contribute to the value.

Thus, each element in the list corresponds to a digit in the factorial number system whose radix is equal to the number of elements in the set. The restriction on digit values presented in (3) is derived from this process of ordering a set of items. The weighting scheme shown in (2) was then established so that the number system would be one-to-one as shown in section IV. This was necessary for the intended steganographic application: full-coverage was nessary so that arbitrary binary messages could be encoded, and uniqeness was necessary so that the encoded message would be unambiguous. Non-redundancy is not strictly necessary but allows a greater number of messages to be encoded in the same size list (in steganographic terms, it provides a greater capacity).

The digit value associated with each element is based on its position in an agreed upon *canonical order* (for instance, a sorted lexicographical order). Importantly, the translation between elements in the list and digit values is not universal, meaning that a given element in a given set does not always represent the same digit value. This is a neccessary consequence of the fact that not all digits can take on the same digit values. Instead, the digit value of an element comes from the canonical order of the elements that have not yet been used. For the first item (corresponding to the least significant digit), this is the entire set of items. For the next item, it is the original set of items, minus the item chosen for the first position.

To be clear, the unused items should be re-sorted after each item is chosen for the presented list, and the digit value for the next chosen item is equal to its 0-based index into this resorted set. For practical purposes, as long as the canonical order does not otherwise change when an item is removed, a full resort is not necessary. Instead, each element should be assigned an initial value based on its index into the canonical order of the full set. Each time an item is removed, all the items above it will have their values reduced by one in order to fill in the gap.

A. Example

Consider the following set of four items:

$$\{\text{apples},\ \text{bananas},\ \text{carrots},\ \text{deer-meat}\}$$

The canonical order will be determined by ascending lexicographical sorting (a, b, c, d) for all subsets.

There are $4! = 24$ different ways to order this set. A particular ordering is

$$(\text{bananas},\ \text{deer-meat},\ \text{apples},\ \text{carrots})$$

Here we assume that the right-most item (“carrots”, in this case) was chosen first and therefore corresponds to the least significant digit. Additional elements were added to the list moving right to left so that the left most item corresponds to the final digit.

When “carrots” was chosen for the first item, all four items were available, so its digit value is equal to its index into the canonical order of the set, which is 2. With this item removed, the canonical order of the remaining elements is

$$(\text{apples},\ \text{bananas},\ \text{deer-meat})$$

“apples” still has a digit value of 0, which will take up the second digit position. The removal of “apples” from the set will cause the two remaining items to move down by one in the canonical ordering, leaving “deer-meat” at index 1, which will be its digit value in the third digit position. This leaves only “bananas” which necessarily has a digit value of 0. Our factorial number system representation of this list is therefore:

$$(0, 1, 0, 2)$$

From the (1) and (2), this has a value of

$$(0 \times 24) + (1 \times 12) + (0 \times 4) + (2 \times 1) = 14$$

VI. Conclusion

The family of number systems described in this paper are closely modeled on the construction of a list from a set of unique items and creates a simple one-to-one mapping between every possible ordering of such a list and the integers in $[0, R!)$ for a set of size R. This has applications in steganography where each integer value can correspond to an arbitrary message which can therefore be encoded into the particular order chosen for an incospicuous ordered set, such as a shopping list or the palette of an image file.

The factorial number system equivalent to a set of R items can encode up to $\log_2 R!$ bits. The rapid growth of the factorial causes this to attain useful levels with lists of practical size. A set of 128 elements, for instance, can encode over 716 bits, equivalent to 89 extended ASCII characters. Representing the set itself requires an average of 7 bits per element, which makes 896 bits for the entire set, giving a data rate of more than 79%. A 256 element set provides over 1600 bits of capacity with a data rate of 82%.

Sets of this magnitude are typical in palette based image files commonly used on the World Wide Web. While the data rate taking the file as a whole may be quite small compared to other steganographic techniques (for instance, discrete cosine steganography used with JPEG files, and least-significant-bit stegnagraphy used with various true-color lossless image formats), this technique benefits from not having to change the visual image itself, leading to greater transparency.