**Four Fundamental Information Symbols Give Maximum Compression, Data Standardization, and Novel Encryption**

Joseph E. Johnson, PhD jjohnson@sc.edu

**Abstract**

 Our research on identifying an optimal fundamental symbol set for the (a) maximum compression of information has led us to a new set of fundamental symbols for the representation of information, (b) a better means for standardizing data for self-identification, and (c) two novel means of encryption. This optimal **4 F**undamental **S**ymbol “**4FS**” set, consists of: two fundamental data symbols **1** & **0** and two fundamental delimiter symbols **(** & **).** We propose a reformulation of information representation and a new information framework with three goals: (1) Optimal Data Compression: We show that maximum information compression can be achieved using exactly these four fundamental symbols: “0”, “1”, “(“, and “)”, where each symbol now represents two bits of Renyi (or Shannon) information. This significantly improves both data storage and data transfer rates with Big Data. This design removes the useless leading zeros of binary numbers for character identification and utilizes rank order character and word frequency to correspond to rank order binary values with variable length strings. (2) Data Standardization: With these four symbols, we demonstrate an efficient means whereby all information can be self-defining to both machines and humans, significantly supporting new levels of artificial intelligence (AI), machine learning (ML), and unlimited nesting for complex expressions. (3) Novel Data Encryption: We next show that this design supports two novel means of encryption that additionally support all current encryption methods. (4) We then show how this methodology can be deployed within the current computer design. (5) DNA Encoding: The system also supports encoding into the four ACTG components DNA. The 4FS symbol set is to information what the fundamental particles are to Physics and the ACTG components are for the DNA information in living systems.

Distinguished Professor Emeritus, Department of Physics and Astronomy, College of Arts and Sciences

University of South Carolina, Columbia SC, 29208; © February 18, 2023 & Patent Pending

**Background, Significance, and Overview:**

The fundamental representation, storage, communication, processing, and components of information technology now frame the basic infrastructure of our current science, economy, industry, national defense, and our very civilization as documents, images, networks, and numerical data tables. Information consists of strings of symbols which we use to store, exchange, and process higher and higher forms of knowledge, complexity, abstraction, and understanding. But the volume of our information is rapidly increasing thus mandating larger storage and transmission costs, especially with the emerging Internet of Things (IoT), astrophysical, particle accelerator, and other data sets in the physical, earth, biological, and economic sciences. The current standard of Unicode UTF-8 has evolved over the last 30 years from the original ASCII code to now encompass 1,112,064 characters and accounts for over 98% of web pages by using up to four variable length sets each of 8-bit bytes.

Information is currently based upon two binary **“fundamental symbols”** “**1**” & “**0**” in terms of which alphabetical, numeric, and symbolic **“characters”** are denoted by associated eight, sixteen, or larger binary numbers forming “bytes” that in turn can be used to form words, sentences, and other higher order constructs. But there is also a single delimitation symbol that separates these bits into eight-bit groups which constitutes a third invisible “delimiter” symbol which must be counted because it is not possible to frame information just in terms of only the 1 and 0 symbols without any delimitation symbol. Thus, our current system actually utilizes exactly three fundamental symbols. For each ASCII character, using Renyi second order information this gives I = log2(3) = 1.585 bits each so that **the current standard 8-bit byte actually contains 9\*1.585 bits = 14.27 bits/character** when counting the single delimiter for ASCII code. But **(a)** more frequent characters cannot be assigned to shorter binary values, and **(b)** the leading zeros carry no information content, so that information is not maximally compressed. Also, **(c)** the current system uses informationally expensive (14.27 bit) character delimiters as separators for words and sentences rather than fundamental delimiter symbols. Also, **(d)** words and common expressions are expressed as sequencies of these informationally expensive characters rather than also using a single binary value as with characters. Finally, **(e)** one must use up to four 8-bit bytes to code all Unicode characters. Although there are multiple means for UTF-8 compression, the current system is far from maximum compression requiring **(i)** larger data storage, **(ii)** and longer transmission times within and among devices. It is also **(iii)** not suitable for research on fundamental advanced information structures and **(iv)** does not easily support data standardization that can substantially enhance AI and ML. Our proposed system addresses these issues from a more fundamental point of view.

It is common knowledge that we seek maximum information compression in speech, prose, and poetry, as well as in science with succinct equations for the objective representation of nature. Our research first seeks **(1) optimal information compression** by showing that an information system of optimal compression consists of variable length strings of two fundamental data symbols “**1”** and ”**0”**, and two fundamental delimiter symbols “**(“** and ”**)”** which provide structure to the data symbols thus linking binary strings to known rank order character, word, and expression usage using frequency tables to achieve optimal compression. This significantly reduces storage space and maximizes data transmission speed. We next propose **(2) that all information be standardized and self-defining** to both humans and machines significantly enhancing AI and ML analytics to process information for complex systems without human intervention. This is critical to manage the vast data needed for complex systems, Big Data, and specifically for the emerging Internet of Things (IoT) including automated data capture, collection, and analysis. Specifically, we have expanded our previously developed and coded standard for all numerical information that tightly integrates each numerical value with its numerical uncertainty, units of measurement, and associated defining metadata (in the form of a Python dictionary usable by any software). Our algorithm automatically manages all dimensional and uncertainty analysis in mathematical calculations and critically extends the current SI (metric) system with eight (8) additional fundamental dimensions (units) that are critical for more advanced complex and socioeconomic systems. We next demonstrate how this framework leads to **(3)** **two novel data encryption means** that were not previously possible, but which can be used in addition to all existing methods thereby providing significantly more powerful encryption for military, governmental, industrial, financial, scientific, and personal security. We finally will show **(4) that this system can be easily implemented within the existing framework** and does not require hardware redesign and **(5)** can be implemented in new physical means of **encoding data in DNA with the ACTG symbols in a natural way**. We contend that these proposed fundamental symbols are to information what the fundamental particles are to quantum theory in physics, and what the four ACTG base structures (symbols) are to DNA information coded in living (biological) systems. We seek to provide a more optimal foundation for future information systems in all forms with significant impacts for complex systems, AI, ML, and Big Data analytics with significantly lower costs and greater functionality.

**Narrative:**

**(1) Optimal Compression**:

Compression of character information is not new and in fact was utilized in Morse code with shorter “.” and “-” variable length sequences for more frequent characters but separated by a pause in transmission as a single separation symbol. There have been many studies of the frequency use of characters, symbols, and words [1]. To remove the leading binary zeros in the current system, one must utilize binary numbers of variable length of fundamental data symbols, such as **1, 0.** Also to maximize information density, one must assign characters, words, and common expressions to rank order binary numbers so that more frequent items are represented by shorter binary numbers. Next, one can then ignore the informationally expensive (14.27 bit) delimiters (space, comma, …) for data by using a single fundamental delimiter symbol: **(** or **)** of two bitsto delineate alternating items in a list. The delineation of the entire item (characters, word, sentence, paragraph …) will additionally require a pair of enclosing delimiter symbols to designate it as a single entity. We additionally require that the compression design can easily operate within the existing digital information framework.

Consider the optimal number of delimiter symbols beginning with the two symbols **(** and **).** We must have two paired delimiter symbols for delineation of any structure to support variable length strings, unlimited nesting, and to especially compress delimitation rather than using the informationally expensive (14.27 bit) current ASCII expensive characters. However, within a structure, such as a word (consisting of a list of characters) or a sentence (consisting of a list of words), a paragraph of sentences, or vectors (matrices and tensors with higher order lists), it would appear at first glance that an additional neutral third delimiter (e.g. “**|**”) would maximize information density although this would then require five (5) fundamental symbols so that each symbol would then carry an information value of I = log2(5) =2.322 bits rather than log2(4) = 2 bits with four fundamental symbols discussed so far. But one can show that in any list (of characters for words, words for sentences, …), that one can enclose every other binary number with parenthesis. This gives the necessary delineation without the additional informational cost of using two fundamental characters to enclose each item. By contrast, with current technology, the ASCII delineation characters (such as a “,”, “;”, “.”, blank, etc.) each require 14.27 bits while with the proposed new fundamental characters of a **(** or a **)**, each requires only two bits. Consequently, we can conclude that exactly two delimiting fundamental symbols **(** and **)** are optimal.

Next, we analyze the optimal set of information symbols as we intend to represent characters, words, and common expressions by rank ordered numerical values. Such analysis requires that we know the use frequencies of characters and words as well as their length. We will begin with the representation of the fundamental ASCII characters for which we use a standard frequency analysis for English. It is known that both the character and word use frequency distributions are approximately log-linear which is known as Zipf’s law. We computed the information content of ASCII characters when represented by sequential numerical values multiplied by their length of character (unity) for differing numbers of fundamental data symbols 0, 1, 2… We found that one can utilize delineation of these associated numerical values using the two () symbols for each alternating value thus requiring one delineation symbol of two bits for each associated binary value. By using a standard log-linear distribution for the frequencies of the ASCII symbols, for two fundamental data symbols 0, 1 we obtained a mean information result of 7.49 bits. This would require that one now must utilize four fundamental symbols for information which can be accomplished by using 1/0 pairs with the designation of “(“=11, “)”=”10, “1” = 01, and “0”= 00. The mean information result for three fundamental data symbols is approximately the same as for two symbols. For four and higher numbers of data symbols the compression is less and less as more fundamental symbols are added. However, in order to utilize three or more data symbols (and thus five or more fundamental symbols) the requirement to achieve maximum compression with the current system designs, counters any gains in compression as one must jump from two bits each (as represented by 00, 01, 10, and 11) to utilizing three successive bits resulting in a loss of compression due to utilizing deployment within the current computing infrastructure. So that four fundamental symbols are optimal. With this designation, each new fundamental symbol, ()10, contains two bits of information and can be deployed with the current technology which is based upon binary information values. With quantum computing it is possible to base systems on other than binary states but constrained to using current technology, the four fundamental symbols are optimal because one can now use the eight, sixteen or higher order bytes as four, eight, or more binary pairs. Thus with current technology, one can ignore the hidden delimiter between bytes and represent information as variable length strings of pairs of binary 1/0 symbols.

 Using the current character frequency one can now assign binary number pairs to rank order frequency based upon usage in each information domain since frequency of words and characters varies from one domain to another. By using a general assignment of the characters in rank order usage to binary pairs, one finds that the previous eight bit byte that is assigned for a character can now be represented by a weighted average of 7.49 bits of information as opposed to the current value of 14.27 bits which includes the current delimiter. This then gives an information compression of 7.49/14.27 = 0.52 thus a 52% size reduction for rank order assignments. But our methodology also extends to the use of binary assignments to words and even phrases. It is known that the first rank order 100 words constitute 50% of all English words which we used as an estimate of the compression that one could get. We utilized the first 100 words in rank order, using the length to correspond to ASCII eight byte characters where each character used 14.29 bits of information and obtained the a 4FS information value of 1.99 mean bit value per word compared to the existing information value of 11.65 when constructed from the standard eight bit bytes. This indicates a compression to 0.17/11.65 = 0.17 or 17% of the current methodology for the compression of the first 100 words.

But any such common standard would vary from language to language and even more from domain to domain of use: (biology, mathematics, military domains, pharmacology, government sectors and other domains with differing rank order). It is well known that the determination of rank order of characters, words, and longer structures will be different in different subjects, (and even different among different authors). We envision that with periodic analysis using ML, one can periodically update the optimal rank order word-number assignment table for selected subject areas, thus providing a novel data encryption for a given military domain, corporation, organization, or scientific sector. The proposed algorithm would compute the frequency of characters and words based upon the domain specific data, and create the paired binary number for representation of both characters and words. This would then be capable of reversing the encryption back to UTF-8. This same table of correspondence linking characters and words to binary numbers would also provide the exact representation in binary for rapid translation. Since the binary correspondence tables are confidentially created by a simple ML algorithm, no human would necessarily need to know the table of correspondence. This will constitute our first significant novel means of encryption for each operational domain if desired. In fact, by using domain optimization for binary assignments, one will be able to achieve substantially greater information compression as well as highly diverse encryption keys.

 However, all of this would seem to be academic if we were restricted to the current eight-bit system design. Fortunately, our design can be executed within the current computer framework by using two adjacent binary bits to represent the four fundamental symbols with the binary pairs 11, 10, 01, 00 representing respectively (, ), 1, 0. Exact memory addressing can be done with an addition of one binary pair to the location in order to indicate where in the eight-bit byte the next string begins (if desired). This now allows four fundamental symbols in each quantized set of eight binary bits and ignores the existing quantization of length to allow information to be of variable length as defined by only the ( and ) binary pairs. This proves that the optimal compression can be easily achieved with current technology prior to a potential future 4-state fundamental quantum system or DNA encoding. This flexibility provides the first type of powerful encryption based upon multiple different rank orderings for 1,0 representations depending upon the operational environment.

**(2) Data Standardization:** We next require that all information is to be standardized and self-defining. This will enable AI and ML algorithms to autonomously seek out information in the data and determine its exact meaning and format. In the current representation of information, one has just binary numbers often without being standardized, and one often does not know if it is a binary number, or a character, a pixel in an image, a computer command in some language, a logic rule for system operation, or a mathematical expression. Many data systems have some form of standard already in place but it is not always obvious to a computer without human intervention, how to proceed with full autonomy on the data being analyzed as with medical patient data. The simplest example is tabular data with column headings of 1990, 2000, 2010, … or Q1, Q2, … where the interpretation is obvious to a person but not necessarily obvious to autonomous software without human intervention. With the proposed structure, all characters, and most common words, acronyms, and expressions will be represented by a binary number beginning with “1” or “0” and enclosed within a () structure. This means that the structures: (), (()), ((())), … and (00..binary), (000…binary) etc. do not ever appear, so they are free to be used to denote data types such as (a) free text, (b) image data, (c) mathematical structures, (d) computer commands and algorithms and (e) associated input data files for processing. One can develop methodologies based upon these new unique patterns along with a framework for knowing the data type of the file and its components, as well as a framework such as we used for the numerical data standardization with a Python type dictionary of fields and values that imply the exact meaning of that value in the form: (var1 (val1) var2 (val2)…). These unique structures can exactly describe the data that follows thus allowing AI and ML programs to correctly process data without human intervention. Then software can act autonomously in pattern discovery and associated information analytics. This possibility opens vast new capabilities for both AI and ML algorithms because they could seek out their objects of interest and know immediately how to perform their function without any human intervention (which also can prevent errors). We are currently testing the format (data\_type (data) defining metadata python dictionary) for example with numerical data: (0(6.35(1)\*mi/hr)\* (max.vel.(AM35Drone))) where “(0(” indicates that a numerical value follows and the value 6.35(1) is for scientific notation: 6.35\*10^1. One would naturally utilize the existing coding methods for web sites, files, and data that have become standard attachments to file names (exe, docx, xls, py, etc. ) and utilize all other existing standards such as ISBN, ISSN, and UPC product barcoding. This will be easily possible using the structures described above in conjunction with a Python type dictionary in the form (var1( value2) var2( value2)) to follow the specific data item to be identified between **)** and **)**. This is easily possible because of the variable length nested strings that are allowed with the delimiter pair () and the use of binary values (), (()), … (00 binary), (000 binary) that can be used to identify data types.

 **(3)** **Proposed Numerical Standardization:** Numerical values are meaningless if the units of each value are not present along with a means for automatic dimensional analysis, and likewise for uncertainty analysis. Also, the value is meaningless if sufficient defining metadata is not also attached to the value. Our proven MetaNumber (MN) software is a powerful module that can be called as a function to correctly evaluate complex strings of scientific, medical, engineering, commercial, and social numerical expressions. We are expanding the author’s recently developed standardization of numerical information [3] where each (a) numerical value has its (b) units of measurement, (c) numerical uncertainty, and (d) full meaning (using a Python dictionary) all integrated into one string. Our development has also now extended the SI (metric) system of units (mass, **kg,** or **kilogram**; length, **m,** or **meter**; time, **s,** or **sec,** or **second**; electrical charge, **qe**; temperature, **k,** or **kelvin**; and luminosity, **cd**, to now include a bit of information, **b,** or **bit**; a flop of double precision arithmetic, **op,** or **flop**; a living person, **p,** or **person**; a measure of value as the US dollar, **d,** or **usd,** or **dollar**; baryon number, **bn**; lepton number, **ln**; and a discrete quanta, **q**; of something specified in the attached metadata by a user (such as an animal, a galaxy, or an orange). Finally, we provide for a 14th “unit”, “**r**” that is a dimensionless ratio of two identical unit values. This applies to ratios such as the circumference of a circle divided by the radius of that circle indicating 2pi radians. The radian is not an actual unit in the normal sense but rather a notation that it is a specific type of dimensionless ratio, and it is totally different from a truly dimensionless value. The exact meaning of such “dimensionless” ratios is to be specified in the metadata dictionary. This expansion along with over 700 units, prefixes, and fundamental constants, is highly significant as it supports a vast increase in numerical computation capability not just in physics, but in all of science, financial systems, military operations, medical applications, and social science. These units also extend the SI system with the conservation of lepton and baryon number in addition to the electrical charge for isotope & ion specification. The software logs one’s computations so that previous results can be exactly tracked and reused. All dimensional errors are trapped so one cannot add a foot to a gallon. **All** units can now be used equally, mixed in any valid way, with results in any form. This means that metric adaption is no longer necessary although it is here treated as the default set of output units. One extremely powerful feature of this MN algorithm is that users can standardize data tables of numerical information (such as medical patient, chemical, scientific, or military data) allowing the values to be referenced as a variable in complex calculations using the form: *TableName\_ RowName\_ ColumnName*. For example, a table of the properties of the chemical elements can be referenced to find and insert the value of the thermal conductivity of gold with the following variable: *ChemicalElements\_Gold\_ThermalConductivity*. This means that users can define and reference their domain specific tables, user defined units, and user defined symbols, in the same manner and execute very complex calculations quickly and automatically regardless of the dimensional units used in the table. Our algorithm is optimal in that it occupies the smallest storage space and executes at very high speed. It can also be deployed in parallel over 15 simultaneous nodes with an associated increase in speed which can be of substantial importance. The author has used this system in physics classes where students can use their smart devices to access the MN system in the cloud and perform calculations very rapidly and solve difficult scientific problems by submitting a single string expression to their phone or tablet. A typical simple problem might be to calculate how many g’s of gravity a person experiences when they accelerate their car from 43 mi/hr to 89 mi/hr in a period of 2.34 sec. The problem is solved by entering: ((89\*mile/hour-43\*mile/hour) /(2.34\*sec))!ag. The exact tracking of input and output expressions exactly documents all computations with results returned in standard MN form.

 A significant amount of numerical information is in the form of the “probability” for an event to occur such as the probability for it to rain tomorrow at a certain location. A probability is always a nonnegative dimensionless value from 0 to 1 inclusively. But it should have the correct level of accuracy. If one is asking for the mathematical probability of a random point to fall somewhere on the circumference of a circle in the second quadrant, then one can compute the exact real number for that to occur as 2\*pi /4, a real number with an “infinite” number of significant digits. But all probabilities in the real world are very approximate in accuracy. If one were told that the probability of rain tomorrow at their location was 37.93283376211%, they would not believe that any model could make such an accurate prediction, nor would it even be useful. Thus, probabilities should be given at the accuracy level that is meaningful. One way to do this is by using binary numbers to state the numerator of the associated fraction such as 1011 where one assumes that the denominator value is of the same length and is in this case 1110. This ratio always allows for a midpoint of 50% with an odd number of binary values of that length and where 0000 means a zero probability and 1110 is 100% probable. This method quantizes the probability to that specific accuracy level. This pertains to the coding of standardized numerical information as described above. Finally, it must have its defining metadata attached to define what the probability represents.

 **(4) Novel Data Encryption**: We have described the use of domain specific binary encryption for characters, symbols, words, and larger structures and how that can be fully automated and designed for specific domains and kept totally confidential from all humans if desired. But the second means of encryption is even more powerful because normal encryption only uses very complex patterns to transpose the binary data sequences for information. But with the four basic symbols used here for encoding information, one has not two, but four symbols and they can be transposed in 4\*3\*2=24 ways rather than two. Here one can mix the delimitation symbols (,) with the data symbols 1,0 so that one cannot seek information patterns in the encrypted representation to break the encryption. Furthermore, with such maximum domain specific compression, the lack of patterns such as leading zeros and non-rank ordered correspondence, defeat many traditional methods of decryption. Thus, information can be encrypted by mixing data symbols with delineation symbols! Furthermore, the total number of symbols in the structure is maximally compressed leading to faster data transfers. Then in addition, any existing encryption methodology, known to those skilled in this art, can be utilized on top of this already encrypted structure (such as multifrequency data bursts). Since our compression algorithms can also act to convert data back to normal Unicode UTF-8 correspondence with binary numbers, it follows that although all computer code will retain its standard UTF-8 form at this time, all data can be stored and transmitted in this new compressed form with standardization and encryption as described. Prior to actions by existing software code, the data is to be converted back to standard UTF-8 form for processing by existing algorithms and software.

 **(4) Data Storage in DNA using four symbols:** We have concluded that an optimal encoding of information is done with four fundamental symbols, two for data **1,0**, and two for delimitation and structure, **(,)**. Although the symbols are different, the nuclides A, C, T, and G have evolved to encode all information for organic life in DNA. Recently9,10, George Church, Ph.D., first pioneered the idea of using short synthetic DNA as a long-term information storage medium. He is a Core Faculty member at the Wyss Institute and Professor of Genetics at Harvard Medical School and of Health Sciences and Technology at Harvard and the Massachusetts Institute of Technology (MIT). His team has converted a complete book, including 5.27 megabits of text and images, into a binary digital code, which they then encoded in DNA, and finally decoded again. More recently Postdoctoral Fellow, Dr. Henry Lee, in a Wyss Institute validation project, is developing a sustainable, low-cost approach for writing large amounts of digital information in DNA. It is known that DNA sequencies of ACTG can occur in any order which therefore has the potential for using these four symbols to encode the 10() symbols using any correspondence between the two sets. All information storage of necessity must be achieved in physical systems as has been done historically in magnetic tape, disk, and other media. DNA storage represents an incredible physical optimization of information storage at extreme compression and minimal energy. Furthermore, DNA information is extremely stable lasting tens of thousands of years! It might be possible that the ACTG symbols each express some (functionally) independent combination of ()10.

**Current Objectives, Tasks, Accomplishments and Outcomes:** We are developing and implementing: (1) Python coded algorithms on a MS Azure cloud, for each of these domains as described plus user registration & authentication, an instructional web site for operations for all users, and with progress tracking of computations. (2) We are testing the algorithm that assigns rank order binary numbers to rank order use of characters, words, acronyms, and potentially common phrases from rank order lists as provided, that then will both encode data and decode it to the original data with “(,),1,0” as described above from its UTF-8 representation. This also describes the first means of encryption when the binary correspondence is not public. (3) We utilize other coded binary structures as (), (()), (00 binary), (000 binary) … to provide a “proof of concept” design for standardizing information in addition to the standard described above for numerical information. (4) Simultaneously, we are designing an optimal algorithm for encryption of the “()10” based data using Lie group transformation patterns to encrypt and decrypt among the four fundamental symbols to be used with known encryption methods, (5) Additionally we are standing up the MetaNumber Python code[3] for numerical standardization with updates that can act as an algorithm to simplify mathematical expressions including units, uncertainty, and operations as described for tracking past results and utilizing the eight new SI units. This standard is also used to store tabular numerical data and user defined units. Upon completion of this framework, we will deploy the pattern identification algorithms [5,6,7] for both general networks and for tabular data based upon the mathematics of Lie algebras and group representation research by the author. This will be possible because of the numerical standardization of such data sets in MN format. We also study the optimization of both characters and words for optimal compression and discuss our framework in view of recent research encoding at the DNA level [8].

At first sight, the deployment and utilization of this system seems to be of substantial difficulty due to the scope of implementation, but this is not true. The essential point is that it does not have to be implemented everywhere or all at once but rather would be extremely effective and profitable in different narrow fields especially as in the engineering, physical, biological, and earth sciences, as well as within a pharmaceutical company, or a military subdivision such as naval or air force domains. In such an area, the tabular data is very familiar such as in chemistry where there is a single table for about 70 million inorganic compounds and their properties that could be standardized in about one hour. It is only when the standardization of many thousands of small tables would need to be done that such standardization would be more difficult. We are exploring data standardization in the form*: (0 binary (information) defining metadata)* where the “binary” value and metadata variables would be in a special table of acronyms for data types and associated operational software programs such as docx, xlcx, ….

In summary, we seek to: **(A)** To code the data conversion algorithm and test it on various domain data sets to determine and document the actual fractional extent of data compression and corresponding increased transmission speed. Our expectation is to achieve an overall reduction in storage to about 15% to 20% of current usage as well as for increased speed of data transmission time. **(B)** To explore and document the associated symbolic means of standardizing information using new symbol structures using the special unique codes as described including the author’s metanumber structure for numerical information. **(C)** To stand up and document the two highly novel information encryptions that are supported with the new fundamental symbols. **Value: (1)** The approximate reduction in both storage space and transmission speeds to about 20% of the current methodology can offer vast savings for all data management systems. **(2)** The alternative fundamental ()10 symbols offer highly unique formats for standardizing information so that all data is fully self-describing. This feature can provide vast new applications in Artificial Intelligence (AI) and in Machine Learning (ML) where algorithms can seek out and identify data for fully automated processing without human intervention. The author’s fully developed standardization of numerical information as described above is of great value to computational processing. Furthermore, the MetaNumber function can also evaluate any valid python expression or program segment. **(3)** The two proposed novel means of encryption were not possible without this new foundation in four fundamental symbols which offers powerful new avenues for information security. Each of these domains have extensive practical application. The exact assignments for these structures must eventually be established by an international standards board such as IEEE. We have here presented the methodology and examples for construction of a fully operational prototype system.

With approximately $60B spent each year just on storage, such data compression could result in a savings of 80% of that value and likewise increase the transmission speed by a factor of five both internally and externally for computational devices. With unique symbols available in the 4FS system, one can now easily standardize all information. With hyperlinks among data types and the associated metadata links, the collection of all information could integrate into a much more powerful autonomous entity. Specifically, this is applicable to automated data collection devices (IoT) and specifically for scientific data collection standardization compression and encryption. This would not only allow powerful new search engines to locate data but also allow the construction of AI and ML environments that without human intervention, could constantly survey the internet to discover new relationships and patterns (such as utilizing the network and data table pattern search engines designed by the author). Finally, encryption of one’s data, unique to their a corporate, military, or personal framework as described, could substantially enhance data security.

**References:**

1. Crystal, David 1995 The Cambridge Encyclopedia of The English Language, Cambridge University, ISBN 0-521-40179-8
2. Zipf, George K. 1935 The psychology of language, Houghton-Miffin; and Zipf, George K. 1949 Human behavior and the principle of least effort. Addison-Wesley.
3. Johnson, Joseph E.; A Numerical Data Standard Joining Units, Numerical Accuracy and Full Metadata with Numerical Values., European Project Space on Computational Intelligence, Knowledge Discovery and Systems Engineering for Health and Sports, Rome Italy October 2014 ISBN978-989-758-154-0
4. Johnson, Joseph E.; Networks, Markov Lie Monoids, and Generalized Entropy, Computer Networks Security – Third International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, St. Petersburg, Russia, September 2005, Proceedings, 129-135
5. Johnson, Joseph E. and Campbell William; A Mathematical Foundation for Networks with Cluster Identification, KDIR Conference Rome Italy Oct 2014
6. Johnson, Joseph E.; Markov-Type Lie Groups in GL(n,R) J. Math Phys. 26 (2) 252-257 February 1985
7. Johnson, Joseph E.; Methods and Systems for Determining Entropy Metrics for Networks, U.S. Patent # 8271412 awarded.
8. news.mit.edu › 2021 › dna-data-storage-0610
9. [www.genengnews.com/topics/omics/dna/could-we-store-all-of-the-worlds-data-in-a-coffee-mug-full-of-dna/](http://www.genengnews.com/topics/omics/dna/could-we-store-all-of-the-worlds-data-in-a-coffee-mug-full-of-dna/)