Vernam Two Cryptographic Engine

Author: Dan P. Milleville, dmilleville@comcast.net 978-772-2928

This cipher methodology utilizes 4 pointers pseudo-randomly set initially and advanced (different pseudo advancement functions for each pointer, executed between blocks) pointing to number streams, Xor'ed together, that are obtained from a large fixed randomly produced key; they produce **E**ffective **K**ey **S**treams (**EKS**'s) that are Xor'ed with the successive blocks of plaintext. The pointer advancement algorithm does not allow any pointer to be advanced the same amount as any other pointer for any specific block, or the resulting advancement being a duplicate of any other pointer number that was used within 500,000 blocks. Both the sender and receiver can now possess a fixed key and still encrypt and decrypt text using the created-on-the-fly non-repeating pseudo-randomly controlled **EKS**'s formed, just as the Vernam cipher engine does now. Because the **EKS**'s are created from a fixed key, both sender and receiver will possess a common key and pseudo-randomly recreate the same set of non-repeating **EKS** streams.

This design is at least 4 times faster, and also experiences no loss of security, as compared to the AES. The same mathematical operator as the Vernam cipher (Xor) is used between the **EKS** and the plaintext. The **EKS** has multiple key solutions that do result in almost countless key sets that all translate correctly with no methodology available to determine which one of these reconstructed key sets is correct. This is illustrated in Appendix A showing 5 of the almost countless reconstructed incorrect keys. When encrypting 128 characters, the 27,000+ (AES Visual Basic version) mathematical operations executed by the AES are decreased to less than 1,000 for this Vernam Two. This results in an increase in speed that is needed in today's increasingly data-intensive and security-dependent world.

Using a fixed randomly created key stream of 2 Megabytes (2,097,407 numbers), hereafter referred to as the **Key**, four pointers are pseudo-randomly determined referencing this **Key**. The four streams of numbers extracted are equal in length to the block size, 128 key numbers in this design. The fourth pointer is an Xor of the first 3 pointers and two other Engine 2 pointers, 69 and 152, described later. Examples:

Pointer 1 = 453,176 Pointer 2 = 1,150,639 Pointer 3 = 1,873,098 Pointer 4 = 782,464

Xor'ing the numbers from these four streams forms an **EKS** that will eventually be Xor'ed with the plaintext, fully detailed in Appendix B. The example in red is 'DA' Xor '5F' Xor '9A' Xor 'A3' = 'BC'. With the attacker not knowing the 4 key numbers, it is Algebraic law that this single equation with 4 unknowns will remain verifiably unsolvable. Here are 4 example key streams using the above pointer numbers and the resulting **EKS**:

```
Key Stream @453,176    - BF 7B C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D
Key Stream @1,150,639  - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E
Key Stream @1,873,098  - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C
Key Stream @782,464    - D6 CE CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9

Effective Key Stream 1 - 32 21 1C 19 36 54 BC D0 B1 CB 0E DD D3 84 5C 2E 30 52 61 BC 1D E0 FA E6
```

The above, as well as all other demonstration data displays within this document, is at least a partial copy/paste from a demonstration app to illustrate the validity of this design. Two screen shots showing the app displaying the data above is provided in Appendix C. If pointer 1 was advanced to 453,177, the key stream shift would cause the above example to show the **EKS** #2 below. Notice that the 4th pointer is also changed because of the change in the first pointer. **EKS** #1 is copied just below it for comparison:

```
Key Stream @453,177    - 7B C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3
Key Stream @1,150,639  - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E
Key Stream @1,873,098  - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C
Key Stream @782,465    - CE CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02

Effective Key Stream 2 - EE 9B 12 A7 84 20 44 85 C9 CE 93 24 BF 01 DA AF 69 A8 66 E9 84 F5 50 B3
Effective Key Stream 1 - 32 21 1C 19 36 54 BC D0 B1 CB 0E DD D3 84 5C 2E 30 52 61 BC 1D E0 FA E6
```

A very minor change in just one pointer causes a major change in the **EKS** due to the position shift of the key numbers in two of the streams used. Appendix D shows the effect of advancing just the first pointer 10 times.

Advancing the pointers for this Vernam Two technology

Advancing the pointers in a pseudo-random fashion is absolutely crucial to the successful implementation of this Vernam Two algorithm. The function, provided in Appendix E, follows these factors in advancing a pointer:

1. The Xor of all 4 pointers determines the order the pointers will be advanced.
2. The value of the pointer determines the mathematical operation used to determine the amount of the advance.
3. The value of any or all other pointers, before **or after** their advancement, determines the amount to advance.
4. Once the pointer is advanced, that value is checked to not have been used within 500,000 blocks, advancing until a value found to not be used within 500,000 blocks is found.

Also included in Appendix E are six pages of examples displaying how this code advances the pointers.

The 512 random number system key

This demonstration design requires only a 512 byte key, but it is way too small for use during the execution of this algorithm. In order to be used, these 512 numbers have to be constructed a certain way in order for it to be expanded. It is to be constructed in a way so that an additional key can be extracted from the array and used in the key expansion algorithm. In this way, the procedure can be pseudo-random because numbers extracted from this key are obviously classified as 'random' and therefore whatever the process, it is therefore repeatable for any number of cipher applications using the same key.

The numbers created for the key for this system are a set of 512 numbers from 0 to 255, 2 each, randomly placed and determined by the random number generator as you will see. Limitations are placed due to the size limitation (512 bytes) and the need for all numbers between 0 and 255 inclusive are needed within this key. A temporary key is initially made of whatever size is needed until 4 of each numerical value between 0 and 255 has been loaded into a temporary array. Then a loop is started to move randomly selected numbers that don't have 2 of that number in the key from this array to randomly selected positions in the key array. The code used to accomplish this function and produce this list of numbers produced for this technology demonstration system is provided in Appendix F.

Creating the 2 Megabyte Key Stream

Suppose you put a group of randomly selected numbers on blocks, put the blocks in a drum and turn the drum to mix the blocks. You then pour out the blocks and align them. Do the numbers in that order, obviously different from the first arrangement change the numbers into a block of non-random numbers? Of course they don't. So rearranging the numbers using whatever is needed in different ways in a process that can be duplicated (pseudo) would produce a stream of 2,097,407 random numbers (only 256 numbers from the last set is used) from the original. With the input numbers listed in Appendix F, with 2 values of each number between 0 and 255, examining and extracting 256 unique numbers from this stream will provide an adequate Transposition Key. The Transposition Algorithm is **not** used here to encrypt the key but to **rearrange** the numbers in pseudo-random order for repeatability. Note that each set of rearranged numbers has the next Transposition Key extracted rather than use the same transposition key. Note: The transposition key numbers are copied to an area 255 locations above the current one, this use is described later in this document. Using the key in Appendix F, the code specified will input those 512 numbers and expand them using the extracted 8,192 Transposition keys to reposition each block of 512 numbers forming a 2,097,407 randomly formed key stream. All the Transposition keys are Xor'ed to the created stream in order to effectively hide the original 512 byte key.

Testing has shown the resulting number stream created using the number set displayed in Appendix F provided in this fashion produces only 2 streams within the 2 Megabytes of 5 numbers long that are duplicated in two places with no streams equal to or greater than 6 duplicated anywhere. Since it is not possible to reconstruct this key within the framework of the Vernam Two engine algorithm, and the numbers created cannot be discounted as not being random, this methodology of expansion should be acceptable to keep the stored key to only 512 bytes.

The Vernam 2 Algorithm – Transposition Engine 2

In addition to the Vernam Engine described on page 1, the Transposition Keys extracted during the key expansion function are then used to transpose the **nibbles** of the Vernam output and is Engine 2 of this design. This quite literally shreds the Vernam Two output so that there would be no hope of any attack being able to be even started against this system. There is an average of about 16 of each digit in the midstream or Vernam Engine output and the attacker is going to have an impossible time trying to figure out which '8' (for example) in the input would be the first '8' in the line.

Just because the Transposition Algorithm has been broken now for decades does not mean it is useless. The breaking of the algorithm was made possible by plaintext as the input. It is used here to 1) expand the key, 2) uses pseudo-random data as the input and 3) transposes Vernam output **nibbles** instead of the Vernam output bytes.

Take this **midstream** from the demo system and look at the blue highlighted '8' digits, both in this and the ciphertext output below:

A383238AC2D5FD77EB5A62FD228EE1009B95540AC43649574DE7E387167DEB6648182B758B50625
5CACBDF74800FD964A93416A94890ABB7BE95BFF593F8F39B6D533251091752537C5FD80F606E2DB2
2A05CDDADBE80BF9180D582169DF77072AEDDF921430F2492E5FDBC0F20CCA5038C226330A615428D
9F56B8AB352BC18

Here's the ciphertext **output** stream with the '8' digits also highlighted so you can see the transpositions:

803A98B5D22547ABDDC5FC1A55202577A23423F4785CB6D2839F32BDD23F913530CF2F9F037AB94
A7798878 46DA118949E5AB22B093C4B55BE6F978E5888E80ECCA019DB9B1A5E80D23100D35050B12A
8883B5564B25D006B2B6F8AD5050624AC2FD4FDC94EBDF07391E2D63F64CDE6BEF02695723FCF9EDF
16506267757AD16

Take the first digit '8' in the ciphertext line highlighted in red. Since you know the midstream above is defined as pseudo-random, how would an attacker attempt to properly determine which '8' in the midstream was placed in that position 1 in the output? How would an attacker be able to determine that the red '8' in the midstream is the first '8' in the ciphertext line, not any of the other 17 '8's? There are 18 digit '8' numbers, highlighted in blue, in both lines, meaning there are 18 factorial or 6,402,373,705,728,000 possible transposition keys, just for all the '8' digits in the line. There are 15 other digits. The product of all the factorials of the numbers of each numbered digit in this **particular** output line is 8.309 x 10^{215} transposition keys that would take that ciphertext block to produce the correct midstream above. The number of transposition keys for each line varies as the quantity of each hex digit varies in each stream. There is no mathematical process that could ever be developed that would be capable of picking the correct transposition key out of all those possible keys even if there was enough space in the universe to store them. Not only are there more possible keys than there are molecules in the universe, but they all convert the ciphertext properly to the given midstream above.

To complicate the attack even more, even if they were able to correctly reconstruct the transposition key, an offset is also determined into that particular transposition key. Remember in the key expansion algorithm it was noted that the transposition key was also copied to the upper 255 locations? This is why. An offset into this table is randomly determined at the start where the start of access to the table is made, from 0 to 255. Even if the attacker were somehow able to reconstruct the correct key, there can never be any indication by any numeric processing of the exposed key as to where location 1 was located in that key. In the above example, the offset was 209. Locations 209 to 465 would contain the transposition key used to transpose the midstream to form the output ciphertext. There could never be any method of determining that position 48 is actually position 1 of the key used with an offset of 209.

For the next block of ciphertext, the pointers are all advanced using separate pseudo-random technology for each pointer that combines both the values of potentially all the pointers and Vernam key values in determining what is used for pointers for the next block of text. In this way, no additional modes of operation are needed for this design. It also checks to make sure no pointer uses a pointer value used by **any** pointer within 500,000 blocks.

Extensive testing of the advancement algorithm has shown that there is no duplication of all 4 pointers used for any ciphertext block even through 100 million simulated blocks of plaintext. Since the 2 Megabyte key has been shown to be random in nature, Xor'ing 4 different pseudo-randomly selected streams from this key and Xor'ing them together to form an Effective Key Stream (the **EKS** described on the first page), they can be described as non-repeating, the same as what is required by the original Vernam system. Because the key is fixed, formed by a pseudo-random process that is easily repeated by both sender and receiver, the transmission of the key with the ciphertext is no longer required.

The communication protocol of pointers to the decrypt engine

Since 5 of the 6 pointers are randomly selected for each ciphertext processed, these numbers need to be communicated with the decrypt engine using an algorithm to hide them from any attacker.

Starting with the Encrypt engine, the 3 pointers need to be inserted. For the first block only, a simple sum of the ASCII characters in the 120 byte plaintext block is obtained, converted to hexadecimal and inserted as the first 4 characters of the plaintext. After the Vernam Engine has executed, producing a stream of pseudo-random midstream numbers and split into 256 nibbles, the 3 pointers are split as follows:

1. The first 5 nibbles are mathematically combined with the top bit cleared to form a single pointer from 0 to 1,048,575 (1FFFFF hex) inclusive.
2. If the value of the location in the Vernam Key is less than 6 or greater than 235, the pointer is incremented and the next value is obtained, repeating until a number within the acceptable range is attained. Example: The first 5 nibbles: 0, 9, 4, 0, 6, forming 9406 Hex or 37,894 decimal. That location contains 160, an acceptable number.
3. Each of the first 3 pointer's first 16 bits are split into two nibbles each and the 8 nibbles are inserted in the midstream starting at the pointer obtained, in this case, locations 160 through 167.
4. The top 5 bits of all 3 pointers are combined into one number (forming bits 0 to 14 of 15 bits) and then that number is split into 2 nibbles and those are inserted into the midstream as well. In this case, locations 168 and 169.

Example using the sample supplied on the first page – the pointers 1 - 1,725,418, 2 - 1,260,439, 3 - 774,161 were changed to 21,482, 1,551, 44,714, then combines the top 5 bits from each pointer (#1-) 26,624 (6800h), (#2-) 992 (3E0h) and (#3-) 9 (9h) for a value of 27,625 (6BE9h). These 4 numbers are broken into the 8 nibbles and inserted in the midstream for the transposition engine. The Transposition engine executes the repositioning of all the nibbles and pairs of nibbles are combined to form the characters of the ciphertext output stream. The transposition numbers themselves, the key number and offset, are communicated by two extra characters inserted in only the first block of ciphertext, making it 130 characters in length instead of 128. The process involves the following steps:

1. The first three characters of the ciphertext line are mathematically combined to form a pointer into the Vernam key table.
2. A loop executes if the location pointed to in the Engine 1 key contains less than 5 or greater than 125 incrementing the pointer if not, exiting if it is within range. Example: the first 3 characters are 69 (45H), 102 (66H) and 85 (55H) forming 456655H. The upper 4 of the hex is stripped, making the pointer 353,877. That location in the Vernam key contains 193, not within range, it advances the pointer to 353,878 that contains 15, that is in range. The function passes that back to the encrypt engine.
3. It then inserts ASCII characters equal to the 1) Engine 2 pointer and 2) zero-based engine 2 offset in positions 15 and 16. Since these numbers are created randomly and inserted in what is defined as pseudo-random data, there is no way any attacker could be able to consistently determine which two characters identified these values.

Then the ciphertext output line is written to the ciphertext output file. Since the pointer handlings for subsequent ciphertext blocks is handled in a pseudo-random fashion, both encrypt and decrypt engines do stay in lock-step with each other in processing the remainder of the plaintext, so no further insertion of register data is needed (steps 1-4 and 1-3 above).

In the decrypt mode, for the first block only, inputting the first 130 ciphertext characters, the second 3-step algorithm above is executed, extracting the two characters, and bringing the block to a 128-character length. Using the pointer and offset numbers extracted, the transposition operation is reversed, constructing the midstream for the Vernam Engine.

For the first block only, the 4-step algorithm is used to extract the nibbles used to identify the 3 Vernam pointers with the 4th pointer then reconstructed by Xor'ing these pointers plus the Transposition pointer and offset. The Vernam engine is then executed, forming the plaintext string.

To help aid in the decrypt engine determining that it has been successful in decrypting the ciphertext, the simple sum of the ASCII of the characters from character 5 to the end are summed, converted to a hex number and compared to the first 4 characters of the first decrypted plaintext block. If the characters are hexadecimal and they equal the sum the decrypt engine reformed, the process continues. Otherwise the error is discovered and the decrypt process aborts and this finding is reported to the user.

Testing performed to confirm this system's design

To test this system, a process was developed to simulate the encryption using 100 million blocks of text, 12.8 Gigabytes of plaintext. The process created just the Effective Key Streams and recorded the 5 pointer values used to create each stream. This produced a single data file almost 30 Gigabytes in size. Another process inputted this data and sorted it into 256 separate files, one for each value of the starting hex number, '00' to 'FF'. Still another process took each one of these 256 files and separated them into 256 additional files, one for each value of the second hex number, '00' to 'FF', for a total of 65,536 different files, approximately 1,550 Effective Key Streams per file. This entire process was duplicated 9 more times for a grand total of 1 billion Effective Key Streams and the associated 65,536 for each process, a total of 655,360 files containing the sorted Effective Key Streams. All 900 Gigabytes of all files, including the intermediate files used in sorting and test result files, have been moved to an external 3 Terabyte hard drive for any viewing needed. The tests done so far:

1. Each of the 65,536 files in each of the 10 cases was opened and each of the 256 nibbles in each key stream was compared with each nibble in every other key stream in the file and the maximum number of nibble matches per stream was recorded. Results: No more than 55 nibbles compared out of 256 in any two key streams matched.
2. Each of the 65,536 files in each of the 10 cases was opened and all streams in each file were compared, nibble by nibble, with each file in the next corresponding case, #1 compared with #2, #2 compared with #3, …. #10 was compared with #1. This process took 4 computing (24 hour) days to complete for each case. Results: No more than 55 nibbles out of 256 nibbles compared in any two key streams matched.
3. Each of the nibbles in the first case (streams that begin with the first two bytes of '00' and '00') were compared with all other streams in each pass. Results are ongoing at this time (another 4 computing days per pass, 10 passes), but results so far are virtually identical with the tests in the first two cases – no more than 55 nibbles compared in any two key streams matched. So even though only one 2 Megabyte key created these 100 million Effective Key Streams, a grand total of 12.8 Gigabytes of key streams, no stream that began with '0000' would come closer than matching 55 hex nibbles with any other key stream in the pass tested so far.

Because of the number of Effective Key Streams, and considering my limited computational capacity, I could conduct tests for the next 3 years and not be completely done. But results have shown me that this system is solid and considering that even if any set of pointers is duplicated anywhere within a plaintext encryption, the indication of this could never be discovered. And even if it was, the Xor of several key streams more than adequately protects the key and plaintext, not to mention that only less than 0.02% of the key would be exposed. And since all the pointers are advanced between plaintext blocks, the attack would have to start all over again for each block.

Because of the inability to discover the correct values used to create the Effective Key Stream, fixed key overlap is going to eventually occur and will remain undetected if and when it occurs by any external stimulus in either the plaintext or ciphertext.

Appendix A

The black numbers are the real key numbers, the red numbers are bogus. Notice all internal streams and ASCII Hex are all identical.

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @453,176  - BF 7B C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88
Key Stream @1,150,639- 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8 0C
Key Stream @1,873,098- 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49 EB
Key Stream @782,464  - D6 CE CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
33 85 8F ED D2 0E 50 2A 93 21 FE 70 E4 03 A1 9D 60 C6 D5 EA 70 A5 52 A9 F4 8D E5 FE DF 52 25 88 47 37 52 A9 DB
92 67 CC 6A 18 79 7D B1 78 51 2A F0 D3 84 3C 18 EC 23 2F 42 EB B7 E1 25 61 E8 97 06 03 D1 97 F8 7F 7C 82 6E C5
99 7D DC 4E 8F AB A8 3F 9A E8 25 F9 9A 0E FD 95 21 BA FE 99 9A B2 C7 48 3B BA 6A EE 62 74 97 D9 38 AF 68 0B EA
C6 30 22 5A 2F DB 14 6D F3 94 9D 33 CF 27 CB 1F D2 FD 16 4C 7D CE 9A 1E F0 50 24 39 74 30 77 1D 22 20 59 5E 68
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
9D 09 AF 0A B4 C9 84 69 69 45 4D 34 D0 22 38 1A D6 04 42 2E 55 A5 EE 10 5B 02 48 C2 86 C1 2C 06 E3 5A E5 CA 19
CF 36 75 1A 2A EC 1E B5 FB 23 A2 35 D3 51 92 35 D3 02 33 B5 CD 70 4E EE 53 79 5A 66 9A FD 89 8E 0C F6 F6 55 10
A9 7C 96 DC 8A 12 68 12 42 78 83 B6 FB CF 9A D3 0E 8C C6 16 7A 4C 30 0A DB 2E AD 02 46 61 74 8B 5D FA 5E 5A 3E
A7 6E 7A E5 D4 3F 7F 45 A1 36 B0 CA 2F 7F 9D 27 A1 EA F3 48 47 C3 01 0B B9 97 89 E7 82 36 12 92 E6 9F 45 19 81
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
|| || || || || || || || || || || || || || || || || || || || || || || ||
9D 95 E0 72 6D 5B 87 9E E1 D4 27 BD F6 67 A7 D3 B9 BF 75 9A BD D1 E1 6A
F1 F0 10 F4 04 15 2E 32 B1 E8 E0 C9 9A BE BD 23 0E 35 8C F7 BD 19 04 39
34 DE F7 3D F8 59 FF A0 6F CA 10 3D 23 12 40 B9 B1 24 38 9A 63 82 D9 80
90 70 4C FE D8 A5 60 26 92 F1 A9 17 26 E0 2F E5 AA 9B 6B 56 29 66 E7 A7
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D
```

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @?,???,???- 93 4A EC FD AF B9 14 BB 3E E4 1D 83 CA EF D9 47 5F F4 8E CC 3D D9 BF 31 C1 6E 3B 9F 90 CA
Key Stream @?,???,???- 42 20 F4 B1 C0 F5 1B 0A 09 B1 FB 72 33 52 4F 0B C4 01 77 6E AF 79 16 36 8D BA EF CF 1F EE
Key Stream @?,???,???- 90 BF 21 6A 60 9A 92 6E 60 73 C0 B0 3F 02 ED FC 5B 7E 65 B8 CA 61 B0 41 AE 61 F9 83 B9 AA
Key Stream @?,???,???- 73 F4 25 3F 39 82 21 0F E6 ED 28 9C 15 3B 27 9E F0 D9 FD A6 45 21 E3 A0 16 D7 F1 4F 1A 0A
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
6B 51 06 E6 D8 C7 DE 7A FA 46 21 17 48 93 53 85 7C EF DC 5E 25 55 B3 EB 72 F2 51 3E 3D E5 D5 90 56 37 73 DC 11
FC 80 04 9B D0 B7 B6 0A 2E 93 CA F7 E9 B7 68 79 FD 4C 08 32 EE 57 6E BB 61 3D E7 49 F8 93 C4 A4 72 2B 17 4A 02
BC DB FE 60 B6 30 F9 53 09 4F E1 04 AE 87 2D D8 20 85 64 9A 46 71 29 37 A0 48 85 B4 B5 7C 76 72 BB C9 B8 A6 88
D5 A5 41 8E D4 47 00 EA 5F 96 66 AE 6D 0D BD 2B DE 84 A2 8B F1 1D 1A BD ED 08 0F EC BA CD 35 F2 BD 11 3D A2 07
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
1C 21 BB 11 6C 84 82 56 5C 18 99 85 80 9F 1E 76 05 CB BA 03 2A 2E C1 0E FF 60 90 D3 61 72 23 70 35 93 F5 B1 E3
5B AC DE EE 8E 4A D6 6B 10 2B 64 B8 16 B8 A4 79 0C 89 06 55 22 9B FE D2 18 D3 EA 6F 49 B3 43 02 F6 5A 08 54 D3
59 D3 12 B8 1B 65 08 1F 60 B0 A1 70 B6 66 38 2E 09 B3 2D D3 F7 8B D7 3F 72 35 CA 63 9C 18 4A 2D DE C4 0E B6 4A
42 73 41 6E 39 A3 D1 A9 5D AB 80 30 F7 82 2F FA AA 91 D5 40 5A 64 79 1C FF 44 86 9E 6C B2 E9 CE 49 C4 FB 8F CC
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
|| || || || || || || || || || || || || || || || || || || || || || || ||
00 1C 20 15 30 A4 4C B1 40 2B 59 13 9C BD C8 8F EE F1 87 26 4B 43 3B 87
C1 66 FA D7 34 AB 76 3F 0E 6C 8E E3 B8 F4 EA BD E3 F0 F4 B6 04 59 EC B3
43 39 A9 32 A1 95 F6 09 B9 11 B7 78 41 7C 62 C4 18 83 30 5F CE E8 CC CB
4A 88 38 B5 EC 28 FA AD 5A 51 1E D6 0C 1E 35 5A B9 B7 E9 6E CB DE C0 8B
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D
```

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @?,???,???- 8E E1 55 A6 80 12 46 62 E3 85 D9 97 4B FB 28 20 E2 26 DF D3 EA 98 52 96 73 E9 DC A2 58 2A
Key Stream @?,???,???- 45 AF 52 70 90 E0 C9 89 B9 24 E6 C0 F5 E4 C5 BC B2 B8 D6 23 20 9D 8D 7B FD 78 48 43 A4 CB
Key Stream @?,???,???- DA 0A F6 71 57 E3 6F 07 A6 05 5A 97 58 22 15 A4 55 66 FD A8 3E C5 6A 83 05 87 F0 26 C9 10
Key Stream @?,???,???- 23 65 ED BE 71 45 5C 3C 4D 6F 6B 1D 35 B9 A4 16 35 AA 95 E4 E9 20 4F 88 7F 74 B8 5B 19 75
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
79 15 7E 8A A6 BE 50 46 78 24 75 D5 FB D7 D4 54 CD FB C9 2F FA 11 A8 10 19 07 72 0C A7 82 D6 0B A2 6A 6A 78 69
80 52 58 B5 34 6E CD 2F 91 30 F7 84 E5 16 16 12 C5 E9 02 3F 7C 51 B3 BD B7 B5 13 01 6F 58 06 00 16 B1 BA 2B 04
D3 7F F6 F3 CA A7 9D 3A F7 49 11 EE 26 55 C2 ED 45 E5 5D ED 24 6F 23 CA 6E 2C 89 17 E6 C7 11 E4 E5 49 67 CC 13
D4 97 6D 5F 32 70 91 9A 9C 51 FF F5 5A 3A AB A4 32 55 84 80 DE 41 D6 BD 9E 11 D4 35 E4 DA 93 5B 73 56 56 0D E2
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 29 2A D2 7A 30 91 8D 2E B4 24 B8 3F 47 09 C2 DB 4B 05 84 F8 B6 A6 9C A5 07 AF 0C 95 88 C4 14 E8 C5 56 4C 3C
0A 2C AD 23 5B FA E3 D8 68 6F 1C 35 18 CC 92 5D 18 D3 A7 36 B0 99 64 55 E4 8C 2E 26 CE C2 C9 5A 3B 84 92 96 83
44 BE 43 64 3E 6D 1D 55 60 C4 61 68 5F 34 4E DD A9 9E AE B1 D3 FB 5A DB 42 2C 0D C5 18 23 6D 09 61 32 57 6C 7D
A5 96 F2 BC DF AF E2 8B 57 37 85 98 AF 7C 78 99 C0 66 48 C6 3E 8E 09 ED 69 65 BA AE 9B 02 A3 D6 E6 BA 9B 6A 74
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
|| || || || || || || || || || || || || || || || || || || || || || || ||
4C D0 04 F3 A5 01 E0 C6 94 CC AE 8A 8D 0F 8A 40 77 D0 34 63 A5 AD B0 6D
1A F8 58 AD D8 3C 2C C2 69 69 0F 71 5A 95 23 E3 E9 59 94 E0 0E 8D CA 23
B2 6A 1F D0 D8 D5 34 A8 D3 64 5E 6E 29 C8 ED 10 52 27 08 AE 95 E3 2C 83
2C 89 08 CB EC 5A CE 86 83 C6 81 CB 97 79 31 1F 60 9B 02 8C 74 EF 8D B9
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D
```

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @?,???,???- EE 15 1E 44 20 C6 05 B7 A8 97 7E 77 6F FD 85 C4 5A EF 8B 18 94 DE E8 BC 1D F3 86 A8 93 8C
Key Stream @?,???,???- 7B 22 7A 13 41 98 CC 7C 82 7D 58 28 B3 61 E4 C2 56 55 C6 54 17 F4 A7 6D 2A B1 7B 96 E3 63
Key Stream @?,???,???- 68 6F 9C 7F C2 37 59 C1 32 41 86 BE CD 0A C9 24 7B 90 08 8C A7 32 3B EE 28 C7 75 68 6E FC
Key Stream @?,???,???- CF 79 E4 31 95 3D 2C DA A9 60 AE 3C C2 12 F4 0C 47 78 24 7C 39 F8 8E D9 EB E7 54 CA 32 97
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
CA A0 3F F8 C6 8C 62 71 9D 16 C1 59 36 89 6E B8 23 E9 D8 F6 D3 4E 15 D4 F6 85 6E B0 31 E9 2B F9 02 7D A0 F0 F5
BF F7 96 0D F5 45 B0 F2 5C EE 16 D8 F4 87 BD 9C B1 75 8A F6 FE DA 6B 8D 74 C2 B6 3D DB 8D C0 A7 61 9F C1 FA 19
30 2F F0 F2 D9 D5 D4 17 9B BC DF 44 6F DA 9F 18 64 AE D7 93 2A 82 2B F5 E5 E2 0C D8 2F 5B E5 8C 6E 78 86 B2 57
BB D7 E4 94 80 1B 97 5D D8 48 64 8F CF 7A E7 33 89 90 97 EE 7B 78 BB 76 39 2A E8 7A 0F F8 5C 66 2F 5E 06 2A 27
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
4E 77 A0 08 07 3F 1F 5A D5 35 05 1A 7D 52 2D AF BA 94 AB 23 DB 90 0C D5 9D 38 D4 5E 46 E8 34 C3 02 B7 0B DC 81
46 E7 00 B6 EB B8 6B 88 7C AC 60 B6 19 0A 44 BC FD 03 C6 28 AE 0C 2C 90 67 7F E1 D6 C5 75 3F 9F B0 DD 99 10 8C
FA CF 1B C2 56 A1 E2 B0 73 01 C8 99 A4 8D CF FB E3 24 19 1F 50 D0 01 2D E8 F7 7A 72 89 C2 A8 F0 26 CC BC 26 46
AE 72 8D 55 7A 2E 1B E9 AB B0 71 48 17 16 0B 33 0E D3 30 D1 80 16 B0 97 78 72 79 BB D2 34 60 3D C0 6F 26 36 FD
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
|| || || || || || || || || || || || || || || || || || || || || || || ||
FC 20 04 D1 1D 9D 02 78 9F CD 36 D5 6A 19 06 B8 5C 40 8B 3E 11 D5 77 F2
E1 8C CB A0 79 86 29 C6 4F 96 B9 54 D6 76 F8 49 A5 D6 51 BB 18 EA 56 7F
0E 14 21 C2 0A 92 C2 BE 36 3C 51 E8 AA 15 42 EF 0D 3A 46 46 BC 3F 95 1A
DB 73 A5 F6 27 3B DF 2A 4B 60 A0 37 7F 51 C9 B2 58 99 36 62 FF 2C 6F E3
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D
```

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @?,???,???- 83 5C 5C 09 3A 38 56 DA 79 BE E3 CD E0 3D D7 33 B1 CE E6 9A 9B CF 91 EE CE 9C 53 CD 3B 90
Key Stream @?,???,???- 2A 59 14 73 17 1A 72 5F 87 CF AA 2F CF 79 F3 5F 57 BC E0 B3 3D 73 9E DC 2A 74 6C 6C 72 A5
Key Stream @?,???,???- 96 27 55 D6 59 A8 1A B5 3A 75 93 0C AF C3 0A 15 EA D1 B4 3C 07 43 FF 15 77 C0 1B A1 AB CE
Key Stream @?,???,???- 0D 03 01 B5 42 DE 82 E0 75 CF D4 33 53 03 72 57 3C F1 D3 A9 BC 1F 0A C1 67 4A F8 9C CE 7F
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
0B 29 2A 3B AC 24 01 E4 23 27 53 41 89 44 27 75 89 CD 89 4A 13 34 56 AF 41 25 44 74 89 E2 2F 43 24 DD AB E2 BD
5A FB 72 3F 45 9A 26 60 12 78 09 D7 29 D2 31 2B 35 1F AA EC 19 7C 12 3E 35 55 75 A2 DA 54 6A 06 09 8E 6D 0F 73
C6 C1 04 EC 5C FA FB F0 1D 6B 8C A3 C0 0B E7 9A 38 26 59 57 C6 19 77 9F 3F 4C 8D 1D 9E 70 45 38 CC FA 9B 31 AE
69 BC E1 7B DF 43 4D BD AE 38 BA 7F 02 33 5A CB FB 56 68 8C B0 3F DD D4 15 B3 80 E4 07 01 52 C9 C3 6D BC 4E FC
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
9B C5 A6 5E C7 1C 3A 96 95 20 92 EE 59 A8 29 8E 67 BA 81 2B 2C 96 DC 33 8D 03 37 3B 7F DD 87 A8 E7 DF 37 9C B3
EC 7F A5 AD D6 FB C0 B3 26 BB 71 9F 4B 6B 33 AA BC 2C 72 63 9F 6C 5C 2D B9 04 E7 FE 99 93 22 55 FF 19 C0 7C 92
DF 00 7F 6F 91 CC 6E 0F 39 E9 F6 2B 35 BD D8 0F D4 57 FF F9 50 DD 15 AE 0B 35 63 42 0D F1 ED 23 BD 80 CD 67 DF
F4 97 4A B5 40 23 19 A1 FB 5A C9 27 F0 BD 6F F0 A5 A1 48 74 46 7D 04 4F 55 F0 85 C6 33 D4 8B 4F F1 8F 32 5B 48
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
|| || || || || || || || || || || || || || || || || || || || || || || ||
AC E0 23 39 24 4A 9E CF E5 A3 6E AA 1E FE E4 FB 9B 0E 68 47 0D 22 E9 07
D6 E1 FF C2 65 23 6B FA 13 E8 4B 0A 84 E9 5F D2 89 FB 1F 33 3B 20 6F 32
92 CF 57 CE CC 4E 3B FA AC 4D 67 A3 BE D9 86 D7 00 9D 25 A9 91 C6 82 B2
20 05 C0 70 C4 95 F8 E5 F7 01 3C 5D 4D E5 48 52 BE 5D F8 7C ED E8 DF F3
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D
```

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @?,???,???- 59 37 1B 89 59 B3 0D C6 C7 AE 6A 20 24 D0 7D BC C8 AC 70 35 C4 E2 58 55 14 76 83 12 09 70
Key Stream @?,???,???- 8F 1E AC 0F DB A1 69 12 37 AB F3 60 C6 BB 98 77 38 84 0A 79 07 9A 38 F8 EF 5D 6C 00 AE 2A
Key Stream @?,???,???- 95 2F EB 0B 83 8E BB 95 45 3A FE 5C 5E 69 36 14 AE B4 C2 86 B9 0D 42 30 3A 00 4B 1B E7 2D
Key Stream @?,???,???- 71 27 40 94 37 C8 63 91 04 F4 69 C1 6F 86 8F F1 6E CE D9 76 67 95 D8 7B 35 49 78 95 6C F3
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
00 CB F4 B4 6F C3 16 23 0B DD C7 F1 F4 99 E5 49 C1 4E 80 1E 45 A9 C2 53 A0 65 23 0E 89 4C CE 59 DA 6D 4D F9 5C
A5 64 FD C0 36 4E CE F0 9F 01 C6 88 AA F0 FD 80 CA 04 F4 65 AF 29 D6 06 2D 19 F3 5D 4E 6D 79 C0 B5 E0 73 48 35
4B FD 85 04 BE 8F 2F C3 91 2F E3 AD D6 B9 4D 8C A9 A1 D7 68 61 54 F0 88 FB D0 9B A9 C2 82 3B 7D 43 5F 9D 67 70
10 FD 31 E3 8D 05 66 D9 87 FF 8E 9E EA 7E FE 4A DD 49 B1 6E F7 BA 0A 07 28 23 77 D5 CF 64 DE 50 0E 16 42 44 85
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
9F 7C 92 D4 35 AB FC 4D 02 15 DF A7 8F 80 95 5F 9B 35 40 A0 D3 22 29 8A D2 4D 5B C2 5B 7C 7F 3A D5 21 FA 40 06
E5 07 80 B1 6C EB 51 36 F0 29 2C 56 C4 13 79 C6 9E 88 40 3B 64 68 16 80 36 54 49 DB BB AA 4E A2 AF C0 69 18 AC
45 47 74 12 93 FF 9B 73 06 9D 38 4C D6 14 C6 EC 10 BE DC A9 36 E7 6A 8B 14 9C 34 37 E5 B5 8E 19 A6 21 59 D4 B8
63 11 50 5E 0A B7 BB 83 85 89 17 C0 4A 44 87 AE BF 63 98 F7 24 F7 C4 7E 9A 47 10 6F DD 08 7C 10 88 09 C2 50 A4
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
|| || || || || || || || || || || || || || || || || || || || || || || ||
8B 44 CC 13 4A 8B 06 B8 E2 88 16 C3 B4 A0 30 D6 28 3E DE 5C 1F 6E 8F CC
5F 9E 8D 38 82 D8 B5 0F 4F 58 47 B2 55 5F D9 2F 07 1E FA 32 78 DD 90 4C
70 3D EE BC A5 27 B7 AA 3D 05 7A 33 7A C0 E6 59 F5 7A BE 4C 62 09 BF 86
6C 2C E4 D2 24 C6 32 37 3D D2 55 1C F2 14 7A 0C 76 6F 30 83 4F 96 7B 72
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D
```

This shows the Transposition Engine #2 operating on the Internal Data Stream. The numbers above the Internal Stream are the position number in vertical format. The numbers above the Ciphertext Output are the position in the Midstream where that hex digit was obtained.

On the next 5 pages are random reconstructions of Transposition Keys that will reconstruct this midstream into the ciphertext in this example. The Internal Data Streams and ciphertext are all identical in this and all 5 bogus examples.

```
                                 1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56
                     12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

   Transposition     - 11 11 11    11 11 2     2     12 21 11    11 1  11        1 11 12  1 2  2  2  21     2
  Key Number 69      - 83 52 70 4  78 34 24 9  04 36 91 43 54 41 51 49 34 89 57 80 51 20 55 59 36 47 59 58 54 82
  Offset Number 152  - 76 83 62 44 03 71 00 59 91 46 70 64 06 30 93 36 98 10 04 29 62 27 63 37 55 07 41 88 53 94
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
  Ciphertext Output  - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD
```

```
                                                                 1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00 01 11 11 11 11 12 22 22 22 22 23 33 33
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

1  2  21 11  1 2     11  1 11 1  12  1 1  1  2  12     2    11    2  21 1   1 2  11 2   2  2  1 11  1 22 11 1
6  41 12 19 80 37 53 38 96 92 98 74 74 93 75 31 13 48 20  3 47 25 35 35 27 69 29 68 41 74 13 45 02 67 25 60 93
65 59 77 69 35 46 98 28 33 25 40 42 15 51 84 71 48 56 92 33 05 62 21 07 89 33 12 85 73 58 26 92 86 11 92 13 09
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 CA 2A 11 38 FC F5 A2 5C B6 09 BF 77 85 A3 03 D7 D6 7C 45 9C FF E0 91 D6 C2 5C F9 DC A4 D4 92 A1 82 05 A0 09
```

```
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 22 22 22 22
33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00
56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

2  21 11 1  1      2 12 12  1 2  1  12  1  1 12 22 21 11 1  22 21  1 21 11 21 1   2  1    22       1  2  2
57 17 82 49 71 76  3 91 12 67 02 42 65 48 36 12 10 07 53 85 41 20 62 20 81 13 32 64 10 28 10 45  4 13 34 23 11
02 37 29 21 96 07 63 86 96 22 43 45 01 80 69 08 96 33 45 63 95 51 81 47 18 18 30 01 84 77 48 67 87 52 12 70 47
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
83 27 19 C7 C0 AF 50 37 A1 CD 8A 4B 84 24 A1 96 DD 2A F2 6B 33 C0 63 AF BE AD FA 0D 26 EC 9A A5 A8 4A 79 6C FC
```

```
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
01 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55
90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D

11  2 12  2  1  1 11 2  1  11  2  1  1 11 22 1      1 2  2   2  2 11  1
83 23 95 80 24 30 40 57 63 26 92 11  5 61 20 86    93 17 12 94 80 52 66
90 19 66 51 87 56 90 58 27 47 83 51 11 57 25 49 72 91 83 22 44 40 50 44
|| || || || || || || || || || || || || || || || || || || || || || || ||
01 68 2D 18 1F 2B FD 01 71 4D 40 FD 60 FA 59 B4 66 5F 8F E3 F0 98 F8 E8
```

```
                                  1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56
                     12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

   Transposition     - 1  2   2 21  1 12     1 11 2  21 11 22 1  11 12 21 11  1 2   2 11 11 21  1 2     1   2 12
    Key Number ???   - 02 58 55 11 18 92 86 49 78 4  12 28 53 92 64 84 14 19  2 49 85 66 80 17  5 27 53 59 55 70
  Offset Number ???  - 47 50 60 00 18 41 83 09 87 34 92 35 49 51 61 47 68 52 99 57 76 57 31 17 36 57 29 40 82 27
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
  Ciphertext Output  - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD

                                                    1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00 01 11 11 11 11 12 22 22 22 22 23 33 33
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

 2    11     1 12    21 11 1  2  22  2 12 11 1  11       2  1  2   1  2 2  21     2 21    12 11 21 2       2 1
23 63 96 82 74 44 93 53 14 07 34 23 73 02 17 58 14 21 91 55 99 41 65 72 08 09 36 63 47 79 13 37 21 14 5  44 06
51 26 31 58 26 90 43 35 32 69 32 04 18 22 70 33 45 40 17 17 15 24 41 46 66 26 85 70 15 08 26 46 46 88 16 64 36
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 CA 2A 11 38 FC F5 A2 5C B6 09 BF 77 85 A3 03 D7 D6 7C 45 9C FF E0 91 D6 C2 5C F9 DC A4 D4 92 A1 82 05 A0 09

11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 22 22 22 22
33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00
56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

 2 21 1        2   1  1 11  1 2     11  2    2   1  1 11  1  1 1  22 1  12    1  2  21 2  12    12 1  12 1  11
44 00 68 32 81 07 98 23 38 70 02 65 68 14 17 0  91 30 07  4 59 66 21 25 51  1 53 34 18 31 40 23 10 27 61 29 57
39 39 94 09 26 83 39 29 22 60 03 93 40 88 98 57 61 58 71 23 48 30 85 79 92 82 02 75 36 23 09 01 91 45 24 52 59
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
83 27 19 C7 C0 AF 50 37 A1 CD 8A 4B 84 24 A1 96 DD 2A F2 6B 33 C0 63 AF BE AD FA 0D 26 EC 9A A5 A8 4A 79 6C FC

22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
01 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55
90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D

1  21 2  12    11 11 11  1  1 22  1  2 11    11     1  1 11  1 2   1 11
35 20 18 30 31 57 33 52  5 82 32 59 62 47 94 84  4 33 46 32 29 46 42 16
70 75 79 04 75 24 18 86 57 18 53 57 89 73 99 14 11 43 78 61 60 61 40 80
|| || || || || || || || || || || || || || || || || || || || || || || ||
01 68 2D 18 1F 2B FD 01 71 4D 40 FD 60 FA 59 B4 66 5F 8F E3 F0 98 F8 E8
```

```
                                     1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56
                       12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

  Transposition      - 1  2  11 1   2 21 1     12 22 21     1 1  12 11 2  11 1     21 1     2  2   2  1 1     11
   Key Number ???    - 26 4  92 46 21 24 73 97 22 24 00 44 96 4  94 82 36 80 27 69 47 69 31 5  58 13 71 38 45 10
 Offset Number ???   - 54 44 30 76 27 91 40 28 77 16 67 92 40 62 23 18 11 06 64 21 02 86 56 35 17 74 70 18 47 20
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
 Ciphertext Output   - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD

                                                          1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00 01 11 11 11 11 12 22 22 22 22 23 33 33
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

11  2 1      2 12 12 11  1 1  11 1  21 2   1 12 11 11 21 1  21     2 11 22  1 1  12 1   1    1     2  22  2 11
80 40 97 8  85 54 52 33 37 61 94 42 37 03 07 31 93 18 17 49 14 11 35 28 12 18 17 40 66 41 86 81 73 15 35  2 52
49 08 65 59 30 45 52 25 45 60 00 36 87 48 82 75 79 47 69 49 42 45 21 32 98 33 36 95 75 65 99 58 07 86 32 83 39
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 CA 2A 11 38 FC F5 A2 5C B6 09 BF 77 85 A3 03 D7 D6 7C 45 9C FF E0 91 D6 C2 5C F9 DC A4 D4 92 A1 82 05 A0 09

11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 22 22 22 22
33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00
56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

 1 2   1    12     1 11 11 2  11  1 2  12 21     2  1  21  2  2  1  1  1  2 22 1  12 2  12 22 1   1  1     1 2
49 19 29 62 53 27 90 26 76 21 61 85 09 83 13 88 42 73 47 72 14 88 47 16 21 05 22 00 0  30 32 73 39  6 78  6 19
38 37 81 39 67 33 33 12 89 52 49 19 08 86 10 46 14 66 21 90 19 29 10 95 52 96 20 17 37 62 04 31 25 31 10 13 05
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
83 27 19 C7 C0 AF 50 37 A1 CD 8A 4B 84 24 A1 96 DD 2A F2 6B 33 C0 63 AF BE AD FA 0D 26 EC 9A A5 A8 4A 79 6C FC

22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
01 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55
90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D

21 12 1  12 1     11  2 1  12 21  1 11 21  1  2       11 1   1     1  1
51 03 54 90 55 45 33 62 45 24 39 51 85 51 33 94 26  5 05 15 65 34 54 20
56 49 25 91 79 83 38 06 50 47 54 21 61 47 34 08 18 65 50 84 78 97 88 72
|| || || || || || || || || || || || || || || || || || || || || || || ||
01 68 2D 18 1F 2B FD 01 71 4D 40 FD 60 FA 59 B4 66 5F 8F E3 F0 98 F8 E8
```

```
                                 1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56
                     12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

   Transposition     -  1 2  21     1 11 11  1     1 11    22    12  1  2  1  2  2 1   2  1 21 2   1  1 22    2
    Key Number ???   - 43 46 10 54 77 01 87 33 36 33 35 83 53 61 94 81  0 80 53 11 74 45 59 17 49 66 59 12 15 48
  Offset Number ???  - 16 46 75 99 26 33 45 00 68 44 84 49 49 10 23 82 51 16 00 76 95 46 64 17 82 58 21 00 57 79
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
  Ciphertext Output  - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD

                                                                 1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00 01 11 11 11 11 12 22 22 22 22 23 33 33
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

1   2  2  1 22     2  1 22 1  21     1 11  1 11  1 21  2     1    21 11 1      2  2    21 21 11  1 11 1   2  2
87 60 15 31 10 68 52 28 52 6  32 55 26 94 17 39 90 18 90 69 76 79 19 45 2  14 34 14 29 02 08 89 36 28 59 42 61
11 29 83 76 50 72 52 38 25 67 73 38 92 51 90 78 69 96 12 93 43 34 20 07 82 38 10 46 45 84 70 56 29 03 19 69 04
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 CA 2A 11 38 FC F5 A2 5C B6 09 BF 77 85 A3 03 D7 D6 7C 45 9C FF E0 91 D6 C2 5C F9 DC A4 D4 92 A1 82 05 A0 09

11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 22 22 22 22
33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00
56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

   22     2 12  1  1 21    21 21  1 12 1   1 11 21 21 1  11     1  1 12    1  1  11 1  1   1 1  11 11  2     1
42 03  4 43 52 22  0 44 77 40 16 95 63 3  72 22 49 13 63 84 81 65  2 73 22 11 4  57 52 17 87 23 04 11 90 88 74
32 31 92 08 63 02 61 95 08 50 81 89 46 53 56 95 17 32 55 73 31 33 11 84 57 92 78 82 21 86 03 78 88 57 75 67 72
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
83 27 19 C7 C0 AF 50 37 A1 CD 8A 4B 84 24 A1 96 DD 2A F2 6B 33 C0 63 AF BE AD FA 0D 26 EC 9A A5 A8 4A 79 6C FC

22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
01 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55
90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
|| || || || || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D

1  11 12 2  11 1   1 2  11 21 12 11 2  12 21 12 2   1 21    2   1 11 22
88 04 70 24 83 99 21 32 39 36 45 51 21 52 21 75 27 34 00 65 45  0 36 35
95 46 16 67 21 30 61 38 99 57 45 04 86 54 10 41 79 39 47 44 21 42 30 20
|| || || || || || || || || || || || || || || || || || || || || || || ||
01 68 2D 18 1F 2B FD 01 71 4D 40 FD 60 FA 59 B4 66 5F 8F E3 F0 98 F8 E8
```

```
                                  1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56
                     12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

   Transposition     -  1 21 22  1 12 2  1  21 22    22 21 22 11 1  21 22  1 1   1 1  11       1  1      1 12  2
    Key Number ???   -  1 22 05 49 21 33 06 41 02 3  13 01 10 02 49 27 33 66 26 73 58 40 36 79 81 67 58 15 52 24
  Offset Number ???  - 78 99 30 41 17 34 63 56 88 34 94 50 04 25 39 02 89 96 66 69 69 90 50 51 07 37 94 49 42 41
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
  Ciphertext Output  - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD

                                                    1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00 01 11 11 11 11 12 22 22 22 22 23 33 33
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

12  2 1   1 22  2 1   1 11        1 11 21 11 1  1   1 1        21 21        1       12    11 1  1     12  1 11 1
83 62 91 23 41 52 4  33 44 27 14 92 46 01 19 38 27 18 04  3 33 50 10 43 48 18 36 57 14 78 31 31 25 44 92 29 52
41 54 69 80 98 85 76 25 12 59 69 02 52 13 98 73 81 26 90 38 90 47 21 27 56 38 12 54 10 01 85 48 70 38 41 00 33
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 CA 2A 11 38 FC F5 A2 5C B6 09 BF 77 85 A3 03 D7 D6 7C 45 9C FF E0 91 D6 C2 5C F9 DC A4 D4 92 A1 82 05 A0 09

11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 22 22 22 22
33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00
56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

1  11 12    2  12 22  2 2  12 1     21 22 1  11 21 11    11  1  2    11 1  22 11 11 12 1  22 1   1 2   1  1
47 57 61 8  05 74 45 21 5  75 9  95 02 15 37 40 41 70 15 87 57 82  1 16 86 10 36 59 82 39 30 75 36 32 28 67 69
82 27 94 75 21 82 35 26 39 96 58 83 04 31 28 04 74 68 56 74 40 23 11 75 14 17 31 87 37 62 09 37 64 53 95 85 75
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
83 27 19 C7 C0 AF 50 37 A1 CD 8A 4B 84 24 A1 96 DD 2A F2 6B 33 C0 63 AF BE AD FA 0D 26 EC 9A A5 A8 4A 79 6C FC

22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
01 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55
90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D

11  1 12 1  1  1   1 22    21 11 1   2    2  12     1 1  22 11 21 11
89 10 70 84 52 98 71 32 98 36 40 39 24 94 58 94 4  96 65 31 55 42 54 26
99 05 16 27 76 38 32 76 75 67 43 16 14 46 20 28 12 38 02 25 51 60 06 71
|| || || || || || || || || || || || || || || || || || || || || || || ||
01 68 2D 18 1F 2B FD 01 71 4D 40 FD 60 FA 59 B4 66 5F 8F E3 F0 98 F8 E8
```

```
                                  1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56
                     12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0

   Transposition     -  2 21  2    21 12 12  1    21  1  1 12 11 1  11 11 2     11 11 11  2    12     2    11  2
    Key Number ???   - 11 54 31 73 49 82 64 65 37 21 26 48 33 08 83 71 64 39 77 60 79 50 13 29 44 19 11 55 21 85
  Offset Number ???  - 02 50 58 39 96 91 60 27 29 20 48 25 39 56 13 41 28 50 84 39 97 40 87 31 45 74 54 93 23 96
                     || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
  Ciphertext Output  - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD

                                                    1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00 01 11 11 11 11 12 22 22 22 22 23 33 33
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
8C CE C9 F6 4A 73 F9 AC F1 69 4C 39 16 CB DB 7C 5F CD 74 5D 08 06 8B FA 79 DD 5D 41 AE A8 3F 94 61 AD 91 FA F9

 1  2 11     2 2  1       12 2  11  2    2      2 1      1 22 2  22 1  11  1 22    1  11 2  12 2  12 1  1  2
83 85 91 32 80 51 35  5 36 92 28 85 93 69 05 52 11 7   9 64 40 14 34 9  32 98 52 1  16 11 1  30 08 41 93 31 34
89 23 37 78 30 43 17 86 43 27 30 40 71 19 84 12 26 27 55 91 62 04 24 19 85 23 25 44 45 95 93 43 95 63 01 26 39
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
B7 CA 2A 11 38 FC F5 A2 5C B6 09 BF 77 85 A3 03 D7 D6 7C 45 9C FF E0 91 D6 C2 5C F9 DC A4 D4 92 A1 82 05 A0 09

11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 22 22 22 22
33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 67 77 77 77 77 78 88 88 88 88 89 99 99 99 99 90 00 00 00 00
56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
2E 0D 79 5C B4 78 F8 FF 02 0F FC 10 B8 A7 C8 FB DF 13 2D AB C2 7A C4 B1 2B 96 62 00 9B 20 82 D3 18 8C 28 9D DA

11 1     1  11  1 21 22  1 11 2  11 22 1  11 21 22 12 12  2 11  2    1  11  2 2   2        1 1  12 2  12    1
67 77 56 42 53 36 45 13 11 71 04 85 04 38 23 38 44 52 41 82 29 82 21 75 41 70 42 60 4  63 87 63 05 37 70  7 04
00 11 06 29 67 65 31 58 96 52 46 09 18 51 70 07 17 24 77 60 18 79 11 82 38 57 20 06 82 40 43 18 80 60 75 16 70
|| || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
83 27 19 C7 C0 AF 50 37 A1 CD 8A 4B 84 24 A1 96 DD 2A F2 6B 33 C0 63 AF BE AD FA 0D 26 EC 9A A5 A8 4A 79 6C FC

22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
01 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55
90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56
|| || || || || || || || || || || || || || || || || || || || || || || ||
AF AE 29 37 28 DB 55 0A C1 66 09 7E 0F 44 07 8C DF 50 C9 D4 38 45 AF 0D

11  1 11 11 12 1  11 12 11 2  11     1  2  1 1  21     1 1   1 11     1
02 60 86 92 63 72 52 52 48 54 29 59 45 51  2 09 20 97 44 37 60 29 24 26
36 82 87 90 94 65 58 36 52 15 44 86 18 51 69 68 84 37 39 62 71 35 67 74
|| || || || || || || || || || || || || || || || || || || || || || || ||
01 68 2D 18 1F 2B FD 01 71 4D 40 FD 60 FA 59 B4 66 5F 8F E3 F0 98 F8 E8
```

Appendix B

Illustrated below is a full view of the entire 128 character block processing executed by the Vernam Two Engine illustrated on the first page of this document. The fourth key stream pointer is an Xor of the five values of the first 3 pointers and the pointer and offset for the Transposition Engine displayed on the next page (453,176 Xor 1,150,639 Xor 1,873,098 Xor 69 Xor 152 = 782,464). The example text in red illustrates the sequence: 73 Xor B9 Xor 9D Xor 85 Xor 7C = AE. The top plaintext hex is Xor'ed with the 4 key values below them, resulting in the hex number in the Internal Data Stream line. This entire example, here and on the next page, is copy/pasted from a technology demonstration application, illustrated in Appendix C later, so all the calculations are guaranteed to be correct.

Sample text block used:

```
This is sample text to demonstrate these steps of the 'Random Cipher Outputs' mode using
UNATTACKABLE Algebraic law for security
```

Vernam Two processing data:

```
Plaintext ASCII Hex  - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Key Stream @453,176  - B2 2C 8D 67 D2 23 B9 34 0F 56 10 A0 D2 24 AC 4F 0B C5 28 96 3F 5F C8 1E 9F 91 66 B7 16 03
Key Stream @1,150,639- F6 F2 B4 F5 57 83 9D 57 8E 17 D8 2C 07 51 8F F7 B1 DE 38 F9 84 80 B5 AA A1 5C F8 FF 3C 6A
Key Stream @1,873,098- 88 D7 E8 6E 0B DC 85 62 CA EA 7E 46 EA B5 C0 B9 B2 96 22 5B 22 83 F3 33 DD AD 88 5A 9D 51
Key Stream @782,464  - AD AB 47 04 2C CD 7C 84 2E DD 82 4B BF 3F 23 ED 97 8F 0A 96 6A 4F A9 0A 6F 31 2D C6 A4 FD
                       || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Internal Data Stream - 35 CA FF 8B 82 D8 AE A5 16 17 59 F1 EC 9A E0 98 FA 7A 4C 82 87 7C 07 E9 E9 3C 54 BA 60 B1


 72 61 74 65 20 74 68 65 73 65 20 73 74 65 70 73 20 6F 66 20 74 68 65 20 27 52 61 6E 64 6F 6D 20 43 69 70 68 65
 || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
 79 B1 53 58 28 72 F0 96 07 9E 9D B0 CC B1 3B 52 45 6E 02 74 83 F5 9A 91 EB E1 FF 2A D7 67 85 1D 43 31 8B 5F B1
 0D 11 60 1C 0A 06 30 85 62 24 25 5F DB 4F B9 E2 3E 87 9E 17 25 86 46 04 3D F7 D4 C1 3D ED B0 F5 38 A4 0B 69 E8
 44 04 D4 A5 31 1D 14 2A 2D E4 3F 78 94 01 5B 01 AC 9C A5 9B 31 39 67 38 D8 8A 44 F4 72 8B EF 2F 03 66 F1 33 03
 89 EB 51 CB 20 C9 4E 31 DF B3 DC 71 59 37 8B 57 B5 85 BA AC 28 A9 24 4B DC 27 34 82 E5 8F C6 76 C0 03 44 09 EE
 || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
 CB 2E C2 4F 13 D4 F2 6D E4 88 7B 95 AE AD 22 95 42 9F E5 74 CB 8B FA C6 F5 E9 3A F3 19 E1 71 91 FB 99 45 64 D1


 72 20 4F 75 74 70 75 74 73 27 20 6D 6F 64 65 20 75 73 69 6E 67 20 55 4E 41 54 54 41 43 4B 41 42 4C 45 20 41 6C
 || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
 91 68 CC B9 DF 69 E7 0F B3 D4 5B 7A A9 A2 8D B8 DA 9D CD 72 D6 40 01 6D 3E 19 E5 0A 21 07 36 79 59 0B 9F 6C 24
 F6 85 31 AF 98 3F 6F D9 34 33 F8 46 D5 60 CE C6 5D B9 7E 03 95 3D 07 D3 35 21 E0 7F D9 2D 8B 32 AA 98 B1 52 AA
 6B EE AA 0C 42 EC A5 7A E4 49 D0 55 87 C2 7B 2C A6 F6 5C 45 0C A5 1F 3E C1 7D A0 65 5B 3E CD 4A 7F 5C BE 23 9C
 D6 8B 07 84 10 0A 2C 99 33 38 C9 AA AE 79 7B E3 8F 42 DB FA A2 48 7E A1 2A E6 45 5A 6C 39 2D 9D 69 F2 31 A4 F2
 || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || ||
 A8 A8 1F EB 61 C0 74 41 23 B1 9A AE 3A 1D 26 91 DB E3 5D A0 8A B0 32 6F A1 F7 B4 0B 8C 66 1C DE A9 78 81 F8 8C


                    67 65 62 72 61 69 63 20 6C 61 77 20 66 6F 72 20 73 65 63 75 72 69 74 79
                    || || || || || || || || || || || || || || || || || || || || || || || ||
                    30 F0 E5 25 4F 2B 42 7D 92 5C B3 C6 B8 6C 38 0C A6 50 5F 0A 47 FD B0 7A
                    7B 34 DB 90 56 FE ED A0 68 D0 00 AA 21 F4 77 F7 4E 31 B6 1F 86 DB BA 65
                    C6 CD 39 60 23 5A 5E 89 63 6E 49 C4 A8 66 6A 91 33 35 7F 8E CC E4 27 45
                    64 1A 36 4E BE C6 FC 5A 85 2F 1D D0 1E 74 B8 22 59 3D 4E 55 14 B3 17 0D
                    || || || || || || || || || || || || || || || || || || || || || || || ||
                    8E 76 53 E9 E5 20 6E 2E 70 AC 90 58 49 E5 EF 68 F1 0C BB BB 6B 18 4E 2E
```

Illustrated on the next page is the process carried out by the second engine to quite literally shred the output of this engine producing the ciphertext block.

Illustrated below is the operation of the Transposition Engine on the Vernam output on the plaintext sample on the previous page. The 'Internal Data Stream' on the previous page is copied here as the input to the Transposition Engine. The digit number is listed in vertical format just above each digit in the stream. Digit '**10**' (indicated in **blue**) for example is a '**2**'. Digit '**122**' is a '**1**'. The Transposition Key Number is the number selected out of 4,096 keys, the offset is the start of the access to that particular key. Remember, each key consists of 256 unique numbers with 255 of them copied to the top portion so that there are effectively 256 keys from this one key in memory. The 'Ciphertext Output' is the output of the Transposition engine. The vertical format number above each digit is the Transposition key number where the engine obtained the Internal Data Stream or Vernam Output digit. For example, notice the first digit is '8', 80 is in vertical format just above it. The '8' in the 'Internal Data Stream' second part in red was where that first digit was obtained. The second digit '0' was obtained from position '146' also in red. Digit 29, '8', obtained from 202.

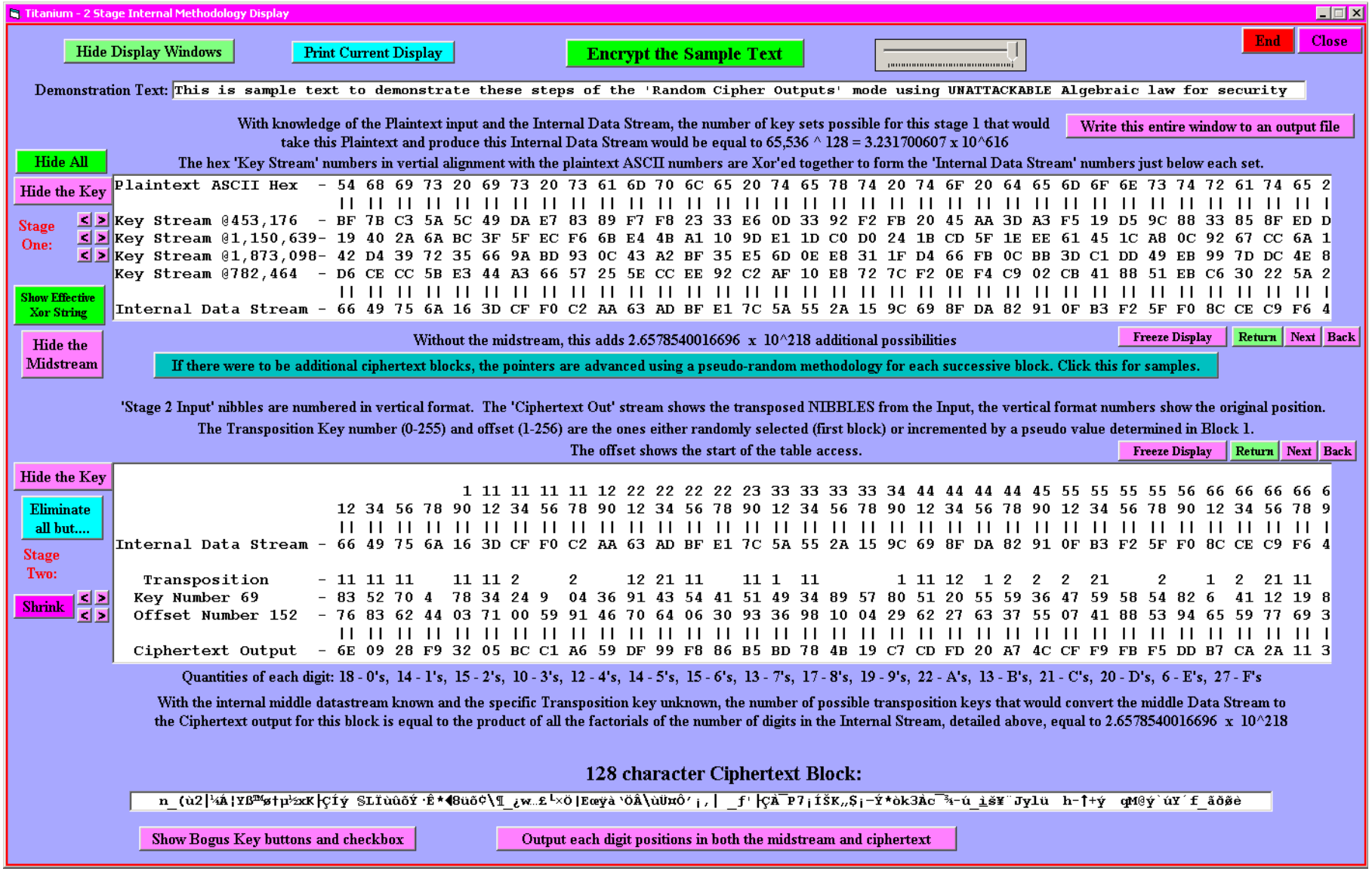
The Transposition Engine quite literally shreds the output of the Vernam Engine. Without the Transposition Key, the first digit '8' could be any one of the 21 - '8' digits in the Internal Data Stream. With the input being defined as pseudo-random, current technology that would normally be used to break the Transposition text would be totally worthless. Since this stream is internal to the cipher, they will never have access to this data stream, any more than they would have access to any internal data while the AES is executing. For example, look at the 2-digit **green** Vernam pair, 23 and 24 ('F1'). Notice that digit 24 ('1') is in the second line of the ciphertext, 23 ('F') is in the third line. Ask yourself how any mathematical process will be able to not only find those two digits but put them in the correct order.

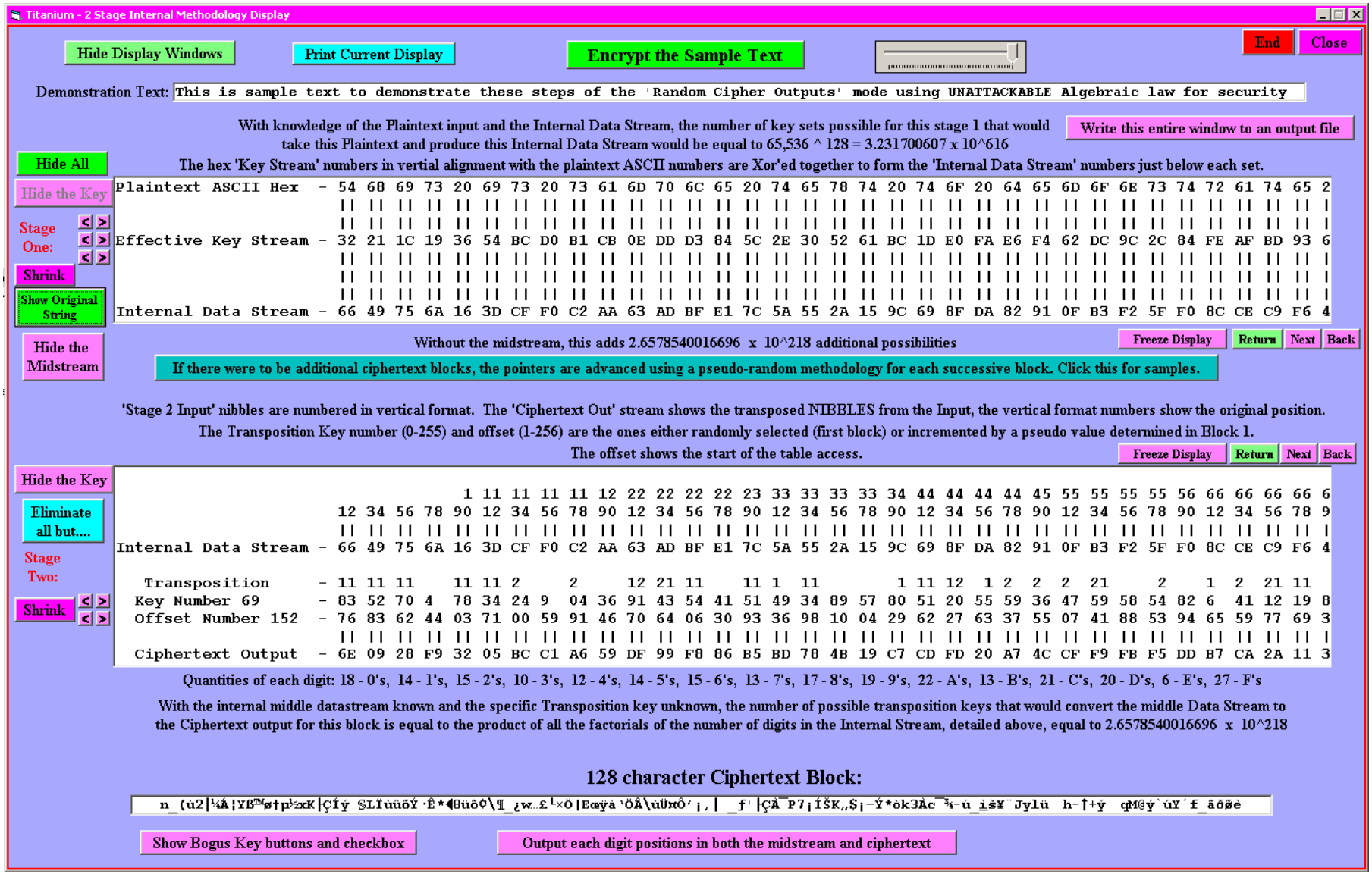
```
                                 1111111111222222222233333333334444444444555555555566666666667777777
                          12345678901234567890123456789012345678901234567890123456789012345678901234567890123456
                          ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Internal Data Stream - 35CAFF8B82D8AEA5161759F1EC9AE098FA7A4C82877C07E9E93C54BA60B1CB2EC24F13D4F26D
-------------------------------^------------^^

   Transposition     -  1  1  21211  211 211211 1  2 11 1 1111    21111  12 1 1 111111111 2211112 1
  Key Number 69      - 84 5181472549415115800793463011196842453 14592412734 185778746764931810910 19
  Offset Number 152  - 06411701396254952332182749952540946574797492257671609941948511008455157934 68
                       ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
  Ciphertext Output  - 80A3EA2FA9AB9C299A4FCCDDA4178AA576EC91A88EE8CF739D888E528048E13A46EE6FCA515E
-----------------------^^--------------------------^

                          11111111111111111111111111111111111111111111111111111111111111111111111111111111
77788888888899999999990000000001111111112222222233333333334444444445555555555666666666677777777
78901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
E4887B95AEAD2295429FE574CB8BFAC6F5E93AF319E17191FB994564D1A8A81FEB61C0744123B19AAE3A1D2691DBE35DA08
-----------------------------------------------^------------------------^

2212 2   1 2  212122 22  211 2 1 22 1 1 1121 21   112221 1 2 11   22 1 21 1 2    1 111 11  11   1 2
24614357481342440044315 72652163312 687 3438529 96634329389012376251883082315464967207507729297419 2
75988094101155303246065551294207806131622333583637548441750688578964891761994223178487751005976638 5
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
ABE590B282D50E0F8BCB092F66D3161ACE0327A5464ABC6F4491B9E84F2869AEFEE3D0987511E7B79D416B6F53714ED7357
-----------------------------^

11111111111111111111111222222222222222222222222222222222222222222222222222222222222222
77778888888888999999999900000000011111111122222222233333333334444444444555555555
67890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
AB0326FA1F7B40B8C661CDEA9781F88C8E7653E9E5206E2E70AC905849E5EF68F10CBBBB6B184E2E
---------------------------^

 211   22  1111 111 211 2211 111 21 112   1 2  212    1 1122 221 2   12 1 1 21  2
3527 654124092612198380634303557438525485535 2 226318360835007503229961175906322 50
2068368713706382080384959236608902701395231628606772334982912100236447734231628 45
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
8BB0C20B7FE4C9B819B9F1FC61DAA1E328B97B6AC56AEBC01E1BF2B28387419592F2E1EF158C51A4F
---------^-----------------------------------------------------------------------^
```

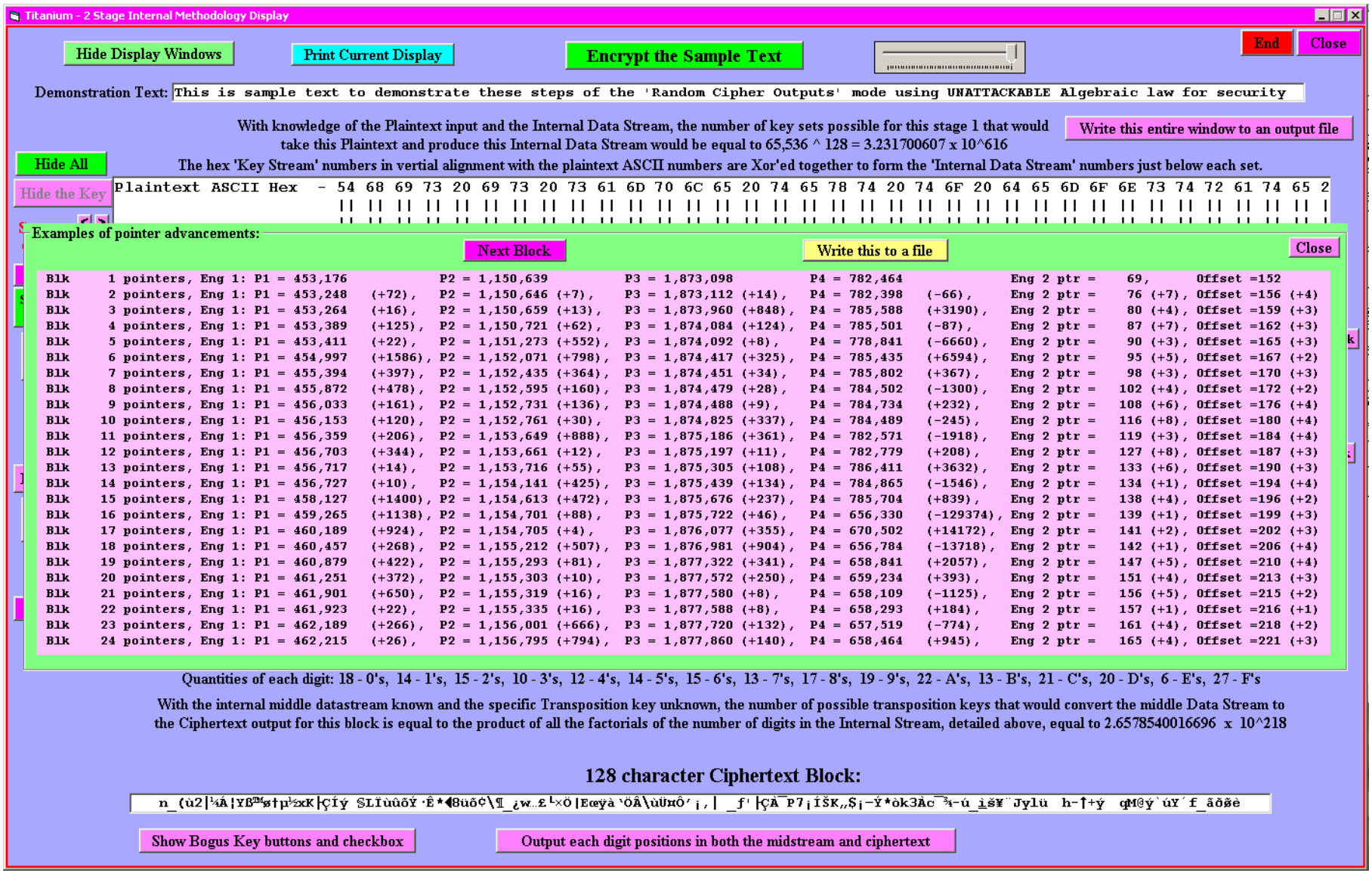
Pairs of hex digits are grouped together, converted to characters and the ciphertext has standard ciphertext appearance. Due to the non-printability of some of the characters, they are converted to something that is printable here.

€£ê/©«œ)šOÌÝ¤Š¥vì‘¨ŽèÏs • ˆŽR€Há:FîoÊQ^«å • ²,Õ_‹Ë_/fÓ_Î'¥FJ¼oD'¹èO(i®þãÐ˜uç· • AkoSqN×5x»_ ·þL› • ›ŸÆª-2‹—¶¬V®¼á¿+(8tY/.-ñXÅ_O

Appendix C


Titanium - 2 Stage Internal Methodology Display
Hide Display Windows
Print Current Display
Encrypt the Sample Text
End
Close
Demonstration Text: This is sample text to demonstrate these steps of the 'Random Cipher Outputs' mode using UNATTACKABLE Algebraic law for security
With knowledge of the Plaintext input and the Internal Data Stream, the number of key sets possible for this stage 1 that would take this Plaintext and produce this Internal Data Stream would be equal to 65,536 ^ 128 = 3.231700607 x 10^616
Write this entire window to an output file
Hide All
The hex 'Key Stream' numbers in vertial alignment with the plaintext ASCII numbers are Xor'ed together to form the 'Internal Data Stream' numbers just below each set.
Hide the Key
Stage One:
Show Effective Xor String
Plaintext ASCII Hex - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74 72 61 74 65 2
Key Stream @453,176 - BF 7B C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F ED D
Key Stream @1,150,639- 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8 0C 92 67 CC 6A 1
Key Stream @1,873,098- 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49 EB 99 7D DC 4E 8
Key Stream @782,464 - D6 CE CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6 30 22 5A 2
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0 8C CE C9 F6 4
Hide the Midstream
Without the midstream, this adds 2.6578540016696 x 10^218 additional possibilities
Freeze Display
Return
Next
Back
If there were to be additional ciphertext blocks, the pointers are advanced using a pseudo-random methodology for each successive block. Click this for samples.
'Stage 2 Input' nibbles are numbered in vertical format. The 'Ciphertext Out' stream shows the transposed NIBBLES from the Input, the vertical format numbers show the original position.
The Transposition Key number (0-255) and offset (1-256) are the ones either randomly selected (first block) or incremented by a pseudo value determined in Block 1.
The offset shows the start of the table access.
Freeze Display
Return
Next
Back
Hide the Key
Eliminate all but....
Stage Two:
Shrink
1 11 11 11 11 12 22 22 22 22 23 33 33 33 33 34 44 44 44 44 45 55 55 55 55 56 66 66 66 66 6
12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 9
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0 8C CE C9 F6 4
Transposition - 11 11 11 11 11 2 2 12 21 11 11 1 11 1 11 12 1 2 2 2 21 2 1 2 21 11
Key Number 69 - 83 52 70 4 78 34 24 9 04 36 91 43 54 41 51 49 34 89 57 80 51 20 55 59 36 47 59 58 54 82 6 41 12 19 8
Offset Number 152 - 76 83 62 44 03 71 00 59 91 46 70 64 06 30 93 36 98 10 04 29 62 27 63 37 55 07 41 88 53 94 65 59 77 69 3
Ciphertext Output - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD B7 CA 2A 11 3
Quantities of each digit: 18 - 0's, 14 - 1's, 15 - 2's, 10 - 3's, 12 - 4's, 14 - 5's, 15 - 6's, 13 - 7's, 17 - 8's, 19 - 9's, 22 - A's, 13 - B's, 21 - C's, 20 - D's, 6 - E's, 27 - F's
With the internal middle datastream known and the specific Transposition key unknown, the number of possible transposition keys that would convert the middle Data Stream to the Ciphertext output for this block is equal to the product of all the factorials of the number of digits in the Internal Stream, detailed above, equal to 2.6578540016696 x 10^218
128 character Ciphertext Block:
Show Bogus Key buttons and checkbox
Output each digit positions in both the midstream and ciphertext


Titanium - 2 Stage Internal Methodology Display
Hide Display Windows
Print Current Display
Encrypt the Sample Text
End
Close
Demonstration Text: This is sample text to demonstrate these steps of the 'Random Cipher Outputs' mode using UNATTACKABLE Algebraic law for security
With knowledge of the Plaintext input and the Internal Data Stream, the number of key sets possible for this stage 1 that would take this Plaintext and produce this Internal Data Stream would be equal to 65,536 ^ 128 = 3.231700607 x 10^616
Write this entire window to an output file
Hide All
The hex 'Key Stream' numbers in vertial alignment with the plaintext ASCII numbers are Xor'ed together to form the 'Internal Data Stream' numbers just below each set.
Hide the Key
Stage One:
Shrink
Show Original String
Plaintext ASCII Hex - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74 72 61 74 65 2
Effective Key Stream - 32 21 1C 19 36 54 BC D0 B1 CB 0E DD D3 84 5C 2E 30 52 61 BC 1D E0 FA E6 F4 62 DC 9C 2C 84 FE AF BD 93 6
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0 8C CE C9 F6 4
Hide the Midstream
Without the midstream, this adds 2.6578540016696 x 10^218 additional possibilities
Freeze Display
Return
Next
Back
If there were to be additional ciphertext blocks, the pointers are advanced using a pseudo-random methodology for each successive block. Click this for samples.
'Stage 2 Input' nibbles are numbered in vertical format. The 'Ciphertext Out' stream shows the transposed NIBBLES from the Input, the vertical format numbers show the original position.
The Transposition Key number (0-255) and offset (1-256) are the ones either randomly selected (first block) or incremented by a pseudo value determined in Block 1.
The offset shows the start of the table access.
Freeze Display
Return
Next
Back
Hide the Key
Eliminate all but....
Stage Two:
Shrink
Internal Data Stream - 66 49 75 6A 16 3D CF F0 C2 AA 63 AD BF E1 7C 5A 55 2A 15 9C 69 8F DA 82 91 0F B3 F2 5F F0 8C CE C9 F6 4
Transposition - 11 11 11 11 11 2 2 12 21 11 11 1 11 1 11 12 1 2 2 2 21 2 1 2 21 11
Key Number 69 - 83 52 70 4 78 34 24 9 04 36 91 43 54 41 51 49 34 89 57 80 51 20 55 59 36 47 59 58 54 82 6 41 12 19 8
Offset Number 152 - 76 83 62 44 03 71 00 59 91 46 70 64 06 30 93 36 98 10 04 29 62 27 63 37 55 07 41 88 53 94 65 59 77 69 3
Ciphertext Output - 6E 09 28 F9 32 05 BC C1 A6 59 DF 99 F8 86 B5 BD 78 4B 19 C7 CD FD 20 A7 4C CF F9 FB F5 DD B7 CA 2A 11 3
Quantities of each digit: 18 - 0's, 14 - 1's, 15 - 2's, 10 - 3's, 12 - 4's, 14 - 5's, 15 - 6's, 13 - 7's, 17 - 8's, 19 - 9's, 22 - A's, 13 - B's, 21 - C's, 20 - D's, 6 - E's, 27 - F's
With the internal middle datastream known and the specific Transposition key unknown, the number of possible transposition keys that would convert the middle Data Stream to the Ciphertext output for this block is equal to the product of all the factorials of the number of digits in the Internal Stream, detailed above, equal to 2.6578540016696 x 10^218
128 character Ciphertext Block:
Show Bogus Key buttons and checkbox
Output each digit positions in both the midstream and ciphertext


Titanium - 2 Stage Internal Methodology Display
Hide Display Windows
Print Current Display
Encrypt the Sample Text
End
Close
Demonstration Text: This is sample text to demonstrate these steps of the 'Random Cipher Outputs' mode using UNATTACKABLE Algebraic law for security
With knowledge of the Plaintext input and the Internal Data Stream, the number of key sets possible for this stage 1 that would take this Plaintext and produce this Internal Data Stream would be equal to 65,536 ^ 128 = 3.231700607 x 10^616
Write this entire window to an output file
Hide All
The hex 'Key Stream' numbers in vertial alignment with the plaintext ASCII numbers are Xor'ed together to form the 'Internal Data Stream' numbers just below each set.
Hide the Key
Plaintext ASCII Hex - 54 68 69 73 20 69 73 20 73 61 6D 70 6C 65 20 74 65 78 74 20 74 6F 20 64 65 6D 6F 6E 73 74 72 61 74 65 2
Examples of pointer advancements:
Next Block
Write this to a file
Close
Blk 1 pointers, Eng 1: P1 = 453,176 P2 = 1,150,639 P3 = 1,873,098 P4 = 782,464 Eng 2 ptr = 69, Offset =152
Blk 2 pointers, Eng 1: P1 = 453,248 (+72), P2 = 1,150,646 (+7), P3 = 1,873,112 (+14), P4 = 782,398 (-66), Eng 2 ptr = 76 (+7), Offset =156 (+4)
Blk 3 pointers, Eng 1: P1 = 453,264 (+16), P2 = 1,150,659 (+13), P3 = 1,873,960 (+848), P4 = 785,588 (+3190), Eng 2 ptr = 80 (+4), Offset =159 (+3)
Blk 4 pointers, Eng 1: P1 = 453,389 (+125), P2 = 1,150,721 (+62), P3 = 1,874,084 (+124), P4 = 785,501 (-87), Eng 2 ptr = 87 (+7), Offset =162 (+3)
Blk 5 pointers, Eng 1: P1 = 453,411 (+22), P2 = 1,151,273 (+552), P3 = 1,874,092 (+8), P4 = 778,841 (-6660), Eng 2 ptr = 90 (+3), Offset =165 (+3)
Blk 6 pointers, Eng 1: P1 = 454,997 (+1586), P2 = 1,152,071 (+798), P3 = 1,874,417 (+325), P4 = 785,435 (+6594), Eng 2 ptr = 95 (+5), Offset =167 (+2)
Blk 7 pointers, Eng 1: P1 = 455,394 (+397), P2 = 1,152,435 (+364), P3 = 1,874,451 (+34), P4 = 785,802 (+367), Eng 2 ptr = 98 (+3), Offset =170 (+3)
Blk 8 pointers, Eng 1: P1 = 455,872 (+478), P2 = 1,152,595 (+160), P3 = 1,874,479 (+28), P4 = 784,502 (-1300), Eng 2 ptr = 102 (+4), Offset =172 (+2)
Blk 9 pointers, Eng 1: P1 = 456,033 (+161), P2 = 1,152,731 (+136), P3 = 1,874,488 (+9), P4 = 784,734 (+232), Eng 2 ptr = 108 (+6), Offset =176 (+4)
Blk 10 pointers, Eng 1: P1 = 456,153 (+120), P2 = 1,152,761 (+30), P3 = 1,874,825 (+337), P4 = 784,489 (-245), Eng 2 ptr = 116 (+8), Offset =180 (+4)
Blk 11 pointers, Eng 1: P1 = 456,359 (+206), P2 = 1,153,649 (+888), P3 = 1,875,186 (+361), P4 = 782,571 (-1918), Eng 2 ptr = 119 (+3), Offset =184 (+4)
Blk 12 pointers, Eng 1: P1 = 456,703 (+344), P2 = 1,153,661 (+12), P3 = 1,875,197 (+11), P4 = 782,779 (+208), Eng 2 ptr = 127 (+8), Offset =187 (+3)
Blk 13 pointers, Eng 1: P1 = 456,717 (+14), P2 = 1,153,716 (+55), P3 = 1,875,305 (+108), P4 = 786,411 (+3632), Eng 2 ptr = 133 (+6), Offset =190 (+3)
Blk 14 pointers, Eng 1: P1 = 456,727 (+10), P2 = 1,154,141 (+425), P3 = 1,875,439 (+134), P4 = 784,865 (-1546), Eng 2 ptr = 134 (+1), Offset =194 (+4)
Blk 15 pointers, Eng 1: P1 = 458,127 (+1400), P2 = 1,154,613 (+472), P3 = 1,875,676 (+237), P4 = 785,704 (+839), Eng 2 ptr = 138 (+4), Offset =196 (+2)
Blk 16 pointers, Eng 1: P1 = 459,265 (+1138), P2 = 1,154,701 (+88), P3 = 1,875,722 (+46), P4 = 656,330 (-129374), Eng 2 ptr = 139 (+1), Offset =199 (+3)
Blk 17 pointers, Eng 1: P1 = 460,189 (+924), P2 = 1,154,705 (+4), P3 = 1,876,077 (+355), P4 = 670,502 (+14172), Eng 2 ptr = 141 (+2), Offset =202 (+3)
Blk 18 pointers, Eng 1: P1 = 460,457 (+268), P2 = 1,155,212 (+507), P3 = 1,876,981 (+904), P4 = 656,784 (-13718), Eng 2 ptr = 142 (+1), Offset =206 (+4)
Blk 19 pointers, Eng 1: P1 = 460,879 (+422), P2 = 1,155,293 (+81), P3 = 1,877,322 (+341), P4 = 658,841 (+2057), Eng 2 ptr = 147 (+5), Offset =210 (+4)
Blk 20 pointers, Eng 1: P1 = 461,251 (+372), P2 = 1,155,303 (+10), P3 = 1,877,572 (+250), P4 = 659,234 (+393), Eng 2 ptr = 151 (+4), Offset =213 (+3)
Blk 21 pointers, Eng 1: P1 = 461,901 (+650), P2 = 1,155,319 (+16), P3 = 1,877,580 (+8), P4 = 658,109 (-1125), Eng 2 ptr = 156 (+5), Offset =215 (+2)
Blk 22 pointers, Eng 1: P1 = 461,923 (+22), P2 = 1,155,335 (+16), P3 = 1,877,588 (+8), P4 = 658,293 (+184), Eng 2 ptr = 157 (+1), Offset =216 (+1)
Blk 23 pointers, Eng 1: P1 = 462,189 (+266), P2 = 1,156,001 (+666), P3 = 1,877,720 (+132), P4 = 657,519 (-774), Eng 2 ptr = 161 (+4), Offset =218 (+2)
Blk 24 pointers, Eng 1: P1 = 462,215 (+26), P2 = 1,156,795 (+794), P3 = 1,877,860 (+140), P4 = 658,464 (+945), Eng 2 ptr = 165 (+4), Offset =221 (+3)
Quantities of each digit: 18 - 0's, 14 - 1's, 15 - 2's, 10 - 3's, 12 - 4's, 14 - 5's, 15 - 6's, 13 - 7's, 17 - 8's, 19 - 9's, 22 - A's, 13 - B's, 21 - C's, 20 - D's, 6 - E's, 27 - F's
With the internal middle datastream known and the specific Transposition key unknown, the number of possible transposition keys that would convert the middle Data Stream to the Ciphertext output for this block is equal to the product of all the factorials of the number of digits in the Internal Stream, detailed above, equal to 2.6578540016696 x 10^218
128 character Ciphertext Block:
Show Bogus Key buttons and checkbox
Output each digit positions in both the midstream and ciphertext

Appendix D

```
Key Stream @453,176   - BF 7B C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,464   - D6 CE CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51
Effective Key Stream  - 32 21 1C 19 36 54 BC D0 B1 CB 0E DD D3 84 5C 2E 30 52 61 BC 1D E0 FA E6 F4 62 DC 9C 2C

Key Stream @453,177   - 7B C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,465   - CE CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB
Effective Key Stream  - EE 9B 12 A7 84 20 44 85 C9 CE 93 24 BF 01 DA AF 69 A8 66 E9 84 F5 50 B3 6B 04 D9 0C 82

Key Stream @453,178   - C3 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,466   - CC 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6
Effective Key Stream  - 54 95 AC 15 F0 D8 11 FD CC 53 6A 48 3A 87 5B F6 93 AF 33 70 91 5F 05 2C 0D 01 49 A2 14

Key Stream @453,179   - 5A 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,467   - 5B E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6 30
Effective Key Stream  - 5A 2B 1E 61 08 8D 69 F8 51 AA 06 CD BC 06 02 0C 94 FA AA 65 3B 0A 9A 4A 08 91 E7 34 54

Key Stream @453,180   - 5C 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,468   - E3 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6 30 22
Effective Key Stream  - E4 99 6A 99 5D F5 6C 65 A8 C6 83 4B 3D 5F F8 0B C1 63 BF CF 6E 95 FC 4F 98 3F 71 74 4C

Key Stream @453,181   - 49 DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F ED
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,469   - 44 A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6 30 22 5A
Effective Key Stream  - 56 ED 92 CC 25 F0 F1 9C C4 43 05 CA 64 A5 FF 5E 58 76 15 9A F1 F3 F9 DF 36 A9 31 6C 56

Key Stream @453,182   - DA E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F ED D2
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,470   - A3 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6 30 22 5A 2F
Effective Key Stream  - 22 15 C7 B4 20 6D 08 F0 41 C5 84 93 9E A2 AA C7 4D DC 40 05 97 F6 69 71 A0 E9 29 76 1C

Key Stream @453,183   - E7 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F ED D2 0E
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,471   - 66 57 25 5E CC EE 92 C2 AF 10 E8 72 7C F2 0E F4 C9 02 CB 41 88 51 EB C6 30 22 5A 2F DB
Effective Key Stream  - DA 40 BF B1 BD 94 64 75 C7 44 DD 69 99 F7 33 D2 E7 89 DF 63 92 66 C7 E7 E0 F1 33 3C 34

Key Stream @453,184   - 83 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F ED D2 0E 50
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,584   - AA 9B 6B 56 29 66 E7 A7 70 7A A4 F1 48 7D BE 0A 0A FB 06 4D 39 80 DA 18 BE DB DD F4 21
Effective Key Stream  - 72 86 8F B6 83 0C C4 FB 26 8F F1 E3 76 1D 6C BB BA 26 FE A3 6A A3 4D 8F 64 6A 8B 3B 90

Key Stream @453,185   - 89 F7 F8 23 33 E6 0D 33 92 F2 FB 20 45 AA 3D A3 F5 19 D5 9C 88 33 85 8F ED D2 0E 50 2A
Key Stream @1,150,639 - 19 40 2A 6A BC 3F 5F EC F6 6B E4 4B A1 10 9D E1 1D C0 D0 24 1B CD 5F 1E EE 61 45 1C A8
Key Stream @1,873,098 - 42 D4 39 72 35 66 9A BD 93 0C 43 A2 BF 35 E5 6D 0E E8 31 1F D4 66 FB 0C BB 3D C1 DD 49
Key Stream @782,585   - 9B 6B 56 29 66 E7 A7 70 7A A4 F1 48 7D BE 0A 0A FB 06 4D 39 80 DA 18 BE DB DD F4 21 2C
Effective Key Stream  - 49 08 BD 12 DC 58 6F 12 8D 31 AD 81 26 31 4F 25 1D 37 79 9E C7 42 39 23 63 53 7E B0 E7


Effective Key Stream  - 32 21 1C 19 36 54 BC D0 B1 CB 0E DD D3 84 5C 2E 30 52 61 BC 1D E0 FA E6 F4 62 DC 9C 2C
Effective Key Stream  - EE 9B 12 A7 84 20 44 85 C9 CE 93 24 BF 01 DA AF 69 A8 66 E9 84 F5 50 B3 6B 04 D9 0C 82
Effective Key Stream  - 54 95 AC 15 F0 D8 11 FD CC 53 6A 48 3A 87 5B F6 93 AF 33 70 91 5F 05 2C 0D 01 49 A2 14
Effective Key Stream  - 5A 2B 1E 61 08 8D 69 F8 51 AA 06 CD BC 06 02 0C 94 FA AA 65 3B 0A 9A 4A 08 91 E7 34 54
Effective Key Stream  - E4 99 6A 99 5D F5 6C 65 A8 C6 83 4B 3D 5F F8 0B C1 63 BF CF 6E 95 FC 4F 98 3F 71 74 4C
Effective Key Stream  - 56 ED 92 CC 25 F0 F1 9C C4 43 05 CA 64 A5 FF 5E 58 76 15 9A F1 F3 F9 DF 36 A9 31 6C 56
Effective Key Stream  - 22 15 C7 B4 20 6D 08 F0 41 C5 84 93 9E A2 AA C7 4D DC 40 05 97 F6 69 71 A0 E9 29 76 1C
Effective Key Stream  - DA 40 BF B1 BD 94 64 75 C7 44 DD 69 99 F7 33 D2 E7 89 DF 63 92 66 C7 E7 E0 F1 33 3C 34
Effective Key Stream  - 72 86 8F B6 83 0C C4 FB 26 8F F1 E3 76 1D 6C BB BA 26 FE A3 6A A3 4D 8F 64 6A 8B 3B 90
Effective Key Stream  - 49 08 BD 12 DC 58 6F 12 8D 31 AD 81 26 31 4F 25 1D 37 79 9E C7 42 39 23 63 53 7E B0 E7
```

Appendix E

This function is called to advance the pointers, calling the functions below as warranted to advance the individual pointers. Notice that the combined (Xor) values of the 4 pointers are used to determine the order the pointers are advanced. It should be obvious that the order of advancement will further affect at least some of the pointer's advancements.

```
Sub advancePointers()
    Dim o1 As Long, o2 As Long, o3 As Long, chg1 As Long, chg2 As Long, chg3 As Long, e1k1 As Long, e1k2 As Long, e1k3 As Long

    ' Save the current pointers for checking that their change isn't the same as any other pointer's change
    o1 = e1p1: o2 = e1p2: o3 = e1p3
    ' Get the values from the key that each pointer is pointing to and make sure these values are not the same as any other
    e1k1 = ln(e1Key(e1p1) + 223)
    e1k2 = ln(e1Key(e1p2) + 223): Do While e1k2 = e1k1: e1p2 = e1p2 + 1: e1k2 = ln(e1Key(e1p2)) And &HF&: Loop
    e1k3 = ln(e1Key(e1p3) + 223): Do While e1k3 = e1k1 Or e1k3 = e1k2: e1p3 = e1p3 + 1: e1k3 = ln(e1Key(e1p3)) And &HF&: Loop
    ' Decide which order to do the advancements - the order is critical because all 3 pointers are used in all 3 pointer advancements
    ' and depending upon the order, each pointer may be the 'old' OR the 'new' value
    Select Case (e1p1 Xor e1p2 Xor e1p3 Xor e1p4) Mod 6
        Case 0: chg1 = initialize_e1p1(e1k1, e1k2, e1k3, o1)
                chg2 = initialize_e1p2(e1k1, e1k2, e1k3, o2): If chg2 = chg1 Then e1p2 = (e1p2 + 1) And &H1FFFFF: chg2 = chg2 + 1
                chg3 = initialize_e1p3(e1k1, e1k2, e1k3, o3): Do While chg1 = chg3 Or chg3 = chg2: e1p3 = e1p3 + 1: chg3 = chg3 + 1: Loop
        Case 1: chg1 = initialize_e1p1(e1k1, e1k2, e1k3, o1)
                chg3 = initialize_e1p3(e1k1, e1k2, e1k3, o3): If chg3 = chg1 Then e1p3 = (e1p3 + 1) And &H1FFFFF: chg3 = chg3 + 1
                chg2 = initialize_e1p2(e1k1, e1k2, e1k3, o2): Do While chg2 = chg1 Or chg2 = chg3: e1p2 = e1p2 + 1: chg2 = chg2 + 1: Loop
        Case 2: chg2 = initialize_e1p2(e1k1, e1k2, e1k3, o2)
                chg1 = initialize_e1p1(e1k1, e1k2, e1k3, o1): If chg2 = chg1 Then e1p1 = (e1p1 + 1) And &H1FFFFF: chg1 = chg1 + 1
                chg3 = initialize_e1p3(e1k1, e1k2, e1k3, o3): Do While chg3 = chg1 Or chg3 = chg2: e1p3 = e1p3 + 1: chg3 = chg3 + 1: Loop
        Case 3: chg2 = initialize_e1p2(e1k1, e1k2, e1k3, o2)
                chg3 = initialize_e1p3(e1k1, e1k2, e1k3, o3): If chg3 = chg2 Then e1p3 = (e1p3 + 1) And &H1FFFFF: chg3 = chg3 + 1
                chg1 = initialize_e1p1(e1k1, e1k2, e1k3, o1): Do While chg1 = chg2 Or chg1 = chg3: e1p1 = e1p1 + 1: chg1 = chg1 + 1: Loop
        Case 4: chg3 = initialize_e1p3(e1k1, e1k2, e1k3, o3)
                chg1 = initialize_e1p1(e1k1, e1k2, e1k3, o1): If chg1 = chg3 Then e1p1 = (e1p1 + 1) And &H1FFFFF: chg1 = chg1 + 1
                chg2 = initialize_e1p2(e1k1, e1k2, e1k3, o2): Do While chg2 = chg1 Or chg2 = chg3: e1p2 = e1p2 + 1: chg2 = chg2 + 1: Loop
        Case 5: chg3 = initialize_e1p3(e1k1, e1k2, e1k3, o3)
                chg2 = initialize_e1p2(e1k1, e1k2, e1k3, o2): If chg2 = chg3 Then e1p2 = (e1p2 + 1) And &H1FFFFF: chg2 = chg2 + 1
                chg1 = initialize_e1p1(e1k1, e1k2, e1k3, o1): Do While chg1 = chg3 Or chg1 = chg2: e1p1 = e1p1 + 1: chg1 = chg1 + 1: Loop
    End Select
    ' Bring the pointers within legal range
    e1p1 = e1p1 And &H1FFFFF: e1p2 = e1p2 And &H1FFFFF: e1p3 = e1p3 And &H1FFFFF
    ' Check for pointer 1 usage
    If lastPtrUsage(e1p1) > 0 Then Do While (ctxBlockNum - lastPtrUsage(e1p1)) < 500000: e1p1 = (e1p1 + 1) And &H1FFFFF: Loop
    ' Set the block count to store what block this pointer number was used
    lastPtrUsage(e1p1) = ctxBlockNum
    ' Check for pointer 2 usage
    If lastPtrUsage(e1p2) > 0 Then Do While (ctxBlockNum - lastPtrUsage(e1p2)) < 500000: e1p2 = (e1p2 + 1) And &H1FFFFF: Loop
    ' Set the block count to store what block this pointer number was used
    lastPtrUsage(e1p2) = ctxBlockNum
    ' Check for pointer 3 usage
    If lastPtrUsage(e1p3) > 0 Then Do While (ctxBlockNum - lastPtrUsage(e1p3)) < 500000: e1p3 = (e1p3 + 1) And &H1FFFFF: Loop
    ' Set the block count to store what block this pointer number was used
    lastPtrUsage(e1p3) = ctxBlockNum
    ' Go initialize the Engine 2 pointer and offset
    Call initialize_e2Ptr_e2Off
    ' Produce pointer 4 as the Xor of the first 3 pointers plus the 2 engine 2 pointers
    e1p4 = e1p1 Xor e1p2 Xor e1p3 Xor e2Ptr Xor e2Off
End Sub
```

This function is called to advance the Engine 2 pointers.

```
Sub initialize_e2Ptr_e2Off()
    ' Now modify the Engine 2 pointers
    Select Case ((e1p1 Xor e1p2) Mod 7)
        Case 0:  e2Ptr = e2Ptr + (e1p1 And 7)
        Case 1:  e2Ptr = e2Ptr + (Int(e1p1 / &H10&) And 7)
        Case 2:  e2Ptr = e2Ptr + (Int(e1p1 / &H100&) And 7)
        Case 3:  e2Ptr = e2Ptr + (Int(e1p1 / &H1000&) And 7)
        Case 4:  e2Ptr = e2Ptr + (e1p2 And 7)
        Case 5:  e2Ptr = e2Ptr + (Int(e1p2 / &H10&) And 7)
        Case 6:  e2Ptr = e2Ptr + (Int(e1p2 / &H100&) And 7)
    End Select
    Select Case ((e1p3 Xor e1p4) Mod 7)
        Case 0:  e2Off = e2Off + (e1p2 And 3)
        Case 1:  e2Off = e2Off + (Int(e1p2 / &H40&) And 3)
        Case 2:  e2Off = e2Off + (Int(e1p2 / &H400&) And 3)
        Case 3:  e2Off = e2Off + (Int(e1p2 / &H4000&) And 3)
        Case 4:  e2Off = e2Off + (e1p3 And 3)
        Case 5:  e2Off = e2Off + (Int(e1p3 / &H40&) And 3)
        Case 6:  e2Off = e2Off + (Int(e1p3 / &H400&) And 3)
    End Select
    ' Restrict the pointers to their legal bounds
    e2Ptr = (e2Ptr + 1) And &H1FFF&: e2Off = (e2Off And &HFF&) + 1
End Sub
```

This function is called to advance the first Vernam 2 pointer.

Notice that it uses the then-current value to determine which case below to use to determine what mathematical operation is used to calculate the advancement value. Due to the order executed in the main function call, the values of the other pointers may or may not be already advanced.

```
Function initialize_e1p1(e1k1 As Long, e1k2 As Long, e1k3 As Long, o1 As Long) As Long
    Dim i As Byte, chg1 As Long, v2Use As Byte

    ' Select which combination of pointers to use based on current value of this pointer.  Note value of other pointers may be old OR new values
    Select Case (e1p1 Mod 25)
        Case 0: v2Use = e1p2 Mod 13
        Case 1: v2Use = e1p3 Mod 13
        Case 2: v2Use = e1p4 Mod 13
        Case 3: v2Use = e2Ptr Mod 13
        Case 4: v2Use = e2Off Mod 13
        Case 5: v2Use = (e1p2 Xor e1p3) Mod 13
        Case 6: v2Use = (e1p2 Xor e1p4) Mod 13
        Case 7: v2Use = (e1p2 Xor e2Ptr) Mod 13
        Case 8: v2Use = (e1p2 Xor e2Off) Mod 13
        Case 9: v2Use = (e1p3 Xor e1p4) Mod 13
        Case 10: v2Use = (e1p3 Xor e2Ptr) Mod 13
        Case 11: v2Use = (e1p3 Xor e2Off) Mod 13
        Case 12: v2Use = (e1p4 Xor e2Ptr) Mod 13
        Case 13: v2Use = (e1p4 Xor e2Off) Mod 13
        Case 14: v2Use = (e2Ptr Xor e2Off) Mod 13
        Case 15: v2Use = (e1p2 Xor e1p3 Xor e2Ptr) Mod 13
        Case 16: v2Use = (e1p2 Xor e1p3 Xor e2Off) Mod 13
        Case 17: v2Use = (e1p2 Xor e2Ptr Xor e2Off) Mod 13
        Case 18: v2Use = (e1p3 Xor e2Ptr Xor e2Off) Mod 13
        Case 19: v2Use = (e1p2 Xor e1p3 Xor e1p4 Xor e2Ptr) Mod 13
        Case 20: v2Use = (e1p2 Xor e1p3 Xor e1p4 Xor e2Off) Mod 13
        Case 21: v2Use = (e1p2 Xor e1p3 Xor e2Ptr Xor e2Off) Mod 13
        Case 22: v2Use = (e1p2 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
        Case 23: v2Use = (e1p3 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
        Case 24: v2Use = (e1p2 Xor e1p3 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
    End Select
    ' Loop the specified number of times for this pointer #1
    For i = 0 To ((e1p2 Xor e1p3) And 1)
        ' Advance the pointers according to the ciphertext block counter mod 7
        Select Case v2Use
            Case 0: e1p1 = e1p1 + e1k2
            Case 1: e1p1 = e1p1 + (e1k2 Xor e1k3) + 1
            Case 2: e1p1 = e1p1 + ((e1k2 + e1k3) And &HF&) + 1
            Case 3: If (e1k2 - e1k3) > 0 Then e1p1 = e1p1 + e1k2 - e1k3 + 1 Else e1p1 = e1p1 + e1k3 - e1k2
            Case 4: e1p1 = e1p1 + e1k3
            Case 5: e1p1 = e1p1 + (e1k3 Xor e1k1) + 1
            Case 6: e1p1 = e1p1 + ((e1k3 + e1k1) And &HF&) + 1
            Case 7: If (e1k3 - e1k1) > 0 Then e1p1 = e1p1 + e1k3 - e1k1 + 1 Else e1p1 = e1p1 + e1k1 - e1k3
            Case 8: e1p1 = e1p1 + e1k1
            Case 9: e1p1 = e1p1 + (e1k1 Xor e1k2) + 1
            Case 10: e1p1 = e1p1 + (e1k1 + e1k2) + 1
            Case 11: If (e1k1 - e1k2) > 0 Then e1p1 = e1p1 + e1k1 - e1k2 + 1 Else e1p1 = e1p1 + e1k2 - e1k1
            Case 12: e1p1 = e1p1 + (e1k1 Xor e1k2 Xor e1k3) + 1
        End Select
    Next i
    ' Limit the pointer to 0 to 2,097,151
    e1p1 = e1p1 And &H1FFFFF
    ' Set the location to indicate the change
    chg1 = Abs(e1p1 - o1): If chg1 = 0 Then e1p1 = e1p1 + 1: chg1 = 1
    ' Return the change number
    initialize_e1p1 = chg1
End Function
```

This function is called to advance the second Vernam 2 pointer.

Notice that it uses the then-current value to determine which case below to use to determine what mathematical operation is used to calculate the advancement value. Due to the order executed in the main function call, the values of the other pointers may or may not be already advanced.

```
Function initialize_e1p2(e1k1 As Long, e1k2 As Long, e1k3 As Long, o2 As Long) As Long
    Dim i As Byte, chg2 As Long, v2Use As Byte

    ' Select which combination of pointers to use based on current value of this pointer.  Note value of other pointers may be old OR new values
    Select Case (e1p2 Mod 25)
        Case 0: v2Use = e1p1 Mod 13
        Case 1: v2Use = e1p3 Mod 13
        Case 2: v2Use = e1p4 Mod 13
        Case 3: v2Use = e2Ptr Mod 13
        Case 4: v2Use = e2Off Mod 13
        Case 5: v2Use = (e1p1 Xor e1p3) Mod 13
        Case 6: v2Use = (e1p1 Xor e1p4) Mod 13
        Case 7: v2Use = (e1p1 Xor e2Ptr) Mod 13
        Case 8: v2Use = (e1p1 Xor e2Off) Mod 13
        Case 9: v2Use = (e1p3 Xor e1p4) Mod 13
        Case 10: v2Use = (e1p3 Xor e2Ptr) Mod 13
        Case 11: v2Use = (e1p3 Xor e2Off) Mod 13
        Case 12: v2Use = (e1p4 Xor e2Ptr) Mod 13
        Case 13: v2Use = (e1p4 Xor e2Off) Mod 13
        Case 14: v2Use = (e2Ptr Xor e2Off) Mod 13
        Case 15: v2Use = (e1p1 Xor e1p3 Xor e2Ptr) Mod 13
        Case 16: v2Use = (e1p1 Xor e1p3 Xor e2Off) Mod 13
        Case 17: v2Use = (e1p1 Xor e2Ptr Xor e2Off) Mod 13
        Case 18: v2Use = (e1p3 Xor e2Ptr Xor e2Off) Mod 13
        Case 19: v2Use = (e1p1 Xor e1p3 Xor e1p4 Xor e2Ptr) Mod 13
        Case 20: v2Use = (e1p1 Xor e1p3 Xor e1p4 Xor e2Off) Mod 13
        Case 21: v2Use = (e1p1 Xor e1p3 Xor e2Ptr Xor e2Off) Mod 13
        Case 22: v2Use = (e1p1 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
        Case 23: v2Use = (e1p3 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
        Case 24: v2Use = (e1p1 Xor e1p3 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
    End Select
    ' Loop the specified number of times for this pointer #2
    For i = 0 To ((e1p1 Xor e1p3) And 1)
        ' Advance the pointers according to the ciphertext block counter mod 7
        Select Case v2Use
            Case 0: e1p2 = e1p2 + e1k3
            Case 1: e1p2 = e1p2 + (e1k3 Xor e1k1) + 1
            Case 2: e1p2 = e1p2 + ((e1k3 + e1k1) And &HF&) + 1
            Case 3: If (e1k3 - e1k1) > 0 Then e1p2 = e1p2 + e1k3 - e1k1 + 1 Else e1p2 = e1p2 + e1k1 - e1k3
            Case 4: e1p2 = e1p2 + e1k1
            Case 5: e1p2 = e1p2 + (e1k1 Xor e1k2) + 1
            Case 6: e1p2 = e1p2 + ((e1k1 + e1k2) And &HF&) + 1
            Case 7: If (e1k1 - e1k2) > 0 Then e1p2 = e1p2 + e1k1 - e1k2 + 1 Else e1p2 = e1p2 + e1k2 - e1k1
            Case 8: e1p2 = e1p2 + e1k2
            Case 9: e1p2 = e1p2 + (e1k1 Xor e1k3) + 1
            Case 10: e1p2 = e1p2 + ((e1k1 + e1k3) And &HF&) + 1
            Case 11: If (e1k1 - e1k3) > 0 Then e1p2 = e1p2 + e1k1 - e1k3 + 1 Else e1p2 = e1p2 + e1k3 - e1k1
            Case 12: e1p2 = e1p2 + (e1k1 Xor e1k2 Xor e1k3) + 1
        End Select
    Next i
    ' Limit the pointer to 0 to 2,097,151
    e1p2 = e1p2 And &H1FFFFF
    ' Set the location to indicate the change
    chg2 = Abs(e1p2 - o2): If chg2 = 0 Then e1p2 = e1p2 + 1: chg2 = 1
    ' Return the change number
    initialize_e1p2 = chg2
End Function
```

This function is called to advance the third Vernam 2 pointer.

Notice that it uses the then-current value to determine which case below to use to determine what mathematical operation is used to calculate the advancement value. Due to the order executed in the main function call, the values of the other pointers may or may not be already advanced.

```
Function initialize_e1p3(e1k1 As Long, e1k2 As Long, e1k3 As Long, o3 As Long) As Long
    Dim i As Byte, chg3 As Long, v2Use As Byte

    ' Select which combination of pointers to use based on current value of this pointer.  Note value of other pointers may be old OR new values
    Select Case (e1p3 Mod 25)
        Case 0: v2Use = e1p1 Mod 13
        Case 1: v2Use = e1p2 Mod 13
        Case 2: v2Use = e1p4 Mod 13
        Case 3: v2Use = e2Ptr Mod 13
        Case 4: v2Use = e2Off Mod 13
        Case 5: v2Use = (e1p1 Xor e1p2) Mod 13
        Case 6: v2Use = (e1p1 Xor e1p4) Mod 13
        Case 7: v2Use = (e1p1 Xor e2Ptr) Mod 13
        Case 8: v2Use = (e1p1 Xor e2Off) Mod 13
        Case 9: v2Use = (e1p2 Xor e1p4) Mod 13
        Case 10: v2Use = (e1p2 Xor e2Ptr) Mod 13
        Case 11: v2Use = (e1p2 Xor e2Off) Mod 13
        Case 12: v2Use = (e1p4 Xor e2Ptr) Mod 13
        Case 13: v2Use = (e1p4 Xor e2Off) Mod 13
        Case 14: v2Use = (e2Ptr Xor e2Off) Mod 13
        Case 15: v2Use = (e1p1 Xor e1p2 Xor e2Ptr) Mod 13
        Case 16: v2Use = (e1p1 Xor e1p2 Xor e2Off) Mod 13
        Case 17: v2Use = (e1p1 Xor e2Ptr Xor e2Off) Mod 13
        Case 18: v2Use = (e1p2 Xor e2Ptr Xor e2Off) Mod 13
        Case 19: v2Use = (e1p1 Xor e1p2 Xor e1p4 Xor e2Ptr) Mod 13
        Case 20: v2Use = (e1p1 Xor e1p2 Xor e1p4 Xor e2Off) Mod 13
        Case 21: v2Use = (e1p1 Xor e1p2 Xor e2Ptr Xor e2Off) Mod 13
        Case 22: v2Use = (e1p1 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
        Case 23: v2Use = (e1p2 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
        Case 24: v2Use = (e1p1 Xor e1p2 Xor e1p4 Xor e2Ptr Xor e2Off) Mod 13
    End Select
    ' Loop the specified number of times for this pointer #3
    For i = 0 To ((e1p1 Xor e1p2) And 1)
        ' Advance the pointers according to the ciphertext block counter mod 100
        Select Case v2Use
            Case 0: e1p3 = e1p3 + e1k1
            Case 1: e1p3 = e1p3 + (e1k1 Xor e1k2) + 1
            Case 2: e1p3 = e1p3 + ((e1k1 + e1k2) And &HF&) + 1
            Case 3: If (e1k1 - e1k2) > 0 Then e1p3 = e1p3 + e1k1 - e1k2 + 1 Else e1p3 = e1p3 + e1k2 - e1k1
            Case 4: e1p3 = e1p3 + e1k2
            Case 5: e1p3 = e1p3 + (e1k1 Xor e1k3) + 1
            Case 6: e1p3 = e1p3 + ((e1k1 + e1k3) And &HF&) + 1
            Case 7: If (e1k1 - e1k3) > 0 Then e1p3 = e1p3 + e1k1 - e1k3 + 1 Else e1p3 = e1p3 + e1k3 - e1k1
            Case 8: e1p3 = e1p3 + e1k3
            Case 9: e1p3 = e1p3 + (e1k2 Xor e1k3) + 1
            Case 10: e1p3 = e1p3 + ((e1k2 + e1k3) And &HF&) + 1
            Case 11: If (e1k2 - e1k3) > 0 Then e1p3 = e1p3 + e1k2 - e1k3 + 1 Else e1p3 = e1p3 + e1k3 - e1k2
            Case 12: e1p3 = e1p3 + (e1k1 Xor e1k2 Xor e1k3) + 1
        End Select
    Next i
    ' Limit the pointer to 0 to 2,097,151
    e1p3 = e1p3 And &H1FFFFF
    ' Set the location to indicate the change
    chg3 = Abs(e1p3 - o3): If chg3 = 0 Then e1p3 = e1p3 + 1: chg3 = 1
    ' Return the change number
    initialize_e1p3 = chg3
End Function
```

On pages 26 and 27 are examples of how the above pointer advancement function behaves when using the pointers in the example on page 1 of this document. Notice how the 4th pointer can be changed in either a positive or negative direction depending upon the Xor of the other 5 pointer values.

To show how thoroughly this function works, I will 'rotate' the values (P1 receives P2's value, P2 receives P3's value, P3 receives P1's value) and you will see on pages 28 and 29 following that even though the pointers have the same values as the first example, the advancements are also position-sensitive. You will also be able to observe that just the position of the first 3 pointers also affects the advancement of the Engine 2 pointers.

Pages 30 and 31 show only pointers 2 and 3 exchanged, keeping pointer 1 the same as on pages 28 and 29. You will be able to see the change to pointer 1's changes as compared to those on pages 26 and 27. This will show how each pointer's advancement is dependent upon the other pointers.

With this type of interdependence on each pointer, plus key table values used and checking for previous use (within 500,000 blocks), the selection of pointer values for each block assures a non-repetitive **EKS** is used.

```
Blk     1 pointers, Eng 1: P1 = 1,725,418          P2 = 1,260,439          P3 = 774,161            P4 = 178,267            Eng 2 ptr =    49,      offset =  6
Blk     2 pointers, Eng 1: P1 = 1,725,433 (+15),   P2 = 1,260,446 (+7),    P3 = 775,103   (+942),  P4 = 179,170   (+903),  Eng 2 ptr =    51 (+2), offset =  9 (+3)
Blk     3 pointers, Eng 1: P1 = 1,726,241 (+808),  P2 = 1,260,609 (+163),  P3 = 775,418   (+315),  P4 = 180,136   (+966),  Eng 2 ptr =    57 (+6), offset = 11 (+2)
Blk     4 pointers, Eng 1: P1 = 1,726,628 (+387),  P2 = 1,260,623 (+14),   P3 = 775,467   (+49),   P4 = 176,629   (-3507), Eng 2 ptr =    58 (+1), offset = 15 (+4)
Blk     5 pointers, Eng 1: P1 = 1,726,695 (+67),   P2 = 1,261,039 (+416),  P3 = 775,871   (+404),  P4 = 177,124   (+495),  Eng 2 ptr =    64 (+6), offset = 19 (+4)
Blk     6 pointers, Eng 1: P1 = 1,727,353 (+658),  P2 = 1,261,294 (+255),  P3 = 776,749   (+878),  P4 = 180,203   (+3079), Eng 2 ptr =    71 (+7), offset = 22 (+3)
Blk     7 pointers, Eng 1: P1 = 1,728,175 (+822),  P2 = 1,261,518 (+224),  P3 = 776,919   (+170),  P4 = 179,171   (-1032), Eng 2 ptr =    79 (+8), offset = 26 (+4)
Blk     8 pointers, Eng 1: P1 = 1,728,813 (+638),  P2 = 1,261,521 (+3),    P3 = 777,279   (+360),  P4 = 164,492   (-14679), Eng 2 ptr =    81 (+2), offset = 30 (+4)
Blk     9 pointers, Eng 1: P1 = 1,729,266 (+453),  P2 = 1,261,550 (+29),   P3 = 777,307   (+28),   P4 = 164,096   (-396),  Eng 2 ptr =    88 (+7), offset = 31 (+1)
Blk    10 pointers, Eng 1: P1 = 1,730,054 (+788),  P2 = 1,261,562 (+12),   P3 = 777,341   (+34),   P4 = 165,373   (+1277), Eng 2 ptr =    95 (+7), offset = 35 (+4)
Blk    11 pointers, Eng 1: P1 = 1,730,896 (+842),  P2 = 1,262,254 (+692),  P3 = 777,597   (+256),  P4 = 194,242   (+28869), Eng 2 ptr =   102 (+7), offset = 39 (+4)
Blk    12 pointers, Eng 1: P1 = 1,731,083 (+187),  P2 = 1,262,940 (+686),  P3 = 777,604   (+7),    P4 = 193,175   (-1067), Eng 2 ptr =   108 (+6), offset = 40 (+1)
Blk    13 pointers, Eng 1: P1 = 1,731,084 (+1),    P2 = 1,263,418 (+478),  P3 = 778,538   (+934),  P4 = 183,366   (-9809), Eng 2 ptr =   113 (+5), offset = 43 (+3)
Blk    14 pointers, Eng 1: P1 = 1,731,407 (+323),  P2 = 1,263,838 (+420),  P3 = 778,552   (+14),   P4 = 180,978   (-2388), Eng 2 ptr =   117 (+4), offset = 46 (+3)
Blk    15 pointers, Eng 1: P1 = 1,731,763 (+356),  P2 = 1,263,854 (+16),   P3 = 778,554   (+2),    P4 = 181,556   (+578),  Eng 2 ptr =   124 (+7), offset = 47 (+1)
Blk    16 pointers, Eng 1: P1 = 1,731,844 (+81),   P2 = 1,264,016 (+162),  P3 = 778,556   (+2),    P4 = 181,531   (-25),   Eng 2 ptr =   130 (+6), offset = 49 (+2)
Blk    17 pointers, Eng 1: P1 = 1,731,990 (+146),  P2 = 1,264,027 (+11),   P3 = 779,252   (+696),  P4 = 182,082   (+551),  Eng 2 ptr =   137 (+7), offset = 50 (+1)
Blk    18 pointers, Eng 1: P1 = 1,732,008 (+18),   P2 = 1,264,051 (+24),   P3 = 780,044   (+792),  P4 = 181,166   (-916),  Eng 2 ptr =   138 (+1), offset = 51 (+1)
Blk    19 pointers, Eng 1: P1 = 1,732,366 (+358),  P2 = 1,264,462 (+411),  P3 = 780,056   (+12),   P4 = 181,247   (+81),   Eng 2 ptr =   146 (+8), offset = 53 (+2)
Blk    20 pointers, Eng 1: P1 = 1,732,762 (+396),  P2 = 1,264,466 (+4),    P3 = 780,488   (+432),  P4 = 185,262   (+4015), Eng 2 ptr =   150 (+4), offset = 56 (+3)
Blk    21 pointers, Eng 1: P1 = 1,732,904 (+142),  P2 = 1,264,609 (+143),  P3 = 780,810   (+322),  P4 = 184,423   (-839),  Eng 2 ptr =   158 (+8), offset = 58 (+2)
Blk    22 pointers, Eng 1: P1 = 1,733,300 (+396),  P2 = 1,264,648 (+39),   P3 = 781,266   (+456),  P4 = 185,841   (+1418), Eng 2 ptr =   163 (+5), offset = 60 (+2)
Blk    23 pointers, Eng 1: P1 = 1,734,916 (+1616), P2 = 1,265,035 (+387),  P3 = 781,666   (+400),  P4 = 186,742   (+901),  Eng 2 ptr =   164 (+1), offset = 63 (+3)
Blk    24 pointers, Eng 1: P1 = 1,734,932 (+16),   P2 = 1,265,041 (+6),    P3 = 782,568   (+902),  P4 = 181,385   (-5357), Eng 2 ptr =   166 (+2), offset = 66 (+3)
Blk    25 pointers, Eng 1: P1 = 1,736,394 (+1462), P2 = 1,265,050 (+9),    P3 = 782,945   (+377),  P4 = 180,697   (-688),  Eng 2 ptr =   171 (+5), offset = 67 (+1)
Blk    26 pointers, Eng 1: P1 = 1,736,606 (+212),  P2 = 1,265,922 (+872),  P3 = 783,311   (+366),  P4 = 187,833   (+7136), Eng 2 ptr =   173 (+2), offset = 71 (+4)
Blk    27 pointers, Eng 1: P1 = 1,737,474 (+868),  P2 = 1,266,524 (+602),  P3 = 783,325   (+14),   P4 = 140,153   (-47680), Eng 2 ptr =   177 (+4), offset = 75 (+4)
Blk    28 pointers, Eng 1: P1 = 1,737,568 (+94),   P2 = 1,266,550 (+26),   P3 = 783,796   (+471),  P4 = 140,631   (+478),  Eng 2 ptr =   184 (+7), offset = 77 (+2)
Blk    29 pointers, Eng 1: P1 = 1,737,581 (+13),   P2 = 1,267,258 (+708),  P3 = 783,996   (+200),  P4 = 140,248   (-383),  Eng 2 ptr =   188 (+4), offset = 79 (+2)
Blk    30 pointers, Eng 1: P1 = 1,737,587 (+6),    P2 = 1,267,302 (+44),   P3 = 784,330   (+334),  P4 = 139,854   (-394),  Eng 2 ptr =   192 (+4), offset = 81 (+2)
Blk    31 pointers, Eng 1: P1 = 1,738,131 (+544),  P2 = 1,267,334 (+32),   P3 = 784,760   (+430),  P4 = 142,075   (+2221), Eng 2 ptr =   194 (+2), offset = 84 (+3)
Blk    32 pointers, Eng 1: P1 = 1,738,147 (+16),   P2 = 1,267,562 (+228),  P3 = 785,448   (+688),  P4 = 142,974   (+899),  Eng 2 ptr =   200 (+6), offset = 87 (+3)
Blk    33 pointers, Eng 1: P1 = 1,738,291 (+144),  P2 = 1,268,012 (+450),  P3 = 785,910   (+462),  P4 = 139,899   (-3075), Eng 2 ptr =   201 (+1), offset = 91 (+4)
Blk    34 pointers, Eng 1: P1 = 1,738,496 (+205),  P2 = 1,268,134 (+122),  P3 = 786,106   (+196),  P4 = 139,400   (-499),  Eng 2 ptr =   202 (+1), offset = 94 (+3)
Blk    35 pointers, Eng 1: P1 = 1,739,388 (+892),  P2 = 1,268,328 (+194),  P3 = 786,129   (+23),   P4 = 142,935   (+3535), Eng 2 ptr =   205 (+3), offset = 95 (+1)
Blk    36 pointers, Eng 1: P1 = 1,740,306 (+918),  P2 = 1,268,423 (+95),   P3 = 786,588   (+459),  P4 = 382,180   (+239245), Eng 2 ptr =   207 (+2), offset = 98 (+3)
Blk    37 pointers, Eng 1: P1 = 1,740,313 (+7),    P2 = 1,268,658 (+235),  P3 = 786,850   (+262),  P4 = 382,143   (-37),   Eng 2 ptr =   211 (+4), offset =101 (+3)
Blk    38 pointers, Eng 1: P1 = 1,740,722 (+409),  P2 = 1,268,662 (+4),    P3 = 787,740   (+890),  P4 = 381,348   (-795),  Eng 2 ptr =   219 (+8), offset =103 (+2)
Blk    39 pointers, Eng 1: P1 = 1,740,853 (+131),  P2 = 1,268,972 (+310),  P3 = 788,388   (+648),  P4 = 379,849   (-1499), Eng 2 ptr =   223 (+4), offset =107 (+4)
Blk    40 pointers, Eng 1: P1 = 1,741,041 (+188),  P2 = 1,269,045 (+73),   P3 = 788,460   (+72),   P4 = 379,556   (-293),  Eng 2 ptr =   227 (+4), offset =111 (+4)
Blk    41 pointers, Eng 1: P1 = 1,741,701 (+660),  P2 = 1,269,341 (+296),  P3 = 788,790   (+330),  P4 = 377,973   (-1583), Eng 2 ptr =   233 (+6), offset =114 (+3)
Blk    42 pointers, Eng 1: P1 = 1,742,565 (+864),  P2 = 1,269,365 (+24),   P3 = 789,210   (+420),  P4 = 377,556   (-417),  Eng 2 ptr =   235 (+2), offset =117 (+3)
Blk    43 pointers, Eng 1: P1 = 1,742,589 (+24),   P2 = 1,269,437 (+72),   P3 = 789,222   (+12),   P4 = 377,391   (-165),  Eng 2 ptr =   241 (+6), offset =120 (+3)
Blk    44 pointers, Eng 1: P1 = 1,742,915 (+326),  P2 = 1,269,671 (+234),  P3 = 789,455   (+233),  P4 = 380,078   (+2687), Eng 2 ptr =   249 (+8), offset =124 (+4)
Blk    45 pointers, Eng 1: P1 = 1,743,429 (+514),  P2 = 1,270,671 (+1000), P3 = 789,723   (+268),  P4 = 390,550   (+10472), Eng 2 ptr =   250 (+1), offset =125 (+1)
Blk    46 pointers, Eng 1: P1 = 1,743,554 (+125),  P2 = 1,270,697 (+26),   P3 = 789,973   (+250),  P4 = 390,206   (-344),  Eng 2 ptr =   255 (+5), offset =127 (+2)
Blk    47 pointers, Eng 1: P1 = 1,743,840 (+286),  P2 = 1,270,743 (+46),   P3 = 790,831   (+858),  P4 = 387,231   (-2975), Eng 2 ptr =   262 (+7), offset =129 (+2)
Blk    48 pointers, Eng 1: P1 = 1,744,342 (+502),  P2 = 1,270,751 (+8),    P3 = 791,345   (+514),  P4 = 388,279   (+1048), Eng 2 ptr =   269 (+7), offset =130 (+1)
Blk    49 pointers, Eng 1: P1 = 1,744,517 (+175),  P2 = 1,270,769 (+18),   P3 = 791,349   (+4),    P4 = 389,079   (+800),  Eng 2 ptr =   277 (+8), offset =131 (+1)
Blk    50 pointers, Eng 1: P1 = 1,745,415 (+898),  P2 = 1,270,771 (+2),    P3 = 791,350   (+1),    P4 = 381,784   (-7295), Eng 2 ptr =   285 (+8), offset =135 (+4)
Blk    51 pointers, Eng 1: P1 = 1,745,427 (+12),   P2 = 1,271,263 (+492),  P3 = 791,445   (+95),   P4 = 382,449   (+665),  Eng 2 ptr =   288 (+3), offset =136 (+1)
Blk    52 pointers, Eng 1: P1 = 1,746,157 (+730),  P2 = 1,271,487 (+224),  P3 = 791,479   (+34),   P4 = 381,003   (-1446), Eng 2 ptr =   295 (+7), offset =137 (+1)
Blk    53 pointers, Eng 1: P1 = 1,746,158 (+1),    P2 = 1,271,585 (+98),   P3 = 791,483   (+4),    P4 = 381,399   (+396),  Eng 2 ptr =   297 (+2), offset =138 (+1)
Blk    54 pointers, Eng 1: P1 = 1,746,252 (+94),   P2 = 1,271,907 (+322),  P3 = 791,504   (+21),   P4 = 384,861   (+3462), Eng 2 ptr =   300 (+3), offset =142 (+4)
Blk    55 pointers, Eng 1: P1 = 1,746,771 (+519),  P2 = 1,271,921 (+14),   P3 = 792,022   (+518),  P4 = 383,824   (-1037), Eng 2 ptr =   308 (+8), offset =144 (+2)
Blk    56 pointers, Eng 1: P1 = 1,746,975 (+204),  P2 = 1,271,977 (+56),   P3 = 792,411   (+389),  P4 = 382,538   (-1286), Eng 2 ptr =   309 (+1), offset =146 (+2)
Blk    57 pointers, Eng 1: P1 = 1,747,672 (+697),  P2 = 1,272,125 (+148),  P3 = 792,833   (+422),  P4 = 383,817   (+1279), Eng 2 ptr =   313 (+4), offset =148 (+2)
Blk    58 pointers, Eng 1: P1 = 1,747,823 (+151),  P2 = 1,273,047 (+922),  P3 = 793,145   (+312),  P4 = 384,088   (+271),  Eng 2 ptr =   321 (+8), offset =152 (+4)
Blk    59 pointers, Eng 1: P1 = 1,748,659 (+836),  P2 = 1,273,070 (+23),   P3 = 793,151   (+6),    P4 = 383,414   (-674),  Eng 2 ptr =   328 (+7), offset =156 (+4)
Blk    60 pointers, Eng 1: P1 = 1,748,673 (+14),   P2 = 1,273,080 (+10),   P3 = 793,166   (+15),   P4 = 383,390   (-24),   Eng 2 ptr =   329 (+1), offset =160 (+4)
```

```
Blk     61 pointers, Eng 1: P1 = 1,749,105 (+432),  P2 = 1,273,376 (+296),  P3 = 793,690   (+524),  P4 = 377,826   (-5564),   Eng 2 ptr =   331 (+2), Offset =162 (+2)
Blk     62 pointers, Eng 1: P1 = 1,749,362 (+257),  P2 = 1,273,378 (+2),    P3 = 793,704   (+14),   P4 = 377,556   (-270),    Eng 2 ptr =   335 (+4), Offset =163 (+1)
Blk     63 pointers, Eng 1: P1 = 1,750,712 (+1350), P2 = 1,273,469 (+91),   P3 = 793,732   (+28),   P4 = 378,294   (+738),    Eng 2 ptr =   339 (+4), Offset =164 (+1)
Blk     64 pointers, Eng 1: P1 = 1,751,054 (+342),  P2 = 1,273,612 (+143),  P3 = 793,752   (+20),   P4 = 379,497   (+1203),   Eng 2 ptr =   340 (+1), Offset =167 (+3)
Blk     65 pointers, Eng 1: P1 = 1,751,105 (+51),   P2 = 1,273,616 (+4),    P3 = 793,784   (+32),   P4 = 379,415   (-82),     Eng 2 ptr =   342 (+2), Offset =168 (+1)
Blk     66 pointers, Eng 1: P1 = 1,751,513 (+408),  P2 = 1,273,976 (+360),  P3 = 794,074   (+290),  P4 = 382,344   (+2929),   Eng 2 ptr =   346 (+4), Offset =169 (+1)
Blk     67 pointers, Eng 1: P1 = 1,753,153 (+1640), P2 = 1,274,198 (+222),  P3 = 794,297   (+223),  P4 = 372,313   (-10031),  Eng 2 ptr =   348 (+2), Offset =171 (+2)
Blk     68 pointers, Eng 1: P1 = 1,753,349 (+196),  P2 = 1,274,633 (+435),  P3 = 794,397   (+100),  P4 = 371,934   (-379),    Eng 2 ptr =   354 (+6), Offset =173 (+2)
Blk     69 pointers, Eng 1: P1 = 1,753,358 (+9),    P2 = 1,275,337 (+704),  P3 = 794,541   (+144),  P4 = 371,378   (-556),    Eng 2 ptr =   360 (+6), Offset =176 (+3)
Blk     70 pointers, Eng 1: P1 = 1,753,372 (+14),   P2 = 1,275,713 (+376),  P3 = 794,683   (+142),  P4 = 366,500   (-4878),   Eng 2 ptr =   368 (+8), Offset =178 (+2)
Blk     71 pointers, Eng 1: P1 = 1,753,621 (+249),  P2 = 1,276,055 (+342),  P3 = 794,861   (+178),  P4 = 367,534   (+1034),   Eng 2 ptr =   373 (+5), Offset =180 (+2)
Blk     72 pointers, Eng 1: P1 = 1,753,622 (+1),    P2 = 1,276,321 (+266),  P3 = 794,863   (+2),    P4 = 367,258   (-276),    Eng 2 ptr =   375 (+2), Offset =181 (+1)
Blk     73 pointers, Eng 1: P1 = 1,753,828 (+206),  P2 = 1,276,341 (+20),   P3 = 795,673   (+810),  P4 = 368,259   (+1001),   Eng 2 ptr =   380 (+5), Offset =183 (+2)
Blk     74 pointers, Eng 1: P1 = 1,754,202 (+374),  P2 = 1,277,235 (+894),  P3 = 795,693   (+20),   P4 = 367,742   (-517),    Eng 2 ptr =   385 (+5), Offset =187 (+4)
Blk     75 pointers, Eng 1: P1 = 1,754,314 (+112),  P2 = 1,278,071 (+836),  P3 = 795,907   (+214),  P4 = 352,388   (-15354),  Eng 2 ptr =   390 (+5), Offset =188 (+1)
Blk     76 pointers, Eng 1: P1 = 1,755,027 (+713),  P2 = 1,278,571 (+500),  P3 = 796,379   (+472),  P4 = 352,872   (+484),    Eng 2 ptr =   395 (+5), Offset =192 (+4)
Blk     77 pointers, Eng 1: P1 = 1,755,045 (+18),   P2 = 1,278,818 (+247),  P3 = 796,889   (+510),  P4 = 355,664   (+2792),   Eng 2 ptr =   399 (+4), Offset =193 (+1)
Blk     78 pointers, Eng 1: P1 = 1,755,807 (+762),  P2 = 1,278,825 (+7),    P3 = 796,903   (+14),   P4 = 352,321   (-3343),   Eng 2 ptr =   404 (+5), Offset =196 (+3)
Blk     79 pointers, Eng 1: P1 = 1,756,729 (+922),  P2 = 1,279,303 (+478),  P3 = 797,131   (+228),  P4 = 353,258   (+937),    Eng 2 ptr =   409 (+5), Offset =198 (+2)
Blk     80 pointers, Eng 1: P1 = 1,757,047 (+318),  P2 = 1,280,015 (+712),  P3 = 797,290   (+159),  P4 = 355,450   (+2192),   Eng 2 ptr =   417 (+8), Offset =201 (+3)
Blk     81 pointers, Eng 1: P1 = 1,757,049 (+2),    P2 = 1,280,441 (+426),  P3 = 797,751   (+461),  P4 = 355,224   (-226),    Eng 2 ptr =   419 (+2), Offset =204 (+3)
Blk     82 pointers, Eng 1: P1 = 1,757,084 (+35),   P2 = 1,280,497 (+56),   P3 = 798,217   (+466),  P4 = 354,572   (-652),    Eng 2 ptr =   421 (+2), Offset =205 (+1)
Blk     83 pointers, Eng 1: P1 = 1,757,769 (+685),  P2 = 1,280,912 (+415),  P3 = 798,857   (+640),  P4 = 354,347   (-225),    Eng 2 ptr =   427 (+6), Offset =208 (+3)
Blk     84 pointers, Eng 1: P1 = 1,757,797 (+28),   P2 = 1,281,256 (+344),  P3 = 799,545   (+688),  P4 = 355,540   (+1193),   Eng 2 ptr =   433 (+6), Offset =209 (+1)
Blk     85 pointers, Eng 1: P1 = 1,757,949 (+152),  P2 = 1,281,609 (+353),  P3 = 799,845   (+300),  P4 = 354,749   (-791),    Eng 2 ptr =   440 (+7), Offset =212 (+3)
Blk     86 pointers, Eng 1: P1 = 1,758,382 (+433),  P2 = 1,281,659 (+50),   P3 = 799,865   (+20),   P4 = 356,293   (+1544),   Eng 2 ptr =   446 (+6), Offset =215 (+3)
Blk     87 pointers, Eng 1: P1 = 1,758,387 (+5),    P2 = 1,282,018 (+359),  P3 = 799,869   (+4),    P4 = 355,892   (-401),    Eng 2 ptr =   450 (+4), Offset =218 (+3)
Blk     88 pointers, Eng 1: P1 = 1,758,636 (+249),  P2 = 1,282,247 (+229),  P3 = 799,929   (+60),   P4 = 356,550   (+658),    Eng 2 ptr =   456 (+6), Offset =220 (+2)
Blk     89 pointers, Eng 1: P1 = 1,758,640 (+4),    P2 = 1,282,265 (+18),   P3 = 800,171   (+242),  P4 = 356,820   (+270),    Eng 2 ptr =   457 (+1), Offset =223 (+3)
Blk     90 pointers, Eng 1: P1 = 1,758,642 (+2),    P2 = 1,282,281 (+16),   P3 = 800,553   (+382),  P4 = 357,214   (+394),    Eng 2 ptr =   460 (+3), Offset =224 (+1)
Blk     91 pointers, Eng 1: P1 = 1,758,769 (+127),  P2 = 1,282,407 (+126),  P3 = 800,857   (+304),  P4 = 359,996   (+2782),   Eng 2 ptr =   466 (+6), Offset =225 (+1)
Blk     92 pointers, Eng 1: P1 = 1,759,031 (+262),  P2 = 1,282,511 (+104),  P3 = 800,951   (+94),   P4 = 360,304   (+308),    Eng 2 ptr =   474 (+8), Offset =229 (+4)
Blk     93 pointers, Eng 1: P1 = 1,759,187 (+156),  P2 = 1,282,517 (+6),    P3 = 801,139   (+188),  P4 = 360,060   (-244),    Eng 2 ptr =   480 (+6), Offset =233 (+4)
Blk     94 pointers, Eng 1: P1 = 1,759,406 (+219),  P2 = 1,282,531 (+14),   P3 = 801,265   (+126),  P4 = 356,785   (-3275),   Eng 2 ptr =   481 (+1), Offset =236 (+3)
Blk     95 pointers, Eng 1: P1 = 1,759,420 (+14),   P2 = 1,283,343 (+812),  P3 = 801,295   (+30),   P4 = 358,070   (+1285),   Eng 2 ptr =   485 (+4), Offset =239 (+3)
Blk     96 pointers, Eng 1: P1 = 1,759,427 (+7),    P2 = 1,283,357 (+14),   P3 = 801,310   (+15),   P4 = 358,103   (+33),     Eng 2 ptr =   486 (+1), Offset =241 (+2)
Blk     97 pointers, Eng 1: P1 = 1,759,475 (+48),   P2 = 1,283,463 (+106),  P3 = 801,568   (+258),  P4 = 358,222   (+119),    Eng 2 ptr =   494 (+8), Offset =244 (+3)
Blk     98 pointers, Eng 1: P1 = 1,759,951 (+476),  P2 = 1,283,581 (+118),  P3 = 801,878   (+310),  P4 = 356,968   (-1254),   Eng 2 ptr =   500 (+6), Offset =248 (+4)
Blk     99 pointers, Eng 1: P1 = 1,760,735 (+784),  P2 = 1,283,669 (+88),   P3 = 802,314   (+436),  P4 = 357,505   (+537),    Eng 2 ptr =   506 (+6), Offset =251 (+3)
Blk    100 pointers, Eng 1: P1 = 1,760,743 (+8),    P2 = 1,284,215 (+546),  P3 = 802,452   (+138),  P4 = 358,907   (+1402),   Eng 2 ptr =   514 (+8), Offset =253 (+2)
Blk    101 pointers, Eng 1: P1 = 1,760,865 (+122),  P2 = 1,284,843 (+628),  P3 = 802,467   (+15),   P4 = 358,433   (-474),    Eng 2 ptr =   521 (+7), Offset =  1 (+4)
Blk    102 pointers, Eng 1: P1 = 1,760,892 (+27),   P2 = 1,284,997 (+154),  P3 = 802,473   (+6),    P4 = 358,747   (+314),    Eng 2 ptr =   527 (+6), Offset =  4 (+3)
Blk    103 pointers, Eng 1: P1 = 1,760,985 (+93),   P2 = 1,285,165 (+168),  P3 = 802,483   (+10),   P4 = 360,020   (+1273),   Eng 2 ptr =   533 (+6), Offset =  6 (+2)
Blk    104 pointers, Eng 1: P1 = 1,761,621 (+636),  P2 = 1,285,188 (+23),   P3 = 802,489   (+6),    P4 = 344,508   (-15512),  Eng 2 ptr =   540 (+7), Offset =  8 (+2)
Blk    105 pointers, Eng 1: P1 = 1,761,835 (+214),  P2 = 1,285,537 (+349),  P3 = 802,945   (+456),  P4 = 343,330   (-1178),   Eng 2 ptr =   546 (+6), Offset = 11 (+3)
Blk    106 pointers, Eng 1: P1 = 1,761,851 (+16),   P2 = 1,286,141 (+604),  P3 = 803,237   (+292),  P4 = 343,625   (+295),    Eng 2 ptr =   549 (+3), Offset = 15 (+4)
Blk    107 pointers, Eng 1: P1 = 1,762,767 (+916),  P2 = 1,286,162 (+21),   P3 = 803,373   (+136),  P4 = 329,162   (-14463),  Eng 2 ptr =   552 (+3), Offset = 18 (+3)
Blk    108 pointers, Eng 1: P1 = 1,763,385 (+618),  P2 = 1,286,471 (+309),  P3 = 803,841   (+468),  P4 = 331,589   (+2427),   Eng 2 ptr =   553 (+1), Offset = 19 (+1)
Blk    109 pointers, Eng 1: P1 = 1,763,549 (+164),  P2 = 1,286,472 (+1),    P3 = 804,609   (+768),  P4 = 330,922   (-667),    Eng 2 ptr =   554 (+1), Offset = 20 (+1)
Blk    110 pointers, Eng 1: P1 = 1,763,577 (+28),   P2 = 1,286,473 (+1),    P3 = 805,083   (+474),  P4 = 328,525   (-2397),   Eng 2 ptr =   561 (+7), Offset = 23 (+3)
Blk    111 pointers, Eng 1: P1 = 1,763,602 (+25),   P2 = 1,286,485 (+12),   P3 = 805,087   (+4),    P4 = 328,368   (-157),    Eng 2 ptr =   563 (+2), Offset = 27 (+4)
Blk    112 pointers, Eng 1: P1 = 1,763,603 (+1),    P2 = 1,287,403 (+918),  P3 = 805,091   (+4),    P4 = 329,522   (+1154),   Eng 2 ptr =   567 (+4), Offset = 30 (+3)
Blk    113 pointers, Eng 1: P1 = 1,763,663 (+60),   P2 = 1,287,649 (+246),  P3 = 805,509   (+418),  P4 = 328,759   (-763),    Eng 2 ptr =   573 (+6), Offset = 33 (+3)
Blk    114 pointers, Eng 1: P1 = 1,764,075 (+412),  P2 = 1,287,962 (+313),  P3 = 805,609   (+100),  P4 = 329,085   (+326),    Eng 2 ptr =   577 (+4), Offset = 36 (+3)
Blk    115 pointers, Eng 1: P1 = 1,764,983 (+908),  P2 = 1,287,974 (+12),   P3 = 805,907   (+298),  P4 = 329,518   (+433),    Eng 2 ptr =   585 (+8), Offset = 37 (+1)
Blk    116 pointers, Eng 1: P1 = 1,765,241 (+258),  P2 = 1,287,988 (+14),   P3 = 805,955   (+48),   P4 = 329,317   (-201),    Eng 2 ptr =   589 (+4), Offset = 38 (+1)
Blk    117 pointers, Eng 1: P1 = 1,765,267 (+26),   P2 = 1,288,393 (+405),  P3 = 806,135   (+180),  P4 = 330,196   (+879),    Eng 2 ptr =   593 (+4), Offset = 40 (+2)
Blk    118 pointers, Eng 1: P1 = 1,765,365 (+98),   P2 = 1,288,673 (+280),  P3 = 806,227   (+92),   P4 = 330,035   (-161),    Eng 2 ptr =   600 (+7), Offset = 44 (+4)
Blk    119 pointers, Eng 1: P1 = 1,765,527 (+162),  P2 = 1,289,423 (+750),  P3 = 806,240   (+13),   P4 = 332,618   (+2583),   Eng 2 ptr =   605 (+5), Offset = 47 (+3)
Blk    120 pointers, Eng 1: P1 = 1,765,623 (+96),   P2 = 1,289,757 (+334),  P3 = 806,246   (+6),    P4 = 332,259   (-359),    Eng 2 ptr =   607 (+2), Offset = 48 (+1)
```

```
Blk      1 pointers, Eng 1: P1 = 774,161           P2 = 1,725,418          P3 = 1,260,439          P4 = 178,267             Eng 2 ptr =    49,       offset =  6
Blk      2 pointers, Eng 1: P1 = 774,403   (+242),  P2 = 1,725,786 (+368),  P3 = 1,260,469 (+30),   P4 = 180,177   (+1910),  Eng 2 ptr =    53 (+4), offset =  8 (+2)
Blk      3 pointers, Eng 1: P1 = 775,359   (+956),  P2 = 1,725,796 (+10),   P3 = 1,261,426 (+957),  P4 = 179,864   (-313),   Eng 2 ptr =    59 (+6), offset = 10 (+2)
Blk      4 pointers, Eng 1: P1 = 776,207   (+848),  P2 = 1,726,390 (+594),  P3 = 1,262,340 (+914),  P4 = 183,538   (+3674),  Eng 2 ptr =    67 (+8), offset = 12 (+2)
Blk      5 pointers, Eng 1: P1 = 776,221   (+14),   P2 = 1,726,413 (+23),   P3 = 1,262,362 (+22),   P4 = 183,427   (-111),   Eng 2 ptr =    68 (+1), offset = 13 (+1)
Blk      6 pointers, Eng 1: P1 = 777,885   (+1664), P2 = 1,727,363 (+950),  P3 = 1,262,837 (+475),  P4 = 180,669   (-2758),  Eng 2 ptr =    70 (+2), offset = 16 (+3)
Blk      7 pointers, Eng 1: P1 = 778,105   (+220),  P2 = 1,727,367 (+4),    P3 = 1,263,275 (+438),  P4 = 180,745   (+76),    Eng 2 ptr =    78 (+8), offset = 18 (+2)
Blk      8 pointers, Eng 1: P1 = 778,107   (+2),    P2 = 1,727,423 (+56),   P3 = 1,263,328 (+53),   P4 = 180,836   (+91),    Eng 2 ptr =    86 (+8), offset = 22 (+4)
Blk      9 pointers, Eng 1: P1 = 778,119   (+12),   P2 = 1,727,857 (+434),  P3 = 1,263,496 (+168),  P4 = 181,560   (+724),   Eng 2 ptr =    94 (+8), offset = 24 (+2)
Blk     10 pointers, Eng 1: P1 = 778,261   (+142),  P2 = 1,727,870 (+13),   P3 = 1,264,162 (+666),  P4 = 194,354   (+12794), Eng 2 ptr =    96 (+2), offset = 27 (+3)
Blk     11 pointers, Eng 1: P1 = 778,272   (+11),   P2 = 1,727,886 (+16),   P3 = 1,264,504 (+342),  P4 = 194,216   (-138),   Eng 2 ptr =    97 (+1), offset = 31 (+4)
Blk     12 pointers, Eng 1: P1 = 778,343   (+71),   P2 = 1,728,532 (+646),  P3 = 1,265,388 (+884),  P4 = 184,023   (-10193), Eng 2 ptr =   104 (+7), offset = 32 (+1)
Blk     13 pointers, Eng 1: P1 = 778,444   (+101),  P2 = 1,728,540 (+8),    P3 = 1,266,112 (+724),  P4 = 184,667   (+644),   Eng 2 ptr =   106 (+2), offset = 33 (+1)
Blk     14 pointers, Eng 1: P1 = 778,474   (+30),   P2 = 1,729,154 (+614),  P3 = 1,266,133 (+21),   P4 = 185,332   (+665),   Eng 2 ptr =   109 (+3), offset = 36 (+3)
Blk     15 pointers, Eng 1: P1 = 779,470   (+996),  P2 = 1,729,362 (+208),  P3 = 1,266,231 (+98),   P4 = 185,854   (+522),   Eng 2 ptr =   114 (+5), offset = 39 (+3)
Blk     16 pointers, Eng 1: P1 = 779,543   (+73),   P2 = 1,729,434 (+72),   P3 = 1,266,302 (+71),   P4 = 185,516   (-338),   Eng 2 ptr =   117 (+3), offset = 42 (+3)
Blk     17 pointers, Eng 1: P1 = 779,552   (+9),    P2 = 1,730,364 (+930),  P3 = 1,266,320 (+18),   P4 = 184,536   (-980),   Eng 2 ptr =   120 (+3), offset = 44 (+2)
Blk     18 pointers, Eng 1: P1 = 779,730   (+178),  P2 = 1,731,078 (+714),  P3 = 1,266,489 (+169),  P4 = 187,579   (+3043),  Eng 2 ptr =   123 (+3), offset = 45 (+1)
Blk     19 pointers, Eng 1: P1 = 779,986   (+256),  P2 = 1,731,105 (+27),   P3 = 1,266,504 (+15),   P4 = 188,392   (+813),   Eng 2 ptr =   125 (+2), offset = 46 (+1)
Blk     20 pointers, Eng 1: P1 = 780,483   (+497),  P2 = 1,731,178 (+73),   P3 = 1,266,810 (+306),  P4 = 186,012   (-2380),  Eng 2 ptr =   126 (+1), offset = 49 (+3)
Blk     21 pointers, Eng 1: P1 = 781,185   (+702),  P2 = 1,731,356 (+178),  P3 = 1,266,812 (+2),    P4 = 185,424   (-588),   Eng 2 ptr =   133 (+7), offset = 52 (+3)
Blk     22 pointers, Eng 1: P1 = 781,191   (+6),    P2 = 1,731,498 (+142),  P3 = 1,266,952 (+140),  P4 = 185,758   (+334),   Eng 2 ptr =   140 (+7), offset = 55 (+3)
Blk     23 pointers, Eng 1: P1 = 781,195   (+4),    P2 = 1,731,672 (+174),  P3 = 1,267,558 (+606),  P4 = 184,321   (-1437),  Eng 2 ptr =   141 (+1), offset = 57 (+2)
Blk     24 pointers, Eng 1: P1 = 781,942   (+747),  P2 = 1,732,456 (+784),  P3 = 1,267,582 (+24),   P4 = 186,062   (+1741),  Eng 2 ptr =   149 (+8), offset = 59 (+2)
Blk     25 pointers, Eng 1: P1 = 782,198   (+256),  P2 = 1,732,580 (+124),  P3 = 1,267,591 (+9),    P4 = 186,294   (+232),   Eng 2 ptr =   157 (+8), offset = 62 (+3)
Blk     26 pointers, Eng 1: P1 = 782,606   (+408),  P2 = 1,732,610 (+30),   P3 = 1,267,879 (+288),  P4 = 186,741   (+447),   Eng 2 ptr =   159 (+2), offset = 65 (+3)
Blk     27 pointers, Eng 1: P1 = 783,256   (+650),  P2 = 1,732,664 (+54),   P3 = 1,267,895 (+16),   P4 = 187,379   (+638),   Eng 2 ptr =   160 (+1), offset = 68 (+3)
Blk     28 pointers, Eng 1: P1 = 783,656   (+400),  P2 = 1,733,052 (+388),  P3 = 1,268,055 (+160),  P4 = 187,694   (+315),   Eng 2 ptr =   168 (+8), offset = 69 (+1)
Blk     29 pointers, Eng 1: P1 = 783,684   (+28),   P2 = 1,733,316 (+264),  P3 = 1,268,082 (+27),   P4 = 187,925   (+231),   Eng 2 ptr =   174 (+6), offset = 73 (+4)
Blk     30 pointers, Eng 1: P1 = 784,856   (+1172), P2 = 1,733,331 (+15),   P3 = 1,268,694 (+612),  P4 = 184,353   (-3572),  Eng 2 ptr =   177 (+3), offset = 77 (+4)
Blk     31 pointers, Eng 1: P1 = 785,094   (+238),  P2 = 1,733,647 (+316),  P3 = 1,269,564 (+870),  P4 = 184,584   (+231),   Eng 2 ptr =   178 (+1), offset = 79 (+2)
Blk     32 pointers, Eng 1: P1 = 785,116   (+22),   P2 = 1,733,671 (+24),   P3 = 1,269,587 (+23),   P4 = 184,654   (+70),    Eng 2 ptr =   183 (+5), offset = 81 (+2)
Blk     33 pointers, Eng 1: P1 = 785,248   (+132),  P2 = 1,733,699 (+28),   P3 = 1,270,307 (+720),  P4 = 191,977   (+7323),  Eng 2 ptr =   187 (+4), offset = 82 (+1)
Blk     34 pointers, Eng 1: P1 = 785,476   (+228),  P2 = 1,734,021 (+322),  P3 = 1,271,051 (+744),  P4 = 191,522   (-455),   Eng 2 ptr =   188 (+1), offset = 84 (+2)
Blk     35 pointers, Eng 1: P1 = 785,480   (+4),    P2 = 1,734,825 (+804),  P3 = 1,271,939 (+888),  P4 = 191,729   (+207),   Eng 2 ptr =   196 (+8), offset = 87 (+3)
Blk     36 pointers, Eng 1: P1 = 785,484   (+4),    P2 = 1,734,903 (+78),   P3 = 1,272,957 (+1018), P4 = 190,550   (-1179),  Eng 2 ptr =   201 (+5), offset = 89 (+2)
Blk     37 pointers, Eng 1: P1 = 785,588   (+104),  P2 = 1,734,965 (+62),   P3 = 1,273,891 (+934),  P4 = 193,846   (+3296),  Eng 2 ptr =   206 (+5), offset = 90 (+1)
Blk     38 pointers, Eng 1: P1 = 785,624   (+36),   P2 = 1,735,879 (+914),  P3 = 1,274,311 (+420),  P4 = 192,848   (-998),   Eng 2 ptr =   214 (+8), offset = 94 (+4)
Blk     39 pointers, Eng 1: P1 = 785,631   (+7),    P2 = 1,736,334 (+455),  P3 = 1,274,329 (+18),   P4 = 193,343   (+495),   Eng 2 ptr =   215 (+1), offset = 96 (+2)
Blk     40 pointers, Eng 1: P1 = 785,841   (+210),  P2 = 1,736,336 (+2),    P3 = 1,274,333 (+4),    P4 = 193,088   (-255),   Eng 2 ptr =   223 (+8), offset = 99 (+3)
Blk     41 pointers, Eng 1: P1 = 785,851   (+10),   P2 = 1,736,345 (+9),    P3 = 1,274,540 (+207),  P4 = 192,778   (-310),   Eng 2 ptr =   225 (+2), offset =101 (+2)
Blk     42 pointers, Eng 1: P1 = 786,047   (+196),  P2 = 1,736,839 (+494),  P3 = 1,274,721 (+181),  P4 = 134,423   (-58355), Eng 2 ptr =   232 (+7), offset =102 (+1)
Blk     43 pointers, Eng 1: P1 = 786,101   (+54),   P2 = 1,737,511 (+672),  P3 = 1,274,990 (+269),  P4 = 133,495   (-928),   Eng 2 ptr =   236 (+4), offset =103 (+1)
Blk     44 pointers, Eng 1: P1 = 786,112   (+11),   P2 = 1,737,965 (+454),  P3 = 1,275,199 (+209),  P4 = 135,062   (+1567),  Eng 2 ptr =   237 (+1), offset =105 (+2)
Blk     45 pointers, Eng 1: P1 = 786,577   (+465),  P2 = 1,737,987 (+22),   P3 = 1,275,405 (+206),  P4 = 389,916   (+254854), Eng 2 ptr =   239 (+2), offset =108 (+3)
Blk     46 pointers, Eng 1: P1 = 787,427   (+850),  P2 = 1,738,789 (+802),  P3 = 1,275,462 (+57),   P4 = 392,477   (+2561),  Eng 2 ptr =   243 (+4), offset =110 (+2)
Blk     47 pointers, Eng 1: P1 = 787,505   (+78),   P2 = 1,738,795 (+6),    P3 = 1,275,507 (+45),   P4 = 391,917   (-560),   Eng 2 ptr =   245 (+2), offset =113 (+3)
Blk     48 pointers, Eng 1: P1 = 787,613   (+108),  P2 = 1,739,092 (+297),  P3 = 1,275,981 (+474),  P4 = 390,410   (-1507),  Eng 2 ptr =   250 (+5), offset =116 (+3)
Blk     49 pointers, Eng 1: P1 = 787,939   (+326),  P2 = 1,739,432 (+340),  P3 = 1,276,251 (+270),  P4 = 390,810   (+400),   Eng 2 ptr =   253 (+3), offset =119 (+3)
Blk     50 pointers, Eng 1: P1 = 788,645   (+706),  P2 = 1,739,711 (+279),  P3 = 1,276,261 (+10),   P4 = 391,941   (+1131),  Eng 2 ptr =   257 (+4), offset =123 (+4)
Blk     51 pointers, Eng 1: P1 = 788,759   (+114),  P2 = 1,739,714 (+3),    P3 = 1,276,737 (+476),  P4 = 391,395   (-546),   Eng 2 ptr =   265 (+8), offset =126 (+3)
Blk     52 pointers, Eng 1: P1 = 788,843   (+84),   P2 = 1,740,024 (+310),  P3 = 1,276,739 (+2),    P4 = 393,050   (+1655),  Eng 2 ptr =   267 (+2), offset =129 (+3)
Blk     53 pointers, Eng 1: P1 = 789,191   (+348),  P2 = 1,740,032 (+8),    P3 = 1,276,755 (+16),   P4 = 392,455   (-595),   Eng 2 ptr =   273 (+6), offset =130 (+1)
Blk     54 pointers, Eng 1: P1 = 789,223   (+32),   P2 = 1,740,077 (+45),   P3 = 1,276,873 (+118),  P4 = 392,593   (+138),   Eng 2 ptr =   276 (+3), offset =134 (+4)
Blk     55 pointers, Eng 1: P1 = 789,676   (+453),  P2 = 1,740,079 (+2),    P3 = 1,276,979 (+106),  P4 = 392,237   (-356),   Eng 2 ptr =   279 (+3), offset =138 (+4)
Blk     56 pointers, Eng 1: P1 = 790,079   (+403),  P2 = 1,740,138 (+59),   P3 = 1,276,981 (+2),    P4 = 392,944   (+707),   Eng 2 ptr =   286 (+7), offset =142 (+4)
Blk     57 pointers, Eng 1: P1 = 790,083   (+4),    P2 = 1,740,574 (+436),  P3 = 1,277,797 (+816),  P4 = 393,103   (+159),   Eng 2 ptr =   293 (+7), offset =146 (+4)
Blk     58 pointers, Eng 1: P1 = 790,213   (+130),  P2 = 1,740,925 (+351),  P3 = 1,278,233 (+436),  P4 = 335,379   (-57724), Eng 2 ptr =   294 (+1), offset =148 (+2)
Blk     59 pointers, Eng 1: P1 = 790,419   (+206),  P2 = 1,740,935 (+10),   P3 = 1,278,694 (+461),  P4 = 334,912   (-467),   Eng 2 ptr =   298 (+4), offset =152 (+4)
Blk     60 pointers, Eng 1: P1 = 790,495   (+76),   P2 = 1,740,955 (+20),   P3 = 1,278,732 (+38),   P4 = 335,359   (+447),   Eng 2 ptr =   299 (+1), offset =156 (+4)
```

```
Blk    61 pointers, Eng 1: P1 = 791,019   (+524),  P2 = 1,741,785 (+830),  P3 = 1,278,994 (+262),  P4 = 329,612   (-5747),   Eng 2 ptr =   305 (+6), offset =157 (+1)
Blk    62 pointers, Eng 1: P1 = 791,461   (+442),  P2 = 1,741,817 (+32),   P3 = 1,279,004 (+10),   P4 = 329,171   (-441),    Eng 2 ptr =   307 (+2), offset =160 (+3)
Blk    63 pointers, Eng 1: P1 = 792,478   (+1017), P2 = 1,742,833 (+1016), P3 = 1,279,078 (+74),   P4 = 329,104   (-67),     Eng 2 ptr =   315 (+8), offset =162 (+2)
Blk    64 pointers, Eng 1: P1 = 792,698   (+220),  P2 = 1,743,018 (+185),  P3 = 1,279,082 (+4),    P4 = 329,052   (-52),     Eng 2 ptr =   323 (+8), offset =165 (+3)
Blk    65 pointers, Eng 1: P1 = 792,949   (+251),  P2 = 1,743,252 (+234),  P3 = 1,279,884 (+802),  P4 = 329,357   (+305),    Eng 2 ptr =   328 (+5), offset =168 (+3)
Blk    66 pointers, Eng 1: P1 = 793,628   (+679),  P2 = 1,743,732 (+480),  P3 = 1,279,898 (+14),   P4 = 327,956   (-1401),   Eng 2 ptr =   333 (+5), offset =171 (+3)
Blk    67 pointers, Eng 1: P1 = 794,206   (+578),  P2 = 1,743,741 (+9),    P3 = 1,280,202 (+304),  P4 = 330,771   (+2815),   Eng 2 ptr =   340 (+7), offset =174 (+3)
Blk    68 pointers, Eng 1: P1 = 794,267   (+61),   P2 = 1,744,166 (+425),  P3 = 1,280,228 (+26),   P4 = 330,429   (-342),    Eng 2 ptr =   342 (+2), offset =178 (+4)
Blk    69 pointers, Eng 1: P1 = 794,646   (+379),  P2 = 1,744,223 (+57),   P3 = 1,280,608 (+380),  P4 = 341,697   (+11268),  Eng 2 ptr =   348 (+6), offset =180 (+2)
Blk    70 pointers, Eng 1: P1 = 794,650   (+4),    P2 = 1,744,226 (+3),    P3 = 1,280,820 (+212),  P4 = 341,924   (+227),    Eng 2 ptr =   351 (+3), offset =183 (+3)
Blk    71 pointers, Eng 1: P1 = 794,690   (+40),   P2 = 1,744,609 (+383),  P3 = 1,280,836 (+16),   P4 = 341,048   (-876),    Eng 2 ptr =   356 (+5), offset =187 (+4)
Blk    72 pointers, Eng 1: P1 = 794,828   (+138),  P2 = 1,744,930 (+321),  P3 = 1,281,033 (+197),  P4 = 331,071   (-9977),   Eng 2 ptr =   359 (+3), offset =191 (+4)
Blk    73 pointers, Eng 1: P1 = 795,624   (+796),  P2 = 1,745,688 (+758),  P3 = 1,281,495 (+462),  P4 = 330,892   (-179),    Eng 2 ptr =   361 (+2), offset =194 (+3)
Blk    74 pointers, Eng 1: P1 = 795,652   (+28),   P2 = 1,745,709 (+21),   P3 = 1,281,530 (+35),   P4 = 330,616   (-276),    Eng 2 ptr =   366 (+5), offset =197 (+3)
Blk    75 pointers, Eng 1: P1 = 796,050   (+398),  P2 = 1,745,890 (+181),  P3 = 1,281,729 (+199),  P4 = 329,989   (-627),    Eng 2 ptr =   370 (+4), offset =198 (+1)
Blk    76 pointers, Eng 1: P1 = 796,240   (+190),  P2 = 1,746,858 (+968),  P3 = 1,281,870 (+141),  P4 = 331,528   (+1539),   Eng 2 ptr =   373 (+3), offset =201 (+3)
Blk    77 pointers, Eng 1: P1 = 796,410   (+170),  P2 = 1,746,874 (+16),   P3 = 1,282,163 (+293),  P4 = 331,909   (+381),    Eng 2 ptr =   381 (+8), offset =203 (+2)
Blk    78 pointers, Eng 1: P1 = 797,346   (+936),  P2 = 1,747,364 (+490),  P3 = 1,282,441 (+278),  P4 = 332,739   (+830),    Eng 2 ptr =   384 (+3), offset =204 (+1)
Blk    79 pointers, Eng 1: P1 = 797,376   (+30),   P2 = 1,748,032 (+668),  P3 = 1,282,470 (+29),   P4 = 333,427   (+688),    Eng 2 ptr =   389 (+5), offset =208 (+4)
Blk    80 pointers, Eng 1: P1 = 797,392   (+16),   P2 = 1,748,078 (+46),   P3 = 1,282,484 (+14),   P4 = 333,405   (-22),     Eng 2 ptr =   390 (+1), offset =209 (+1)
Blk    81 pointers, Eng 1: P1 = 797,469   (+77),   P2 = 1,748,844 (+766),  P3 = 1,282,488 (+4),    P4 = 332,946   (-459),    Eng 2 ptr =   398 (+8), offset =213 (+4)
Blk    82 pointers, Eng 1: P1 = 797,478   (+9),    P2 = 1,748,850 (+6),    P3 = 1,282,510 (+22),   P4 = 333,018   (+72),     Eng 2 ptr =   406 (+8), offset =214 (+1)
Blk    83 pointers, Eng 1: P1 = 797,494   (+16),   P2 = 1,749,149 (+299),  P3 = 1,282,754 (+244),  P4 = 329,770   (-3248),   Eng 2 ptr =   410 (+4), offset =217 (+3)
Blk    84 pointers, Eng 1: P1 = 798,399   (+905),  P2 = 1,749,160 (+11),   P3 = 1,283,108 (+354),  P4 = 330,612   (+842),    Eng 2 ptr =   411 (+1), offset =220 (+3)
Blk    85 pointers, Eng 1: P1 = 798,413   (+14),   P2 = 1,749,222 (+62),   P3 = 1,283,368 (+260),  P4 = 330,305   (-307),    Eng 2 ptr =   418 (+7), offset =224 (+4)
Blk    86 pointers, Eng 1: P1 = 798,641   (+228),  P2 = 1,749,290 (+68),   P3 = 1,284,388 (+1020), P4 = 329,465   (-840),    Eng 2 ptr =   421 (+3), offset =227 (+3)
Blk    87 pointers, Eng 1: P1 = 798,676   (+35),   P2 = 1,749,360 (+70),   P3 = 1,284,484 (+96),   P4 = 329,326   (-139),    Eng 2 ptr =   426 (+5), offset =228 (+1)
Blk    88 pointers, Eng 1: P1 = 798,684   (+8),    P2 = 1,749,860 (+500),  P3 = 1,284,935 (+451),  P4 = 329,385   (+59),     Eng 2 ptr =   433 (+7), offset =231 (+3)
Blk    89 pointers, Eng 1: P1 = 799,202   (+518),  P2 = 1,749,878 (+18),   P3 = 1,285,434 (+499),  P4 = 335,615   (+6230),   Eng 2 ptr =   441 (+8), offset =232 (+1)
Blk    90 pointers, Eng 1: P1 = 799,577   (+375),  P2 = 1,749,902 (+24),   P3 = 1,285,616 (+182),  P4 = 334,964   (-651),    Eng 2 ptr =   447 (+6), offset =236 (+4)
Blk    91 pointers, Eng 1: P1 = 799,600   (+23),   P2 = 1,750,578 (+676),  P3 = 1,285,662 (+46),   P4 = 334,450   (-514),    Eng 2 ptr =   451 (+4), offset =237 (+1)
Blk    92 pointers, Eng 1: P1 = 799,636   (+36),   P2 = 1,750,663 (+85),   P3 = 1,286,274 (+612),  P4 = 337,079   (+2629),   Eng 2 ptr =   456 (+5), offset =238 (+1)
Blk    93 pointers, Eng 1: P1 = 800,517   (+881),  P2 = 1,751,086 (+423),  P3 = 1,286,594 (+320),  P4 = 339,921   (+2842),   Eng 2 ptr =   457 (+1), offset =241 (+3)
Blk    94 pointers, Eng 1: P1 = 800,919   (+402),  P2 = 1,751,388 (+302),  P3 = 1,287,318 (+724),  P4 = 337,022   (-2899),   Eng 2 ptr =   465 (+8), offset =242 (+1)
Blk    95 pointers, Eng 1: P1 = 801,071   (+152),  P2 = 1,751,693 (+305),  P3 = 1,287,622 (+304),  P4 = 337,731   (+709),    Eng 2 ptr =   467 (+2), offset =244 (+2)
Blk    96 pointers, Eng 1: P1 = 801,196   (+125),  P2 = 1,751,963 (+270),  P3 = 1,287,823 (+201),  P4 = 337,303   (-428),    Eng 2 ptr =   471 (+4), offset =248 (+4)
Blk    97 pointers, Eng 1: P1 = 801,599   (+403),  P2 = 1,752,071 (+108),  P3 = 1,287,914 (+91),   P4 = 336,117   (-1186),   Eng 2 ptr =   475 (+4), offset =252 (+4)
Blk    98 pointers, Eng 1: P1 = 801,909   (+310),  P2 = 1,752,195 (+124),  P3 = 1,287,935 (+21),   P4 = 337,705   (+1588),   Eng 2 ptr =   479 (+4), offset =255 (+3)
Blk    99 pointers, Eng 1: P1 = 801,916   (+7),    P2 = 1,752,222 (+27),   P3 = 1,287,941 (+6),    P4 = 337,412   (-293),    Eng 2 ptr =   481 (+2), offset =  2 (+3)
Blk   100 pointers, Eng 1: P1 = 802,775   (+859),  P2 = 1,752,330 (+108),  P3 = 1,288,024 (+83),   P4 = 336,993   (-419),    Eng 2 ptr =   487 (+6), offset =  3 (+1)
Blk   101 pointers, Eng 1: P1 = 802,838   (+63),   P2 = 1,752,722 (+392),  P3 = 1,288,414 (+390),  P4 = 350,132   (+13139),  Eng 2 ptr =   488 (+1), offset =  6 (+3)
Blk   102 pointers, Eng 1: P1 = 803,322   (+484),  P2 = 1,752,737 (+15),   P3 = 1,288,899 (+485),  P4 = 349,307   (-825),    Eng 2 ptr =   491 (+3), offset =  8 (+2)
Blk   103 pointers, Eng 1: P1 = 803,650   (+328),  P2 = 1,752,839 (+102),  P3 = 1,289,555 (+656),  P4 = 348,394   (-913),    Eng 2 ptr =   496 (+5), offset = 12 (+4)
Blk   104 pointers, Eng 1: P1 = 803,755   (+105),  P2 = 1,753,239 (+400),  P3 = 1,289,703 (+148),  P4 = 339,769   (-8625),   Eng 2 ptr =   498 (+2), offset = 16 (+4)
Blk   105 pointers, Eng 1: P1 = 803,799   (+44),   P2 = 1,753,287 (+48),   P3 = 1,289,733 (+30),   P4 = 339,190   (-579),    Eng 2 ptr =   503 (+5), offset = 20 (+4)
Blk   106 pointers, Eng 1: P1 = 803,809   (+10),   P2 = 1,753,735 (+448),  P3 = 1,289,986 (+253),  P4 = 339,843   (+653),    Eng 2 ptr =   511 (+8), offset = 24 (+4)
Blk   107 pointers, Eng 1: P1 = 804,639   (+830),  P2 = 1,754,319 (+584),  P3 = 1,290,001 (+15),   P4 = 339,673   (-170),    Eng 2 ptr =   516 (+5), offset = 28 (+4)
Blk   108 pointers, Eng 1: P1 = 804,780   (+141),  P2 = 1,754,413 (+94),   P3 = 1,290,009 (+8),    P4 = 339,855   (+182),    Eng 2 ptr =   521 (+5), offset = 30 (+2)
Blk   109 pointers, Eng 1: P1 = 804,812   (+32),   P2 = 1,754,847 (+434),  P3 = 1,290,073 (+64),   P4 = 339,045   (-810),    Eng 2 ptr =   527 (+6), offset = 32 (+2)
Blk   110 pointers, Eng 1: P1 = 804,921   (+109),  P2 = 1,754,880 (+33),   P3 = 1,290,223 (+150),  P4 = 336,612   (-2433),   Eng 2 ptr =   531 (+4), offset = 33 (+1)
Blk   111 pointers, Eng 1: P1 = 805,641   (+720),  P2 = 1,754,887 (+7),    P3 = 1,290,249 (+26),   P4 = 343,603   (+6991),   Eng 2 ptr =   535 (+4), offset = 35 (+2)
Blk   112 pointers, Eng 1: P1 = 805,735   (+94),   P2 = 1,755,839 (+952),  P3 = 1,290,472 (+223),  P4 = 340,749   (-2854),   Eng 2 ptr =   538 (+3), offset = 39 (+4)
Blk   113 pointers, Eng 1: P1 = 806,703   (+968),  P2 = 1,756,367 (+528),  P3 = 1,290,762 (+290),  P4 = 340,959   (+210),    Eng 2 ptr =   541 (+3), offset = 40 (+1)
Blk   114 pointers, Eng 1: P1 = 807,023   (+320),  P2 = 1,756,601 (+234),  P3 = 1,290,783 (+21),   P4 = 339,455   (-1504),   Eng 2 ptr =   543 (+2), offset = 41 (+1)
Blk   115 pointers, Eng 1: P1 = 807,611   (+588),  P2 = 1,756,755 (+154),  P3 = 1,290,788 (+5),    P4 = 339,138   (-317),    Eng 2 ptr =   547 (+4), offset = 45 (+4)
Blk   116 pointers, Eng 1: P1 = 807,697   (+86),   P2 = 1,757,635 (+880),  P3 = 1,290,831 (+43),   P4 = 340,619   (+1481),   Eng 2 ptr =   551 (+4), offset = 49 (+4)
Blk   117 pointers, Eng 1: P1 = 807,791   (+94),   P2 = 1,757,649 (+14),   P3 = 1,291,151 (+320),  P4 = 340,783   (+164),    Eng 2 ptr =   557 (+6), offset = 51 (+2)
Blk   118 pointers, Eng 1: P1 = 807,793   (+2),    P2 = 1,757,657 (+8),    P3 = 1,291,255 (+104),  P4 = 340,807   (+24),     Eng 2 ptr =   559 (+2), offset = 55 (+4)
Blk   119 pointers, Eng 1: P1 = 808,709   (+916),  P2 = 1,757,898 (+241),  P3 = 1,291,651 (+396),  P4 = 340,551   (-256),    Eng 2 ptr =   562 (+3), offset = 57 (+2)
Blk   120 pointers, Eng 1: P1 = 809,617   (+908),  P2 = 1,758,134 (+236),  P3 = 1,291,827 (+176),  P4 = 343,322   (+2771),   Eng 2 ptr =   564 (+2), offset = 58 (+1)
```

```
Blk     1 pointers, Eng 1: P1 = 1,725,418          P2 = 774,161           P3 = 1,260,439          P4 = 178,267            Eng 2 ptr =    49,      offset =  6
Blk     2 pointers, Eng 1: P1 = 1,725,474 (+56),   P2 = 774,275   (+114), P3 = 1,260,809 (+370), P4 = 178,580   (+313), Eng 2 ptr =    54 (+5), offset = 10 (+4)
Blk     3 pointers, Eng 1: P1 = 1,725,714 (+240),  P2 = 775,215   (+940), P3 = 1,261,159 (+350), P4 = 180,072   (+1492), Eng 2 ptr =    60 (+6), offset = 14 (+4)
Blk     4 pointers, Eng 1: P1 = 1,725,849 (+135),  P2 = 775,237   (+22),  P3 = 1,261,168 (+9),   P4 = 180,098   (+26),  Eng 2 ptr =    62 (+2), offset = 16 (+2)
Blk     5 pointers, Eng 1: P1 = 1,725,899 (+50),   P2 = 775,269   (+32),  P3 = 1,261,178 (+10),  P4 = 180,098   (0),   Eng 2 ptr =    68 (+6), offset = 18 (+2)
Blk     6 pointers, Eng 1: P1 = 1,725,915 (+16),   P2 = 775,305   (+36),  P3 = 1,261,197 (+19),  P4 = 180,096   (-2),  Eng 2 ptr =    74 (+6), offset = 21 (+3)
Blk     7 pointers, Eng 1: P1 = 1,726,429 (+514),  P2 = 775,562   (+257), P3 = 1,261,273 (+76),  P4 = 179,419   (-677), Eng 2 ptr =    77 (+3), offset = 24 (+3)
Blk     8 pointers, Eng 1: P1 = 1,727,301 (+872),  P2 = 775,598   (+36),  P3 = 1,261,925 (+652), P4 = 184,262   (+4843), Eng 2 ptr =    84 (+7), offset = 28 (+4)
Blk     9 pointers, Eng 1: P1 = 1,727,313 (+12),   P2 = 775,608   (+10),  P3 = 1,261,945 (+20),  P4 = 184,277   (+15),  Eng 2 ptr =    88 (+4), offset = 29 (+1)
Blk    10 pointers, Eng 1: P1 = 1,727,373 (+60),   P2 = 775,998   (+390), P3 = 1,261,977 (+32),  P4 = 183,637   (-640), Eng 2 ptr =    94 (+6), offset = 33 (+4)
Blk    11 pointers, Eng 1: P1 = 1,727,703 (+330),  P2 = 776,000   (+2),   P3 = 1,261,985 (+8),   P4 = 182,900   (-737), Eng 2 ptr =   102 (+8), offset = 36 (+3)
Blk    12 pointers, Eng 1: P1 = 1,729,003 (+1300), P2 = 776,123   (+123), P3 = 1,262,315 (+330), P4 = 193,787   (+10887), Eng 2 ptr =   104 (+2), offset = 40 (+4)
Blk    13 pointers, Eng 1: P1 = 1,729,885 (+882),  P2 = 776,404   (+281), P3 = 1,262,533 (+218), P4 = 196,106   (+2319), Eng 2 ptr =   111 (+7), offset = 41 (+1)
Blk    14 pointers, Eng 1: P1 = 1,729,947 (+62),   P2 = 776,407   (+3),   P3 = 1,263,319 (+786), P4 = 195,523   (-583), Eng 2 ptr =   117 (+6), offset = 45 (+4)
Blk    15 pointers, Eng 1: P1 = 1,730,124 (+177),  P2 = 776,417   (+10),  P3 = 1,263,389 (+70),  P4 = 195,044   (-479), Eng 2 ptr =   122 (+5), offset = 46 (+1)
Blk    16 pointers, Eng 1: P1 = 1,730,130 (+6),    P2 = 776,429   (+12),  P3 = 1,263,402 (+13),  P4 = 194,854   (-190), Eng 2 ptr =   129 (+7), offset = 50 (+4)
Blk    17 pointers, Eng 1: P1 = 1,730,577 (+447),  P2 = 776,438   (+9),   P3 = 1,263,754 (+352), P4 = 194,768   (-86),  Eng 2 ptr =   136 (+7), offset = 53 (+3)
Blk    18 pointers, Eng 1: P1 = 1,731,386 (+809),  P2 = 777,106   (+668), P3 = 1,263,892 (+138), P4 = 194,826   (+58),  Eng 2 ptr =   143 (+7), offset = 57 (+4)
Blk    19 pointers, Eng 1: P1 = 1,731,964 (+578),  P2 = 777,115   (+9),   P3 = 1,264,096 (+204), P4 = 196,520   (+1694), Eng 2 ptr =   147 (+4), offset = 60 (+3)
Blk    20 pointers, Eng 1: P1 = 1,732,010 (+46),   P2 = 777,141   (+26),  P3 = 1,264,143 (+47),  P4 = 195,766   (-754), Eng 2 ptr =   153 (+6), offset = 63 (+3)
Blk    21 pointers, Eng 1: P1 = 1,732,179 (+169),  P2 = 777,519   (+378), P3 = 1,264,522 (+379), P4 = 194,601   (-1165), Eng 2 ptr =   159 (+6), offset = 64 (+1)
Blk    22 pointers, Eng 1: P1 = 1,732,189 (+10),   P2 = 777,537   (+18),  P3 = 1,264,769 (+247), P4 = 196,474   (+1873), Eng 2 ptr =   165 (+6), offset = 66 (+2)
Blk    23 pointers, Eng 1: P1 = 1,732,193 (+4),    P2 = 777,545   (+8),   P3 = 1,265,148 (+379), P4 = 196,155   (-319), Eng 2 ptr =   171 (+6), offset = 68 (+2)
Blk    24 pointers, Eng 1: P1 = 1,732,465 (+272),  P2 = 777,551   (+6),   P3 = 1,265,604 (+456), P4 = 195,854   (-301), Eng 2 ptr =   178 (+7), offset = 70 (+2)
Blk    25 pointers, Eng 1: P1 = 1,733,691 (+1226), P2 = 777,567   (+16),  P3 = 1,265,893 (+289), P4 = 194,931   (-923), Eng 2 ptr =   184 (+6), offset = 74 (+4)
Blk    26 pointers, Eng 1: P1 = 1,733,726 (+35),   P2 = 777,689   (+122), P3 = 1,266,221 (+328), P4 = 195,418   (+487), Eng 2 ptr =   190 (+6), offset = 78 (+4)
Blk    27 pointers, Eng 1: P1 = 1,733,765 (+39),   P2 = 778,153   (+464), P3 = 1,266,385 (+164), P4 = 194,923   (-495), Eng 2 ptr =   198 (+8), offset = 80 (+2)
Blk    28 pointers, Eng 1: P1 = 1,733,773 (+8),    P2 = 778,225   (+72),  P3 = 1,266,609 (+224), P4 = 194,644   (-279), Eng 2 ptr =   203 (+5), offset = 82 (+2)
Blk    29 pointers, Eng 1: P1 = 1,734,299 (+526),  P2 = 778,259   (+34),  P3 = 1,266,642 (+33),  P4 = 181,699   (-12945), Eng 2 ptr =   207 (+4), offset = 86 (+4)
Blk    30 pointers, Eng 1: P1 = 1,734,411 (+112),  P2 = 779,095   (+836), P3 = 1,266,655 (+13),  P4 = 182,019   (+320), Eng 2 ptr =   215 (+8), offset = 87 (+1)
Blk    31 pointers, Eng 1: P1 = 1,734,573 (+162),  P2 = 779,399   (+304), P3 = 1,266,657 (+2),   P4 = 180,300   (-1719), Eng 2 ptr =   220 (+5), offset = 91 (+4)
Blk    32 pointers, Eng 1: P1 = 1,735,223 (+650),  P2 = 779,464   (+65),  P3 = 1,266,681 (+24),  P4 = 183,684   (+3384), Eng 2 ptr =   221 (+1), offset = 95 (+4)
Blk    33 pointers, Eng 1: P1 = 1,736,167 (+944),  P2 = 779,595   (+131), P3 = 1,266,823 (+142), P4 = 183,465   (-219), Eng 2 ptr =   227 (+6), offset = 97 (+2)
Blk    34 pointers, Eng 1: P1 = 1,736,335 (+168),  P2 = 779,606   (+11),  P3 = 1,267,213 (+390), P4 = 183,644   (+179), Eng 2 ptr =   235 (+8), offset = 99 (+2)
Blk    35 pointers, Eng 1: P1 = 1,736,977 (+642),  P2 = 779,727   (+121), P3 = 1,268,149 (+936), P4 = 146,941   (-36703), Eng 2 ptr =   240 (+5), offset =102 (+3)
Blk    36 pointers, Eng 1: P1 = 1,737,277 (+300),  P2 = 780,160   (+433), P3 = 1,268,265 (+116), P4 = 147,202   (+261), Eng 2 ptr =   241 (+1), offset =103 (+1)
Blk    37 pointers, Eng 1: P1 = 1,737,863 (+586),  P2 = 780,168   (+8),   P3 = 1,268,852 (+587), P4 = 147,424   (+222), Eng 2 ptr =   242 (+1), offset =105 (+2)
Blk    38 pointers, Eng 1: P1 = 1,737,873 (+10),   P2 = 780,826   (+658), P3 = 1,269,684 (+832), P4 = 143,776   (-3648), Eng 2 ptr =   245 (+3), offset =106 (+1)
Blk    39 pointers, Eng 1: P1 = 1,737,884 (+11),   P2 = 780,834   (+8),   P3 = 1,270,098 (+414), P4 = 135,033   (-8743), Eng 2 ptr =   248 (+3), offset =109 (+3)
Blk    40 pointers, Eng 1: P1 = 1,737,952 (+68),   P2 = 780,846   (+12),  P3 = 1,270,463 (+365), P4 = 134,398   (-635), Eng 2 ptr =   255 (+7), offset =112 (+3)
Blk    41 pointers, Eng 1: P1 = 1,738,230 (+278),  P2 = 781,522   (+676), P3 = 1,270,473 (+10),  P4 = 133,786   (-612), Eng 2 ptr =   262 (+7), offset =113 (+1)
Blk    42 pointers, Eng 1: P1 = 1,738,485 (+255),  P2 = 781,932   (+410), P3 = 1,270,634 (+161), P4 = 133,768   (-18),  Eng 2 ptr =   270 (+8), offset =117 (+4)
Blk    43 pointers, Eng 1: P1 = 1,738,864 (+379),  P2 = 782,210   (+278), P3 = 1,270,654 (+20),  P4 = 132,597   (-1171), Eng 2 ptr =   271 (+1), offset =118 (+1)
Blk    44 pointers, Eng 1: P1 = 1,739,097 (+233),  P2 = 783,014   (+804), P3 = 1,270,696 (+42),  P4 = 137,528   (+4931), Eng 2 ptr =   278 (+7), offset =121 (+3)
Blk    45 pointers, Eng 1: P1 = 1,739,217 (+120),  P2 = 783,268   (+254), P3 = 1,270,716 (+20),  P4 = 137,387   (-141), Eng 2 ptr =   280 (+2), offset =122 (+1)
Blk    46 pointers, Eng 1: P1 = 1,739,682 (+465),  P2 = 783,435   (+167), P3 = 1,270,728 (+12),  P4 = 138,567   (+1180), Eng 2 ptr =   285 (+5), offset =123 (+1)
Blk    47 pointers, Eng 1: P1 = 1,740,544 (+862),  P2 = 783,443   (+8),   P3 = 1,270,790 (+62),  P4 = 138,806   (+239), Eng 2 ptr =   286 (+1), offset =125 (+2)
Blk    48 pointers, Eng 1: P1 = 1,740,576 (+32),   P2 = 783,463   (+20),  P3 = 1,271,442 (+652), P4 = 138,380   (-426), Eng 2 ptr =   294 (+8), offset =127 (+2)
Blk    49 pointers, Eng 1: P1 = 1,741,242 (+666),  P2 = 783,698   (+235), P3 = 1,271,447 (+5),   P4 = 132,055   (-6325), Eng 2 ptr =   298 (+4), offset =130 (+3)
Blk    50 pointers, Eng 1: P1 = 1,741,258 (+16),   P2 = 783,711   (+13),  P3 = 1,271,888 (+441), P4 = 134,509   (+2454), Eng 2 ptr =   300 (+2), offset =132 (+2)
Blk    51 pointers, Eng 1: P1 = 1,741,520 (+262),  P2 = 784,101   (+390), P3 = 1,272,210 (+322), P4 = 134,162   (-347), Eng 2 ptr =   306 (+6), offset =135 (+3)
Blk    52 pointers, Eng 1: P1 = 1,741,525 (+5),    P2 = 784,164   (+63),  P3 = 1,272,336 (+126), P4 = 134,737   (+575), Eng 2 ptr =   312 (+6), offset =136 (+1)
Blk    53 pointers, Eng 1: P1 = 1,741,531 (+6),    P2 = 784,412   (+248), P3 = 1,272,636 (+300), P4 = 131,150   (-3587), Eng 2 ptr =   313 (+1), offset =140 (+4)
Blk    54 pointers, Eng 1: P1 = 1,741,534 (+3),    P2 = 784,880   (+468), P3 = 1,273,148 (+512), P4 = 133,029   (+1879), Eng 2 ptr =   314 (+1), offset =141 (+1)
Blk    55 pointers, Eng 1: P1 = 1,741,570 (+36),   P2 = 785,748   (+868), P3 = 1,273,165 (+17),  P4 = 131,753   (-1276), Eng 2 ptr =   317 (+3), offset =143 (+2)
Blk    56 pointers, Eng 1: P1 = 1,742,396 (+826),  P2 = 786,161   (+413), P3 = 1,273,256 (+91),  P4 = 132,298   (+545), Eng 2 ptr =   319 (+2), offset =144 (+1)
Blk    57 pointers, Eng 1: P1 = 1,742,836 (+440),  P2 = 787,043   (+882), P3 = 1,273,785 (+529), P4 = 392,187   (+259889), Eng 2 ptr =   324 (+5), offset =145 (+1)
Blk    58 pointers, Eng 1: P1 = 1,742,845 (+9),    P2 = 787,153   (+110), P3 = 1,274,094 (+309), P4 = 386,075   (-6112), Eng 2 ptr =   330 (+6), offset =147 (+2)
Blk    59 pointers, Eng 1: P1 = 1,742,912 (+67),   P2 = 787,419   (+266), P3 = 1,274,317 (+223), P4 = 387,978   (+1903), Eng 2 ptr =   331 (+1), offset =151 (+4)
Blk    60 pointers, Eng 1: P1 = 1,742,924 (+12),   P2 = 787,593   (+174), P3 = 1,274,711 (+394), P4 = 388,676   (+698), Eng 2 ptr =   333 (+2), offset =155 (+4)
```

```
Blk     61 pointers, Eng 1: P1 = 1,742,934 (+10),   P2 = 787,717   (+124),  P3 = 1,274,739 (+28),   P4 = 389,103   (+427),    Eng 2 ptr =   339 (+6), Offset =156 (+1)
Blk     62 pointers, Eng 1: P1 = 1,743,293 (+359),  P2 = 787,813   (+96),   P3 = 1,274,755 (+16),   P4 = 388,752   (-351),    Eng 2 ptr =   341 (+2), Offset =158 (+2)
Blk     63 pointers, Eng 1: P1 = 1,743,533 (+240),  P2 = 787,829   (+16),   P3 = 1,274,985 (+230),  P4 = 387,701   (-1051),   Eng 2 ptr =   347 (+6), Offset =159 (+1)
Blk     64 pointers, Eng 1: P1 = 1,744,002 (+469),  P2 = 788,031   (+202),  P3 = 1,275,923 (+938),  P4 = 385,902   (-1799),   Eng 2 ptr =   352 (+5), Offset =160 (+1)
Blk     65 pointers, Eng 1: P1 = 1,744,067 (+65),   P2 = 788,202   (+171),  P3 = 1,276,053 (+130),  P4 = 385,914   (+12),     Eng 2 ptr =   357 (+5), Offset =163 (+3)
Blk     66 pointers, Eng 1: P1 = 1,744,091 (+24),   P2 = 788,294   (+92),   P3 = 1,276,243 (+190),  P4 = 385,794   (-120),    Eng 2 ptr =   362 (+5), Offset =166 (+3)
Blk     67 pointers, Eng 1: P1 = 1,744,419 (+328),  P2 = 788,689   (+395),  P3 = 1,276,253 (+10),   P4 = 388,708   (+2914),   Eng 2 ptr =   364 (+2), Offset =167 (+1)
Blk     68 pointers, Eng 1: P1 = 1,745,015 (+596),  P2 = 789,485   (+796),  P3 = 1,276,354 (+101),  P4 = 381,831   (-6877),   Eng 2 ptr =   372 (+8), Offset =171 (+4)
Blk     69 pointers, Eng 1: P1 = 1,745,027 (+12),   P2 = 790,029   (+544),  P3 = 1,276,360 (+6),    P4 = 382,611   (+780),    Eng 2 ptr =   376 (+4), Offset =173 (+2)
Blk     70 pointers, Eng 1: P1 = 1,745,368 (+341),  P2 = 790,667   (+638),  P3 = 1,276,369 (+9),    P4 = 379,209   (-3402),   Eng 2 ptr =   378 (+2), Offset =177 (+4)
Blk     71 pointers, Eng 1: P1 = 1,745,405 (+37),   P2 = 790,893   (+226),  P3 = 1,277,203 (+834),  P4 = 379,979   (+770),    Eng 2 ptr =   381 (+3), Offset =181 (+4)
Blk     72 pointers, Eng 1: P1 = 1,745,513 (+108),  P2 = 790,904   (+11),   P3 = 1,277,533 (+330),  P4 = 380,036   (+57),     Eng 2 ptr =   383 (+2), Offset =183 (+2)
Blk     73 pointers, Eng 1: P1 = 1,746,419 (+906),  P2 = 790,928   (+24),   P3 = 1,278,465 (+932),  P4 = 341,854   (-38182),  Eng 2 ptr =   389 (+6), Offset =185 (+2)
Blk     74 pointers, Eng 1: P1 = 1,746,503 (+84),   P2 = 791,084   (+156),  P3 = 1,279,477 (+1012), P4 = 340,140   (-1714),   Eng 2 ptr =   392 (+3), Offset =186 (+1)
Blk     75 pointers, Eng 1: P1 = 1,746,627 (+124),  P2 = 791,453   (+369),  P3 = 1,280,347 (+870),  P4 = 343,349   (+3209),   Eng 2 ptr =   397 (+5), Offset =189 (+3)
Blk     76 pointers, Eng 1: P1 = 1,747,443 (+816),  P2 = 791,746   (+293),  P3 = 1,280,701 (+354),  P4 = 341,692   (-1657),   Eng 2 ptr =   399 (+2), Offset =191 (+2)
Blk     77 pointers, Eng 1: P1 = 1,748,177 (+734),  P2 = 792,235   (+489),  P3 = 1,280,971 (+270),  P4 = 340,193   (-1499),   Eng 2 ptr =   401 (+2), Offset =193 (+2)
Blk     78 pointers, Eng 1: P1 = 1,748,188 (+11),   P2 = 792,806   (+571),  P3 = 1,281,541 (+570),  P4 = 342,895   (+2702),   Eng 2 ptr =   404 (+3), Offset =196 (+3)
Blk     79 pointers, Eng 1: P1 = 1,748,794 (+606),  P2 = 792,877   (+71),   P3 = 1,281,634 (+93),   P4 = 342,314   (-581),    Eng 2 ptr =   410 (+6), Offset =197 (+1)
Blk     80 pointers, Eng 1: P1 = 1,748,980 (+186),  P2 = 793,002   (+125),  P3 = 1,281,649 (+15),   P4 = 342,388   (+74),     Eng 2 ptr =   413 (+3), Offset =198 (+1)
Blk     81 pointers, Eng 1: P1 = 1,749,988 (+1008), P2 = 793,159   (+157),  P3 = 1,281,654 (+5),    P4 = 337,590   (-4798),   Eng 2 ptr =   420 (+7), Offset =199 (+1)
Blk     82 pointers, Eng 1: P1 = 1,750,867 (+879),  P2 = 793,204   (+45),   P3 = 1,282,490 (+836),  P4 = 343,547   (+5957),   Eng 2 ptr =   428 (+8), Offset =202 (+3)
Blk     83 pointers, Eng 1: P1 = 1,751,530 (+663),  P2 = 793,362   (+158),  P3 = 1,282,641 (+151),  P4 = 340,427   (-3120),   Eng 2 ptr =   430 (+2), Offset =204 (+2)
Blk     84 pointers, Eng 1: P1 = 1,751,750 (+220),  P2 = 793,377   (+15),   P3 = 1,282,696 (+55),   P4 = 340,497   (+70),     Eng 2 ptr =   433 (+3), Offset =207 (+3)
Blk     85 pointers, Eng 1: P1 = 1,752,428 (+678),  P2 = 793,799   (+422),  P3 = 1,283,634 (+938),  P4 = 341,759   (+1262),   Eng 2 ptr =   438 (+5), Offset =208 (+1)
Blk     86 pointers, Eng 1: P1 = 1,753,270 (+842),  P2 = 793,948   (+149),  P3 = 1,283,996 (+362),  P4 = 346,905   (+5146),   Eng 2 ptr =   445 (+7), Offset =210 (+2)
Blk     87 pointers, Eng 1: P1 = 1,753,970 (+700),  P2 = 794,244   (+296),  P3 = 1,284,184 (+188),  P4 = 345,285   (-1620),   Eng 2 ptr =   446 (+1), Offset =213 (+3)
Blk     88 pointers, Eng 1: P1 = 1,754,452 (+482),  P2 = 794,250   (+6),    P3 = 1,284,649 (+465),  P4 = 344,292   (-993),    Eng 2 ptr =   452 (+6), Offset =215 (+2)
Blk     89 pointers, Eng 1: P1 = 1,754,481 (+29),   P2 = 794,324   (+74),   P3 = 1,284,654 (+5),    P4 = 344,213   (-79),     Eng 2 ptr =   454 (+2), Offset =216 (+1)
Blk     90 pointers, Eng 1: P1 = 1,754,490 (+9),    P2 = 794,520   (+196),  P3 = 1,285,490 (+836),  P4 = 345,733   (+1520),   Eng 2 ptr =   462 (+8), Offset =219 (+3)
Blk     91 pointers, Eng 1: P1 = 1,754,973 (+483),  P2 = 794,942   (+422),  P3 = 1,285,628 (+138),  P4 = 359,057   (+13324),  Eng 2 ptr =   464 (+2), Offset =222 (+3)
Blk     92 pointers, Eng 1: P1 = 1,755,023 (+50),   P2 = 794,966   (+24),   P3 = 1,286,074 (+446),  P4 = 358,485   (-572),    Eng 2 ptr =   470 (+6), Offset =224 (+2)
Blk     93 pointers, Eng 1: P1 = 1,755,275 (+252),  P2 = 794,990   (+24),   P3 = 1,286,578 (+504),  P4 = 346,465   (-12020),  Eng 2 ptr =   471 (+1), Offset =225 (+1)
Blk     94 pointers, Eng 1: P1 = 1,755,917 (+642),  P2 = 795,444   (+454),  P3 = 1,287,406 (+828),  P4 = 347,624   (+1159),   Eng 2 ptr =   477 (+6), Offset =226 (+1)
Blk     95 pointers, Eng 1: P1 = 1,756,102 (+185),  P2 = 796,018   (+574),  P3 = 1,287,776 (+370),  P4 = 346,580   (-1044),   Eng 2 ptr =   484 (+7), Offset =228 (+2)
Blk     96 pointers, Eng 1: P1 = 1,756,107 (+5),    P2 = 796,486   (+468),  P3 = 1,288,568 (+792),  P4 = 345,338   (-1242),   Eng 2 ptr =   488 (+4), Offset =231 (+3)
Blk     97 pointers, Eng 1: P1 = 1,756,527 (+420),  P2 = 796,826   (+340),  P3 = 1,288,572 (+4),    P4 = 347,534   (+2196),   Eng 2 ptr =   495 (+7), Offset =232 (+1)
Blk     98 pointers, Eng 1: P1 = 1,756,767 (+240),  P2 = 796,843   (+17),   P3 = 1,288,588 (+16),   P4 = 347,744   (+210),    Eng 2 ptr =   499 (+4), Offset =235 (+3)
Blk     99 pointers, Eng 1: P1 = 1,756,769 (+2),    P2 = 796,874   (+31),   P3 = 1,288,591 (+3),    P4 = 347,698   (-46),     Eng 2 ptr =   506 (+7), Offset =236 (+1)
Blk    100 pointers, Eng 1: P1 = 1,756,789 (+20),   P2 = 797,338   (+464),  P3 = 1,289,519 (+928),  P4 = 346,316   (-1382),   Eng 2 ptr =   508 (+2), Offset =240 (+4)
Blk    101 pointers, Eng 1: P1 = 1,756,971 (+182),  P2 = 797,429   (+91),   P3 = 1,289,730 (+211),  P4 = 346,832   (+516),    Eng 2 ptr =   511 (+3), Offset =243 (+3)
Blk    102 pointers, Eng 1: P1 = 1,757,827 (+856),  P2 = 797,477   (+48),   P3 = 1,289,786 (+56),   P4 = 349,544   (+2712),   Eng 2 ptr =   514 (+3), Offset =246 (+3)
Blk    103 pointers, Eng 1: P1 = 1,758,296 (+469),  P2 = 797,641   (+164),  P3 = 1,289,849 (+63),   P4 = 348,951   (-593),    Eng 2 ptr =   519 (+5), Offset =248 (+2)
Blk    104 pointers, Eng 1: P1 = 1,758,848 (+552),  P2 = 798,175   (+534),  P3 = 1,289,865 (+16),   P4 = 349,988   (+1037),   Eng 2 ptr =   526 (+7), Offset =252 (+4)
Blk    105 pointers, Eng 1: P1 = 1,758,946 (+98),   P2 = 798,343   (+168),  P3 = 1,290,034 (+169),  P4 = 349,630   (-358),    Eng 2 ptr =   532 (+6), Offset =253 (+1)
Blk    106 pointers, Eng 1: P1 = 1,759,234 (+288),  P2 = 798,357   (+14),   P3 = 1,290,222 (+188),  P4 = 350,819   (+1189),   Eng 2 ptr =   538 (+6), Offset =256 (+3)
Blk    107 pointers, Eng 1: P1 = 1,759,906 (+672),  P2 = 798,370   (+13),   P3 = 1,290,688 (+466),  P4 = 346,073   (-4746),   Eng 2 ptr =   541 (+3), Offset =  4 (+4)
Blk    108 pointers, Eng 1: P1 = 1,759,930 (+24),   P2 = 798,866   (+496),  P3 = 1,291,045 (+357),  P4 = 350,999   (+4926),   Eng 2 ptr =   543 (+2), Offset =  5 (+1)
Blk    109 pointers, Eng 1: P1 = 1,759,943 (+13),   P2 = 798,984   (+118),  P3 = 1,291,050 (+5),    P4 = 350,919   (-80),     Eng 2 ptr =   548 (+5), Offset =  6 (+1)
Blk    110 pointers, Eng 1: P1 = 1,760,014 (+71),   P2 = 799,876   (+892),  P3 = 1,291,802 (+752),  P4 = 351,166   (+247),    Eng 2 ptr =   553 (+5), Offset =  7 (+1)
Blk    111 pointers, Eng 1: P1 = 1,760,041 (+27),   P2 = 799,924   (+48),   P3 = 1,292,622 (+820),  P4 = 349,428   (-1738),   Eng 2 ptr =   557 (+4), Offset = 10 (+3)
Blk    112 pointers, Eng 1: P1 = 1,760,694 (+653),  P2 = 799,942   (+18),   P3 = 1,293,510 (+888),  P4 = 350,088   (+660),    Eng 2 ptr =   563 (+6), Offset = 13 (+3)
Blk    113 pointers, Eng 1: P1 = 1,760,712 (+18),   P2 = 800,012   (+70),   P3 = 1,293,951 (+441),  P4 = 349,325   (-763),    Eng 2 ptr =   569 (+6), Offset = 15 (+2)
Blk    114 pointers, Eng 1: P1 = 1,761,018 (+306),  P2 = 800,480   (+468),  P3 = 1,293,953 (+2),    P4 = 349,361   (+36),     Eng 2 ptr =   570 (+1), Offset = 16 (+1)
Blk    115 pointers, Eng 1: P1 = 1,761,244 (+226),  P2 = 800,550   (+70),   P3 = 1,294,335 (+382),  P4 = 349,520   (+159),    Eng 2 ptr =   577 (+7), Offset = 20 (+4)
Blk    116 pointers, Eng 1: P1 = 1,761,746 (+502),  P2 = 800,786   (+236),  P3 = 1,294,341 (+6),    P4 = 334,739   (-14781),  Eng 2 ptr =   579 (+2), Offset = 21 (+1)
Blk    117 pointers, Eng 1: P1 = 1,762,382 (+636),  P2 = 800,800   (+14),   P3 = 1,294,543 (+202),  P4 = 335,614   (+875),    Eng 2 ptr =   582 (+3), Offset = 25 (+4)
Blk    118 pointers, Eng 1: P1 = 1,763,822 (+1440), P2 = 800,803   (+3),    P3 = 1,294,714 (+171),  P4 = 332,512   (-3102),   Eng 2 ptr =   589 (+7), Offset = 26 (+1)
Blk    119 pointers, Eng 1: P1 = 1,764,446 (+624),  P2 = 800,830   (+27),   P3 = 1,294,736 (+22),   P4 = 333,758   (+1246),   Eng 2 ptr =   595 (+6), Offset = 29 (+3)
Blk    120 pointers, Eng 1: P1 = 1,764,767 (+321),  P2 = 801,314   (+484),  P3 = 1,294,744 (+8),    P4 = 332,910   (-848),    Eng 2 ptr =   597 (+2), Offset = 30 (+1)
```

Appendix F

```
Sub MakeKeyTable()
    Dim i As Integer, keyTable(511) As Integer, rndNums(10000) As Byte, used(255) As Byte, p As Long, _
        p1 As Long, p2 As Long, ptr As Long, progTime As Single

    ' Loop to prepare the used array
    For i = 0 To 255: used(i) = 0: Next i
    ' Loop to initialize the random number array - set the pointer
    ptr = 0
    ' Do until we have at least 4 of every number from 0 to 255
    Do
        ' Select a random number and count the occurence of this number
        rndNums(ptr) = GRN: used(rndNums(ptr)) = used(rndNums(ptr)) + 1
        ' Advance the pointer
        ptr = ptr + 1
        ' Look at the array for at least 4 of every value from 0 to 255
        For p = 0 To 255
            ' If less than 4, another # needs to be placed, prematurely exit the loop - 'p' will not be 256
            If used(p) < 4 Then Exit For
        Next p
        ' Keep going until all values have recorded at least 4 occurrences in the 'rndNums' array
    Loop While p < 256
    ' Loops to prepare the used and key table arrays
    For i = 0 To 255: used(i) = 0: Next i
    For i = 0 To 511: keyTable(i) = 300: Next i
    ' Loop to transfer the numbers to the key array
    For i = 0 To 511
        ' Randomly find one in the list that doesn't have 2 numbers in the key table yet
        Do: p1 = GRN(ptr): Loop While used(rndNums(p1)) = 2
        ' Randomly find a location that needs to be loaded
        Do: p2 = GRN(512): Loop While keyTable(p2) < 300
        ' Load it
        keyTable(p2) = rndNums(p1)
        ' Count this one
        used(keyTable(p2)) = used(keyTable(p2)) + 1
    Next i
    ' Open the key table for writing
    Open App.Path + "\vernamTwo.tbl" For Output As #1
    ' Write the key table to the file
    For i = 0 To 510: Print #1, ls(keyTable(i)); ",";: Next i: Print #1, ls(keyTable(i))
    ' Close the file so the input function operates
    Close #1
    ' Report
    report.Caption = "A new key table is constructed in" + Str$(Timer - progTime) + " seconds"
End Sub

Function ls(num) as String: ls = Ltrim$(Str$(num)): End Function
```

The key numbers used for this technology demonstration document:

192,189,67,23,199,12,219,192,13,186,96,161,53,33,86,155,250,110,11,174,233,191,254,59,194,185,60,145,46,228,178,105,15,238,164
99,181,0,67,121,81,107,194,79,170,62,24,101,109,222,244,92,243,69,104,147,212,176,116,10,6,25,255,218,44,128,180,18,26,169,111
245,4,24,171,197,193,57,211,76,112,61,22,93,95,17,49,55,144,201,33,119,51,14,65,210,6,46,27,133,98,3,183,28,198,154,254,182,227
146,236,83,160,53,40,132,249,208,34,141,179,5,126,84,162,152,58,116,119,50,246,153,70,35,30,238,14,250,10,104,42,247,19,108,103
37,143,123,21,27,146,113,158,52,148,96,223,105,210,174,78,217,240,203,221,178,228,209,206,196,218,88,230,237,118,198,153,107,215
202,43,200,54,184,166,63,212,167,150,74,56,188,82,221,113,195,173,40,130,36,231,181,206,252,75,95,48,71,103,249,186,29,235,89
217,132,151,134,157,39,64,156,224,248,72,157,122,191,236,126,137,41,149,5,120,73,239,244,207,239,161,226,4,202,91,147,60,123,31
72,55,32,224,78,74,90,137,54,94,240,45,168,131,165,115,20,160,232,68,9,66,91,177,43,209,167,112,124,225,11,131,253,184,79,118,69
223,220,115,106,22,251,38,200,101,173,12,148,242,130,231,51,138,149,88,129,38,97,216,156,2,68,58,145,179,197,185,87,121,29,19
175,241,102,117,48,253,237,127,172,100,26,208,117,214,143,133,241,190,37,155,144,90,125,188,159,13,77,203,232,159,162,70,71,142
193,213,7,98,50,3,252,139,39,251,108,129,110,242,154,47,187,139,255,165,205,172,80,83,77,28,75,109,80,229,150,32,18,42,1,136,201
127,57,86,20,243,102,171,44,183,248,97,17,180,124,175,158,216,61,125,34,151,92,106,136,230,120,89,213,205,135,8,138,135,114,85
84,187,189,140,169,30,246,163,87,211,66,163,1,226,41,229,76,234,23,49,122,52,227,182,94,134,25,207,63,7,152,45,8,62,170,114,166
190,0,222,93,64,16,31,225,21,168,196,234,214,177,140,35,235,73,176,99,9,47,233,220,128,59,2,111,199,215,164,16,204,82,142,195
245,56,85,100,15,36,65,141,204,219,81,247

Appendix G

This is a display of the final report on 3 tests. The test code first converts the first 2,000 Effective Key Streams to 256 ASCII numbers and stores them in a 2,000 x 256 structure. Then it inputs one Effective Key Stream at a time, from 2,001 to 2,000,000, converts that string to 256 ASCII numbers and compares all 2,000 sets of ASCII numbers of the first 2,000 with that one set. The code counts the ASCII numbers that match, then records the maximum encountered. So for this report, each test has executed almost 514 billion 256-character loops for each of the 3 tests.

Maximum # digits equal = 45, minimum = 0
Maximum match 0 was encountered 33,757 times
Maximum match 1 was encountered 580,583 times
Maximum match 2 was encountered 4,913,396 times
Maximum match 3 was encountered 28,058,957 times
Maximum match 4 was encountered 118,445,887 times
Maximum match 5 was encountered 397,957,701 times
Maximum match 6 was encountered 1,111,120,960 times
Maximum match 7 was encountered 2,643,247,243 times
Maximum match 8 was encountered 5,486,496,311 times
Maximum match 9 was encountered 10,077,008,213 times
Maximum match 10 was encountered 16,589,405,202 times
Maximum match 11 was encountered 24,736,752,007 times
Maximum match 12 was encountered 33,664,046,718 times
Maximum match 13 was encountered 42,126,865,474 times
Maximum match 14 was encountered 48,753,722,490 times
Maximum match 15 was encountered 52,433,354,102 times
Maximum match 16 was encountered 52,655,561,351 times
Maximum match 17 was encountered 49,557,332,660 times
Maximum match 18 was encountered 43,871,551,980 times
Maximum match 19 was encountered 36,633,832,607 times
Maximum match 20 was encountered 28,943,677,783 times
Maximum match 21 was encountered 21,683,923,875 times
Maximum match 22 was encountered 15,442,768,204 times
Maximum match 23 was encountered 10,475,308,531 times
Maximum match 24 was encountered 6,777,830,500 times
Maximum match 25 was encountered 4,193,664,069 times
Maximum match 26 was encountered 2,484,780,048 times
Maximum match 27 was encountered 1,411,207,717 times
Maximum match 28 was encountered 768,705,061 times
Maximum match 29 was encountered 402,970,817 times
Maximum match 30 was encountered 203,032,183 times
Maximum match 31 was encountered 98,787,517 times
Maximum match 32 was encountered 46,425,506 times
Maximum match 33 was encountered 20,938,723 times
Maximum match 34 was encountered 9,225,147 times
Maximum match 35 was encountered 3,874,007 times
Maximum match 36 was encountered 1,571,512 times
Maximum match 37 was encountered 625,445 times
Maximum match 38 was encountered 247,571 times
Maximum match 39 was encountered 89,379 times
Maximum match 40 was encountered 31,422 times
Maximum match 41 was encountered 13,211 times
Maximum match 42 was encountered 3,875 times
Maximum match 43 was encountered 1,354 times
Maximum match 44 was encountered 578 times
Maximum match 45 was encountered 366 times
Grand Total of checks = 513,859,992,000

Maximum # digits equal = 44, minimum = 0
Maximum match 0 was encountered 37,836 times
Maximum match 1 was encountered 589,129 times
Maximum match 2 was encountered 4,952,581 times
Maximum match 3 was encountered 28,064,870 times
Maximum match 4 was encountered 118,448,065 times
Maximum match 5 was encountered 397,604,961 times
Maximum match 6 was encountered 1,108,503,567 times
Maximum match 7 was encountered 2,640,933,352 times
Maximum match 8 was encountered 5,480,861,432 times
Maximum match 9 was encountered 10,063,564,755 times
Maximum match 10 was encountered 16,571,349,882 times
Maximum match 11 was encountered 24,706,631,509 times
Maximum match 12 was encountered 33,636,343,015 times
Maximum match 13 was encountered 42,092,126,439 times
Maximum match 14 was encountered 48,708,480,371 times
Maximum match 15 was encountered 52,385,886,648 times
Maximum match 16 was encountered 52,611,727,620 times
Maximum match 17 was encountered 49,518,162,018 times
Maximum match 18 was encountered 43,838,323,379 times
Maximum match 19 was encountered 36,612,754,998 times
Maximum match 20 was encountered 28,919,533,566 times
Maximum match 21 was encountered 21,672,742,385 times
Maximum match 22 was encountered 15,432,422,939 times
Maximum match 23 was encountered 10,470,082,203 times
Maximum match 24 was encountered 6,774,038,136 times
Maximum match 25 was encountered 4,192,664,303 times
Maximum match 26 was encountered 2,482,866,417 times
Maximum match 27 was encountered 1,411,187,546 times
Maximum match 28 was encountered 769,236,369 times
Maximum match 29 was encountered 403,064,039 times
Maximum match 30 was encountered 203,285,211 times
Maximum match 31 was encountered 98,452,046 times
Maximum match 32 was encountered 46,297,366 times
Maximum match 33 was encountered 21,022,551 times
Maximum match 34 was encountered 9,181,557 times
Maximum match 35 was encountered 3,873,302 times
Maximum match 36 was encountered 1,581,359 times
Maximum match 37 was encountered 643,249 times
Maximum match 38 was encountered 242,150 times
Maximum match 39 was encountered 92,893 times
Maximum match 40 was encountered 31,929 times
Maximum match 41 was encountered 11,859 times
Maximum match 42 was encountered 4,748 times
Maximum match 43 was encountered 1,329 times
Maximum match 44 was encountered 121 times
Grand Total of checks = 513,437,906,000

Maximum # digits equal = 47, minimum = 0
Maximum match 0 was encountered 31,655 times
Maximum match 1 was encountered 600,478 times
Maximum match 2 was encountered 4,959,624 times
Maximum match 3 was encountered 28,091,049 times
Maximum match 4 was encountered 118,263,709 times
Maximum match 5 was encountered 397,944,652 times
Maximum match 6 was encountered 1,110,171,846 times
Maximum match 7 was encountered 2,642,115,995 times
Maximum match 8 was encountered 5,480,836,173 times
Maximum match 9 was encountered 10,069,368,821 times
Maximum match 10 was encountered 16,580,548,303 times
Maximum match 11 was encountered 24,717,355,214 times
Maximum match 12 was encountered 33,647,169,575 times
Maximum match 13 was encountered 42,100,546,074 times
Maximum match 14 was encountered 48,714,384,320 times
Maximum match 15 was encountered 52,402,286,789 times
Maximum match 16 was encountered 52,618,671,750 times
Maximum match 17 was encountered 49,516,762,397 times
Maximum match 18 was encountered 43,833,578,760 times
Maximum match 19 was encountered 36,606,515,214 times
Maximum match 20 was encountered 28,921,542,928 times
Maximum match 21 was encountered 21,667,862,647 times
Maximum match 22 was encountered 15,427,524,681 times
Maximum match 23 was encountered 10,467,357,508 times
Maximum match 24 was encountered 6,773,602,557 times
Maximum match 25 was encountered 4,190,774,551 times
Maximum match 26 was encountered 2,482,166,238 times
Maximum match 27 was encountered 1,409,656,853 times
Maximum match 28 was encountered 767,872,193 times
Maximum match 29 was encountered 402,581,776 times
Maximum match 30 was encountered 203,306,782 times
Maximum match 31 was encountered 98,905,682 times
Maximum match 32 was encountered 46,312,997 times
Maximum match 33 was encountered 20,963,784 times
Maximum match 34 was encountered 9,094,170 times
Maximum match 35 was encountered 3,918,248 times
Maximum match 36 was encountered 1,569,119 times
Maximum match 37 was encountered 656,425 times
Maximum match 38 was encountered 240,982 times
Maximum match 39 was encountered 92,287 times
Maximum match 40 was encountered 35,673 times
Maximum match 41 was encountered 11,467 times
Maximum match 42 was encountered 4,235 times
Maximum match 43 was encountered 1,033 times
Maximum match 44 was encountered 508 times
Maximum match 45 was encountered 124 times
Maximum match 47 was encountered 154 times
Grand Total of checks = 513,486,258,000