



(51) International Patent Classification:

H03M 7/40 (2006.01) H04L 9/28 (2006.01)
G06F 21/62 (2013.01)

(21) International Application Number:

PCT/CA2013/050969

(22) International Filing Date:

16 December 2013 (16.12.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: MCAFEE, INC. [US/US]; 2821 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventor: BRUNO, Richard; Unit 10, 133 High Street, St. Albans, Vermont 05478 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

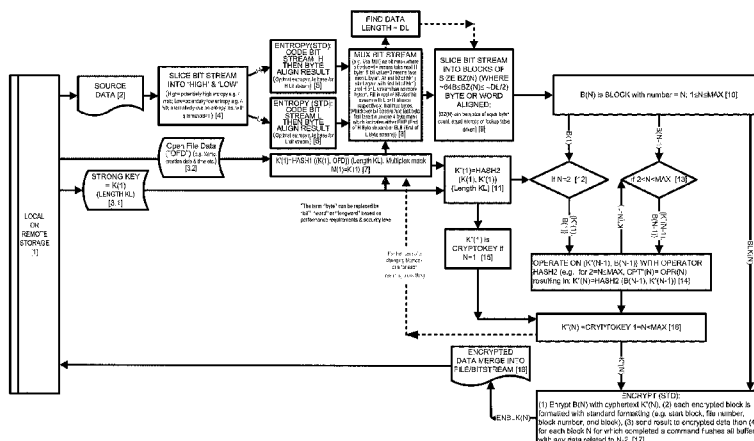
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: PROCESS EFFICIENT PREPROCESSING FOR AN ENCRYPTION STANDARD

Figure 1: Bit Stream Sliced into Entropy Bands, Multiplexed and Block-Sliced and Encrypted in a Process Based on any Entropy and Encryption Coding Standards



(57) Abstract: A pre-encryption process for symmetric encryption processes that inputs a bit stream into any existing or future encryption standard to increase encryption complexity with a disproportionate increase in processing time. The first encoding step is the two-stage generation of two strong keys based on a seed strong crypto key and known information from the Source Data bit stream. The second step is to split and encode the bit stream based on entropy levels. After entropy coding, the aligned bit streams are multiplexed in a cyclic fashion to generate one resulting bit stream. The third step is to slice the resulting bit stream into blocks, encrypting each block and adding each block to a coded output bit stream. Each new strong crypto key is derived from the previous crypto key and the previous pre-processed bit stream data. The decoding process is provided that is a simplified inverse of the encoding process.

Process Efficient Preprocessing for any Encryption Standard

Field

The present invention generally relates to methods and processes to enhance the value of any encryption standard for securing data as well as securely communicating any data. More particularly, it relates to an overall method of encryption and decryption based on any existing encryption standard such that the encoding and decoding process, as well as the resulting cipher text bit stream, is made more robust.

10 Background

Encryption has become, amongst others, an integral part of digital transaction systems, secure digital communications, anonymous data storage and digital vault systems. Many encryption standards, de jure and de facto, are in use today and more are to come. Some of these standards use asymmetric encryption requiring two keys, a public key and a private key, whilst others use symmetric encryption requiring one key. In this latter case the encryption key needs to be securely shared between a digital information source (the "Sender") and the digital information destination (the "Receiver").

While encryption standards use a variety of algorithms they have also, in the past, been enhanced by such processes as entropy coding, on the one hand, and changing encryption (crypto-) keys during the encoding process of one file, on the other hand. In the former case of preprocessing via entropy coding before encryption, many existing entropy encoding standards, de jure and de facto, have been or can be used. For the latter case, there are examples of using an initial crypto key plus data from the file's bit stream to create a new crypto key, for example via a hash, for the encoding of the next sample of the file's bit stream.

The problem with most of these approaches is that they do not increase encoding and decoding complexity sufficiently to thwart an attack against a weakened encryption algorithm whether with or without the use of entropy coding. Although it is easy to add complexity and computational intensity to an encryption process it is more difficult to add complexity whilst roughly retaining computational intensity.

Various encryption solutions exist in the prior art yet there has been no solution proposed that adds complexity to an encryption process (rendering the encryption process more secure) while maintaining the system resources required to operate the encryption process in a disproportionate manner.

- 5 US 8,127,135 to Ibrahim *et al.* discloses methods and systems where a first operating system and a second operating system share encrypted information using a first shared key. After a predetermined number of encrypted communications using the first shared key, the two operating systems cooperate to share encrypted communications with a second shared key in order to change the first shared key. Once the first shared key is changed the second shared
10 key is also changed.

US 8,284,945 to Breyel discloses methods and systems where a first encryption key is changed once a volume of data encrypted with that first key exceeds a predetermined threshold, this threshold determined by a particular formula derived to automatically change the key in response to the number of changes that have occurred in view of the amount of
15 encrypted data being transmitted.

US 8,369,525 to Jordan *et al.* discloses methods and systems for dynamically changing encryption keys wherein a new encryption key is sent in a message having a password key indicator randomly placed in the message text and encrypted with a previous encryption key, and the new encryption key is retrieved from the encrypted message based on the location of
20 the password key indicator.

US 8,487,791 to Korodi *et al.* discloses methods and systems for entropy encoding and decoding data using parallel coders to encode an overall input sequence as multiple subsequences, which are eventually combined to form an output bit stream. A context model is provided to separate the input sequence into multiple subsequences. Multiple encoded
25 subsequences are extracted from the output bit stream and decoded in parallel to generate a plurality of decoded subsequences which are in turn interleaved to create a reconstructed sequence.

US 8,509,555 to Meany discloses methods and systems for error-resilient entropy coding where a plurality of code words are generated that are representative of data to be encoded.
30 The code words can be generated with first and second portions, with each first portion

including information that is representative of predetermined characteristic of an associated second portion, and the second portion associated with a portion of the data to be encoded.

US 8,514,943 to Lee *et al.* discloses methods and systems for enhancing the performance of entropy coding by coding a first coefficient without loss, which includes obtaining a
5 distribution of a second coefficient included in an area of a base layer corresponding to a block of a predetermined layer among coefficients generating when coding data, selecting a suitable lookup table, and transforming the first coefficient into a value mapped on to the selected lookup table.

US Publication No. 2002/0025040 to Stein discloses methods and apparatus for generating an
10 encryption key, where a randomly ordered sequence of bytes is retrieved from memory and each byte in the sequence is assigned to a number of groups, where the number of the groups is defined by the number of bytes in the encryption key. Each group is then in turn reduced to a single byte to generate one of the resultant bytes of the encryption key.

Therefore, there is need for a more thwart-proof method and system for bit stream pre-
15 processing before encryption which adds complexity without significantly adding to computational complexity.

This background information is provided to reveal information believed by the applicant to be of possible relevance to the present invention. No admission is necessarily intended, nor should be construed, that any of the preceding information constitutes prior art against the
20 present invention.

Brief Summary

In at least one embodiment the present invention provides a method of creating at least two band entropy coded bit streams from a Source Data bit stream, the method having the steps of
25 separating the Source Data bit stream into at least two bit streams, each of the at least two bit streams associated with an entropy band having a predetermined entropy designation, the predetermined entropy designation selected from an entropy designation range between a predetermined high entropy designation and a predetermined low entropy designation, entropy coding each of the at least two bit streams separately and in parallel using a standard
30 entropy coder to create at least two band entropy coded bit streams, generating a bit mask

with a hash, the hash having inputs of at least a predetermined strong encryption key and selected data from the Source Data bit stream, and merging each of the at least two band entropy coded bit streams into a resultant band entropy coded bit stream based on a sequence of indexed values obtained from the bit mask, the sequence of indexed values proceeding in steps from a first index value of the bit mask to a last index value of the bit mask and repeating from the first index value, the steps of a step size such that the minimum step size is one bit and the maximum step size is the width of the indexed values in bits, the indexed values obtained from the bit mask being associated with one of the at least two band entropy coded bit streams to be merged.

In another embodiment the present invention provides a method of encoding sequential blocks, each of the sequential blocks being associated with a unique encryption key, the method comprising the steps of slicing a resultant band entropy coded bit stream into a plurality of sequential blocks of a predetermined block size, generating a unique strong encryption key for a current block of the plurality of sequential blocks from a hash of a previous unique encryption key associated with a previous block of the plurality of sequential blocks and the data of the previous block of the plurality of sequential blocks, encrypting, with a standard encryption coder, the current block of the plurality of sequential blocks by associated unique strong encryption key, merging the encrypted blocks into a resultant bit stream, and starting with a fixed block size for the first block in a plurality of sequential blocks, varying the block size for all subsequent blocks of the plurality of sequential blocks based on a unit value at a current mask position or retaining the block size as constant.

Description of the Drawings

The present invention will be better understood in connection with the following Figures, in which:

Figure 1 is a flow diagram illustrating the process by which the bit stream is allocated to an entropy band, multiplexed, block-sliced and encrypted based on any entropy and encryption standard algorithms in accordance with at least one embodiment of the present invention; and,

Figure 2 is a flow diagram illustrating the process by which the encoded bit stream is decrypted, block-adjoined, de-multiplexed, entropy decoded into entropy bands and bit stream adjoined in the decryption process based on any encryption and entropy standard algorithms in accordance with at least one embodiment of the present invention.

5

Detailed Description of the Embodiments

In at least one embodiment, there is provided a method for creating a pre-process to any symmetric encryption process based on an initial single strong crypto key. This method can be used with any existing or future encryption standard as well as any existing or future entropy compression standard.

10

The following process elements described herein are well within the instruction sets of current processing chips.

In at least one embodiment, the present invention provides a method of creating a streamed symmetric encryption process based on an initial, strong crypto key, any existing or future encryption standard as well as any existing or future entropy compression standard.

15

The first step of this embodiment of the process is the two-stage generation of two keys based on the strong crypto key as a seed and known information from the bit stream. The second step of this embodiment of the process is to split the bit stream into at least one high entropy (for example, high nibble) and one low entropy (for example, low nibble) bit stream and the combination of which define the whole bit stream. High nibble (also known by the alternate spelling "nybble") and low nibble will be understood as, respectively, the four most significant digits and the four least significant digits within a byte as will be readily appreciated by the skilled person.

20

Each of these bit streams can be separately byte aligned and are then, separately and in parallel, entropy compressed. After entropy coding, the two byte aligned streams can be multiplexed using one of the initial crypto keys as a cyclic multiplexing bit mask with, for example, a '0' denoting insertion of a low aligned byte and, for example, a '1' denoting insertion of a high aligned byte into the resulting bit stream. The third step in this process is to slice the resulting bit stream into equal sized blocks and, based on a different crypto key per block, to encrypt the current block and add the encrypted block to the coded output bit

25
30

stream. Each new crypto key is derived from the previous crypto key value and the previous pre-encrypted bit stream data via a hash. As will be readily understood by the appropriately skilled person, the decoding process is a simplified inverse of the encoding process.

In another embodiment, it is contemplated that the present invention provides a method of
5 defining at least two high and low entropy bit streams based on a thresholding process. In another embodiment the multiplexing unit may multiplex either a bit, a nibble, a word or a longword instead of a byte and in yet another embodiment the bit mask may be changed based on each new crypto key derived and incremented at increments of a bit or up to the multiplexing unit size.

10 In another embodiment the block size can be cyclically modulated by a look-up table in accordance to the Strong Key bit or nibble value with a last bit offset.

The following definitions will be used throughout the following specification:

(1) Definitions

Bit Mask (also referred to as “bit mask”): a sequential list of bits which are the entries of a
15 binary number.

Block: one or more group of bytes that function as the smallest encryption data unit.

Cipher Text: the resulting data after an encryption process which uses a cipher (also known as a crypto key) and an encryption process on Plain Text, i.e. the Source Data.

Crypto Key: a series of bits used within a cryptographic algorithm, an encryption process, to
20 transform Plain Text into Cipher Text or inversely Cipher Text into Plain Text.

Data Owner (also referred to as the “DO”): The person or persons, service or services, which define the party that possesses Digital Data for storage or sharing, as will be discussed in further detail below. It is contemplated that the Data Owner can be an asset use, originator or owner.

25 **Digital Data** (also referred to as the “DD”): Any digital information, such as but not limited to, a username plus password combination, software code, data files, digital media such as a photo, music, video, a document, text, notes, binary string, among any other digital information that will be readily understood by the skilled person.

Encrypted Block (also referred to as “ENBLK”): Is the resulting block after encryption by the algorithm $\text{ENCRYPT}\{K''(n); B(n)\}$ where $K''(n)$ is the n^{th} crypto key to be used to encrypt the n^{th} block $B(n)$.

Encryption Key: for the purposes of the present invention can be considered analogous to the Crypto Key.

Encryption symmetric key: A random-character bit string used with a symmetric encryption algorithm (in these embodiments, the 'Cipher' or crypto key) and which is generated and securely held by a Data Owner.

Hash 1 operator $\text{HASH1}\{K(1); \text{OFD}\}$: the operator HASH1 is some standard hash operator that operates on the strong key $K(1)$ (or other crypto key $K''(n)$) of length KL and on the selected Open File Data OFD to produce a unique multiplexing/de-multiplexing bit mask $M(1)$ (or variable multiplexing/demultiplexing bit mask $M(n)$) and the intermediate encryption key $K'(1)$, of length KL . This same hash process can be used to generate $M(n)$ of length KL from $K''(n-1)$ the actual $n-1^{\text{th}}$ encryption key and OFD.

Hash 2 operator $\text{HASH2}\{D1; D2\}$: the operator HASH2 is some standard hash operator that operates on data :

- (1) $D1$, which initially is the strong key K and then the previous block $B(n-1)$, and
- (2) $D2$, which initially is the first intermediate encryption key K' and thereafter the previous actual encryption key K'' each of length KL ,

to produce a unique new encryption key associated with one and only one block, i.e. the n^{th} crypto key, $K''(n)$ of length KL associated with block $B(n)$.

High Bit Stream: a bit stream which is anticipated to have a higher entropy content (the high entropy band) than another bit stream.

Index Value (also referred to as “index value”): the value of the index ‘ i ’ of any scalar, vector, list or matrix V_i which is an element of the same or larger vector, list or matrix

Indexed Value (also referred to as “indexed value”): the value of any scalar, vector, list or matrix V_i which is an element of the same or larger vector, list or matrix and which can be incremented by the index value ‘ i ’.

Key Length (also referred to as the “ KL ”): the length in bits of any crypto key.

Low Bit Stream: a bit stream which is anticipated to have a lower entropy content (the low entropy band) than another bit stream.

Open File Data: any data associated with a data file which is readable from a file and which is not encrypted during any encryption process.

- 5 **Slice Bit Stream:** to separate a bit stream into two or more bit streams whose contents are mutually exclusive, not redundant and when combined form the whole bit stream.

Strong Key: the well protected encryption key initially supplied by the system.

Source Data: the Digital Data which a Data Owner desires to retain as secret or communicate securely.

- 10 **Threshold** (also referred to as the "T"): an integer value between 0 and 15 determined ahead of time, e.g., amongst others, by the entropy calculation of a block, indicating the bits belonging to the high bit stream band in any byte, word or longword, as the case may be.

Vi or V(i): any variable V that can be indexed from 1 to some maximum integer value. The index integer here is represented by "i" but the characters j, k, l, m or n are also used below.

- 15 The variable can be any capital letter(s).

In at least one embodiment, the present invention provides a novel end-to-end encryption method for Digital Data sharing through a Third Party, as will be described hereinafter. Although the invention is described in terms of specific illustrative embodiment(s), it is to be understood that the embodiment(s) described herein are by way of example only and that the scope of the invention is not intended to be limited thereby.

20

(2) Example:

With reference to **Figure 1**, at least one embodiment of the present invention is illustrated in which Source Data [2], a Strong Key K(1) [3.1] and Open File Data [3.2] are obtained from either local or remote storage [1] and encrypted.

- 25 The Source Data [2] is sliced into two bit streams, a High ("H") band and a Low ("L") band each of which forms a separate bit stream, whereby one has a higher entropy content according to a thresholding rule T (e.g. the four most significant bits per byte or high nibble) and the other has a lower entropy content according to T (e.g. the four least significant bits per byte or low nibble) [4]. Then, using any known standard entropy coder that will be

readily apparent to the skilled person ("ENTROPY(...)") encode the H bit stream (i.e. ENTROPY(H)) [5] and in parallel encode the L bit stream (i.e. ENTROPY(L)) [6].

Next, this embodiment of the method proceeds to use the Strong Key $K(1)$ [3.1] and selected Open File Data ("OFD") [3.2] under the standard hash operator HASH1 to obtain the bit
 5 multiplexing mask $M(1)$ whereby $M(1)=K'(1)$ [7]. Note that, in another embodiment, by using $K''(n-1)$ (i.e. instead of $K(1)$) for the case of n greater than or equal to 2 along with previous mask selected data from OFD, i.e. OFD(n), one can generate a new multiplexing mask $M(n)$ for each new block. This multiplexing mask is then used to select whether a byte
 10 (or, in another embodiment, bit, word or longword depending on the encoder settings) is taken from the H bit stream (i.e. current $M(1)$ bit is '1') or from the L bit stream (i.e. current $M(1)$ bit is '0'), with the mask bit entry value stepped one for each byte allocated; starting at mask bit value 0 to mask bit value equal Maximum. Once Maximum is reached start with mask bit value 0 again and the process proceeds to increment as before to multiplex the two
 15 bit streams into one bit stream [8]. If the H or L bit stream has reached the last bit therein then a unique End of High Bit stream ("EHB") flag can be set after the last bit of the H bit in the multiplexed bit stream or an unique End of Low Bit stream ("ELB") flag can be set after the last bit of the L bit in the multiplexed bit stream respectively and the remaining bit stream is filled with the remaining L or H bytes respectively. Depending on the initial settings the multiplexed bit stream can now be sliced into blocks $B(n)$ (e.g. 256 byte-long blocks) whose
 20 size should preferably not be less than about 64 bytes (due to considerations discussed above) and not more than about half the file size ("DL") [9] with the term "about" here meaning within a factor of two and n being greater than 1 and less than or equal to the maximum number of blocks in the bit stream $N_{\max}$. After slicing into blocks an End Of Block (EOB) flag is added to each block.

25 Next each encryption key $K''(n)$, which is uniquely associated with a block $B(n)$, for all valid n is produced [11-16]. For $n=1$ one used the hash function HASH1 to produce $K'(1)$ and $M(1)=K'(1)$ in [7] and the hash function HASH2 to produce the first actual encryption key $K''(1)$ used to encrypted a $B(n)$ as follows:

$K''(1)=\text{HASH2}\{K(1);K'(1)\}$ [11] and [15] and then FIFO stored in [16].

30 Next, the hash function HASH2 is used as follows to produce $K''(n)$ for $1<n\leq N_{\max}$ with [12] and [13] and then [14]:

$K''(n)=HASH2\{B(n-1);K''(n-1)\}$ in [14] and then FIFO stored in [16].

Once the FIFO [16] has at least one entry the block encryption process [17] can begin. The encrypted block ENBLK(n) is produced by any standard encryption algorithm ENCRYPT as follows:

5 ENBLK(n)=ENCRYPT{K''(n); B(n)} [17].

After a block is encrypted, excluding the End Of Block (EOB) flag, the End Of Block (EOB) flag can be added post encryption and is part of the transferred encrypted block ENBLK(n). Once ENBLK(n) is transferred all data in buffers associated with index values less than or equal to n-2 can be flushed.

10 The resulting ENBLK blocks are structured in the normal fashion into a file with the same Open File Data as the Source Data file [18] and can then be sent to the system for either local or remote storage for further use [1].

Referring to **Figure 2** which is a flow diagram illustrating at least one embodiment of the inverse of the encoding process of **Figure 1** [20-30] (in other words, the decoding process) to
 15 derive the Source Data [31]. This is done by first, from local or remote storage [1], obtaining an encoded block from the encoded bit stream ENBLK(n) [1, 19], the initial Strong Key K [3.1] and the Open File Data [3.2], the latter of which is openly available in the non-encrypted portion of the encrypted file wherein the encrypted blocks ENBLK(n) are contained.

20 Using the Strong Key K(1) [3.1] and the Open File Data OFD [3.2] under the hash operator HASH1 the bit mask M(1) is obtained for de-multiplexing whereby $M(1)=K'(1)$ [7]. The first actual decryption key $K''(1)$ [11] for the first encrypted block ENBLK(1) is obtained via HASH2 as shown in [11], that is:

$$K''(1)=HASH2\{K(1);K'(1)\} [11]$$

25 and then is sent to [21] for the first encrypted block decryption using the standard decryption algorithm [21]. It is also sent to [14] to generate the next decryption crypto key based on HASH2 as follows:

$$B(1)=DECRYPT\{K''(1); ENBLK(1)\} [21]$$

$$K''(2)=HASH2\{B(1); K''(1)\} [21], [22] \text{ and } [20] \text{ to } [14]$$

which is then used for all n , with values $1 < n \leq \text{MAX}$, to generate subsequent crypto keys according to:

$K''(n) = \text{HASH2}\{B(n-1); K''(n-1)\}$ in [14] with results from [23], the previous block $B(n-1)$, and last crypto key $K''(n-1)$.

- 5 The current crypto key $K''(n)$ is used with the current encoded block $\text{ENBLK}(n)$, excluding the End Of Block (EOB) flag, to generate the current decoded block $B(n)$ as follows:

$B(n) = \text{DECRYPT}\{K''(n); \text{ENBLK}(n)\}$ [21] whereby after decryption the End Of Block (EOB) flag is added back to $B(n)$.

- 10 Once both the current crypto key $K''(n)$ and block $B(n)$ are obtained for $n > 2$ then, except for the de-multiplexing mask, all buffers can be flushed of all data associated with decryption increment value of $n-2$ or less.

- The current blocks are put into a FIFO [25] and, after removal of the End Of Block (EOB) flag, are merged into a byte aligned bit stream [26] which is de-multiplexed using the de-multiplexing mask $M(1) = K'(1)$ [27]. It is contemplated that in other embodiments, the
15 current blocks can be merged into a bit, word or longword, depending on the needs of the instant application of the embodiment.

De-multiplexing of the at least two multiplexed bit streams can then occur as follows:

- (1) Select whether a byte (or alternatively, a bit, word or longword) is placed into the H (High) bit stream (i.e. current value of $M(1)$ bit is '1'), associated with the High
20 entropy band, for Entropy Decoding [28] or is placed into the L (Low) bit stream (i.e. current $M(1)$ bit is '0'), associated with the Low entropy band, for Entropy Decoding [29].
- (2) The de-multiplexing mask bit entry number is stepped by one for each byte allocated starting at de-multiplexing mask bit number 0 until the de-multiplexing mask bit
25 reaches Maximum.
- (3) Once it has reached Maximum start with de-multiplexing mask bit number 0 again etc..
- (4) If the H or L bit stream has reached an end then either an End of High Bit stream ("EHB") flag is found after the last H bit in the multiplexed bit stream or an End of
30 Low Bit stream ("ELB") flag is found after the last L bit in the multiplexed bit stream

respectively and, then, the remaining bit stream is sent L or H Entropy Decoder respectively.

Note that for an embodiment where the multiplexing mask was varied upon encoding then one uses $K''(n-1)$, for the case of n greater than or equal to 2, along with OFD($n-1$) to
5 generate each new de-multiplexing mask $M(n)$ for each new block via HASH1. Here $K''(n-1)$ replaces $K(1)$ in [7]. Also, once both the current crypto key $K''(n)$ and block $B(n)$ are obtained for $n > 2$ then all buffers can be flushed of all data associated with decryption and de-multiplexing increment value of $n-2$ or less.

Once the H and L bit streams are entropy decoded and each placed in a FIFO then they are
10 merged by inverting the thresholding rule T; for example, each four H bits become the most significant bits in one and the same data byte and each four L bits become the least significant bits in one and the same data byte [30]. The merged bit stream is delivered to Source Data [31] and transferred to local or remote storage [1] for further use.

In other embodiments, another rule may be used in place of thresholding rule T; for example,
15 the bit number in each byte or word depending on a buffered entropy test.

Numerous modifications could be made to any of the embodiments described above without departing from the scope of the present invention. Further, it is obvious that the foregoing
embodiments of the invention are examples and can be varied in many ways. Such present or
future variations are not to be regarded as a departure from the spirit and scope of the
20 invention, and all such modifications as would be obvious to one skilled in the art are intended to be included within the scope of the following claims.

THE EMBODIMENTS OF THE INVENTION FOR WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method of creating at least two band entropy coded bit streams from a Source Data bit stream, the method comprising the steps of:

5 Separating the Source Data bit stream into at least two bit streams, each of the at least two bit streams associated with an entropy band having a predetermined entropy designation, the predetermined entropy designation selected from an entropy designation range between a predetermined high entropy designation and a predetermined low entropy designation; and,

10 Entropy coding each of the at least two bit streams separately and in parallel using a standard entropy coder to create at least two band entropy coded bit streams.

2. The method of claim 1, further comprising the step of:

15 Generating a bit mask with a hash, the hash having inputs of at least a predetermined strong encryption key and selected data from the Source Data bit stream.

3. The method of claim 2, further comprising the step of:

20 Merging each of the at least two band entropy coded bit streams into a resultant band entropy coded bit stream based on a sequence of at least one indexed value obtained from the bit mask,

25 the sequence of the at least one indexed value proceeding in steps from a first index value of the bit mask to a last index value of the bit mask and repeating from the first index value, the steps of a step size such that a minimum step size is one bit and a maximum step size is the width of at least one indexed value in bits,

 the at least one indexed value obtained from the bit mask being associated with at least one of the at least two band entropy coded bit streams to be merged.

4. The method of claim 3 wherein the at least one indexed value has a predetermined bit width and is determined based on the maximum number of the at least two band
- 30

entropy coded bit streams created, the at least one indexed value selected from the group consisting of: a bit, a nibble or a byte.

- 5 5. The method of claim 4, wherein a current mask position indexed value is used to select a unit of multiplexing data from the at least two band entropy coded bit streams, the unit of multiplexing data having a bit size obtained from the bit mask, the bit size selected from the group consisting of: a bit, a nibble or a byte, a word or a longword.

6. The method of claim 5, further comprising the steps of:

- 10 Adding an end of band flag to the resultant band entropy coded bit stream when one of the at least two band entropy coded bit streams has no further bits;
 Merging the remaining of the at least two band entropy coded bit streams until no further bits are left in the remaining of the at least two band entropy coded bit streams.

- 15 7. A method of encoding sequential blocks, each of the sequential blocks being associated with a unique encryption key, the method comprising the steps of:

 Slicing a resultant band entropy coded bit stream into a plurality of sequential blocks of a predetermined block size;

- Generating a unique strong encryption key for a current block of the plurality of sequential blocks from a hash of a previous unique encryption key associated with a previous block of the plurality of sequential blocks and the data of a previous block of the plurality of sequential blocks;

- Encrypting, with a standard encryption coder, the current block of the plurality of sequential blocks by the unique strong encryption key associated with the current block of the plurality of sequential blocks; and

 Merging the encrypted sequential blocks into a resultant bit stream.

8. The method of claim 7, further comprising the step of:

- 30 Adding an end of block flag to the end of each of the plurality of sequential blocks.

9. The method of claim 8, wherein the predetermined block size is initially a fixed block size for the first block in a plurality of sequential blocks and further comprising the step of:

5 varying the predetermined block size for all subsequent blocks of the plurality of sequential blocks based on a current mask position.

10. The method of claim 8 or claim 9, wherein the predetermined block size is initially a fixed block size for the first block in a plurality of sequential blocks and further comprising the step of:

10 maintaining the predetermined block size as constant based on any current mask position.

11. A method of decoding a resultant bit stream, the method comprising the steps of:

15 Slicing a resultant bit stream into a plurality of sequential blocks of predetermined block sizes;

 Decrypting, with a standard decryption decoder, a first block of the plurality of sequential blocks using a first generated unique strong decryption key from a hash of a known strong decryption key and the bit stream data, this first unique strong decryption key associated with the first block of the plurality of sequential blocks to generate a first segment of a band entropy coded bit stream;

 Generating a second unique strong encryption key for a subsequent block of the plurality of sequential blocks from a hash of the first generated unique encryption key associated with the first block of the plurality of sequential blocks and the data of the first block of the plurality of sequential blocks;

25 Decrypting, with a standard decryption decoder, the subsequent block of the plurality of sequential blocks using the second unique strong decryption key associated with the subsequent block of the plurality of sequential blocks to generate a second segment of a band entropy coded bit stream; and

30 Merging at least the first segment of a band entropy coded bit stream and the second segment of a band entropy coded bit stream into a resultant band entropy coded bit stream.

12. The method of claim 10, wherein the predetermined block size is fixed.

13. The method of claim 10, wherein the predetermined block size is variable.

5 14. A method of decoding the at least two band entropy coded bit streams into a Source Data bit stream, the method comprising the steps of:

10 Entropy decoding each of the at least two band entropy coded bit streams separately and in parallel using a standard entropy decoder to generate at least two band entropy decoded bit streams, each of the at least two band entropy coded bit streams decoded with an entropy band decoder having a predetermined entropy designation, the predetermined entropy designation selected from an entropy designation range between a predetermined high entropy designation and a predetermined low entropy designation;

15 Merging each of the at least two band entropy decoded bit streams together to generate a Source Data bit stream.

Figure 1: Bit Stream Sliced into Entropy Bands, Multiplexed and Block-Sliced and Encrypted in a Process Based on any Entropy and Encryption Coding Standards

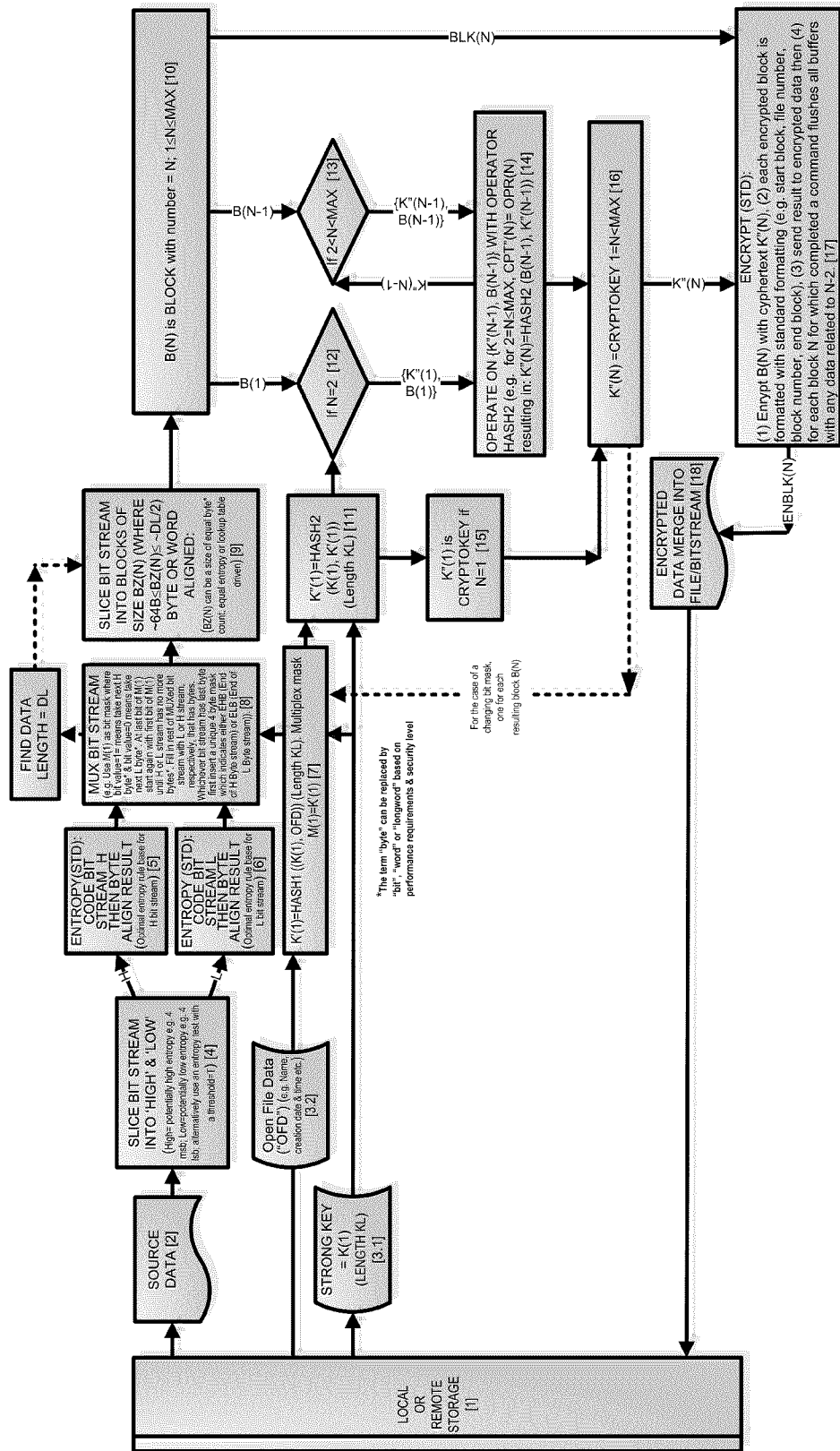
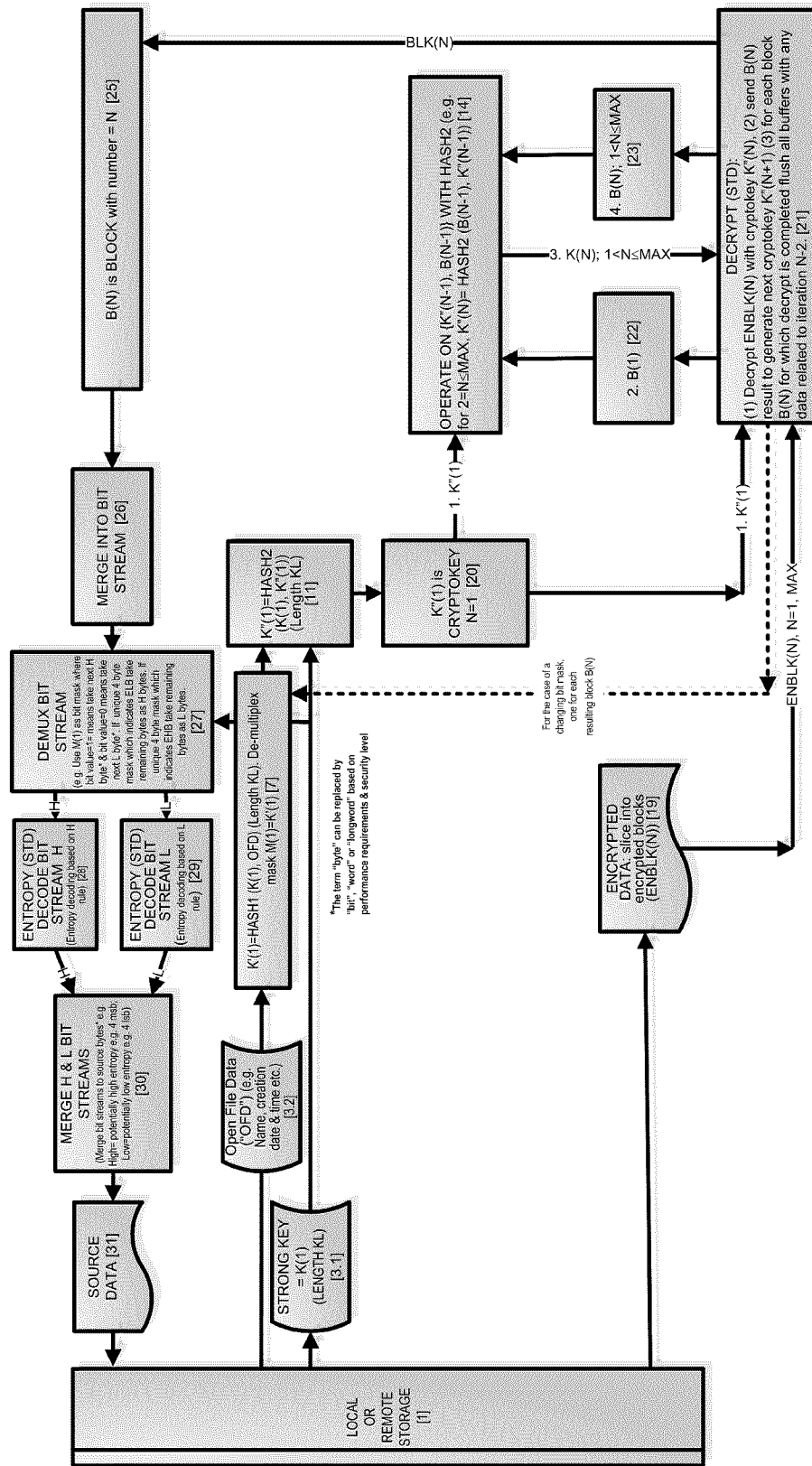


Figure 2: Bit Stream is Decrypted, Block-Adjoin, Demultiplex and Entropy-Band-Adjoined in a Decryption Process Based on any Encryption and Entropy Coding Standards



INTERNATIONAL SEARCH REPORT

International application No.

PCT/CA2013/050969

A. CLASSIFICATION OF SUBJECT MATTER

IPC: **H03M 7/40** (2006.01), **G06F 21/62** (2013.01), **H04L 9/28** (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H03M 7/40 (2006.01), G06F 21/62 (2013.01), H04L 9/28 (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)

Questel Orbit: encryption, two band, entropy, entropy designation, hash, entropy coding

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2012/172115A1, (GEORGE et al.) 20 December 2012 (20-12-2012) *Entire document*	1-6 and 14
A	WO 2013/050612A1, (BARTNIK et al.) 11 April 2013 (11-04-2013) *Entire document*	1-6 and 14
A	US 8,487,791B2, (KORODI et al.) 16 July 2013 (16-07-2013) *Entire document*	1-6 and 14

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* "A" "E" "L" "O" "P"	Special categories of cited documents: document defining the general state of the art which is not considered to be of particular relevance earlier application or patent but published on or after the international filing date document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) document referring to an oral disclosure, use, exhibition or other means document published prior to the international filing date but later than the priority date claimed	"T" "X" "Y" "&"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art document member of the same patent family
--------------------------------------	--	--------------------------	--

Date of the actual completion of the international search
2 September 2014 (02-09-2014)Date of mailing of the international search report
15 September 2014 (15-09-2014)Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 001-819-953-2476

Authorized officer

Mara Gravelle (819) 934-4893

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of the first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claim Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claim Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Group A – Claims 1-6 and 14: methods for encoding and decoding a two band entropy stream from/to a Source Data bit stream using a standard entropy decoder through a predetermined entropy designation selected from an entropy designation range between a predetermined high entropy designation and predetermined low entropy designation.

Group B – Claims 7-13: methods for encoding and decoding sequential blocks comprising: the use of a unique encryption key and generating a unique strong encryption key for a current block from a hash of a previous unique encryption key associated with a previous block.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claim Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim Nos.:

1-6

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/CA2013/050969

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
WO2013050612A1	11 April 2013 (11-04-2013)	WO2013050612A1 EP2764692A1 US2014210652A1	11 April 2013 (11-04-2013) 13 August 2014 (13-08-2014) 31 July 2014 (31-07-2014)
WO2012172115A1	20 December 2012 (20-12-2012)	WO2012172115A1 AP201407360D0 AP201407361D0 AU2012268950A1 AU2012268951A1 CA2839560A1 CA2839569A1 CN103733622A CN103748886A CN103931194A CO6852030A2 CO6852031A2 EP2721819A1 EP2721820A1 EP2721822A1 JP2014518473A KR20140022957A KR20140028106A MX2013014867A MX2013014868A US2014140400A1 US2014177707A1 US2014198841A1 WO2012172113A1 WO2012172114A1	20 December 2012 (20-12-2012) 31 January 2014 (31-01-2014) 31 January 2014 (31-01-2014) 30 January 2014 (30-01-2014) 16 January 2014 (16-01-2014) 20 December 2012 (20-12-2012) 20 December 2012 (20-12-2012) 16 April 2014 (16-04-2014) 23 April 2014 (23-04-2014) 16 July 2014 (16-07-2014) 30 January 2014 (30-01-2014) 30 January 2014 (30-01-2014) 23 April 2014 (23-04-2014) 23 April 2014 (23-04-2014) 23 April 2014 (23-04-2014) 28 July 2014 (28-07-2014) 25 February 2014 (25-02-2014) 07 March 2014 (07-03-2014) 21 May 2014 (21-05-2014) 28 May 2014 (28-05-2014) 22 May 2014 (22-05-2014) 26 June 2014 (26-06-2014) 17 July 2014 (17-07-2014) 20 December 2012 (20-12-2012) 20 December 2012 (20-12-2012)
US8487791B2	16 July 2013 (16-07-2013)	US2011200104A1	18 August 2011 (18-08-2011)