(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 February 2010 (25.02.2010)

PCT

(10) International Publication Number
WO 2010/020264 A1

(51) International Patent Classification:
H04L 1/00 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/EP2008/006832

(22) International Filing Date:
20 August 2008 (20.08.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NEC EUROPE LTD.** [DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BOHLI, Jens, Matthias** [DE/DE]; O3, 1, 69161 Mannheim (DE). **HESSLER, Alban** [CH/DE]; Kettengasse 2, 69117 Heidelberg (DE). **UGUS, Osman** [TR/DE]; Elfendgrund 8a, 64367 Mühltal (DE). **WESTHOFF, Dirk** [DE/DE]; Villenstrasse 36a, 67433 Neustadt an der Weinstrasse (DE).

(74) Agent: **ULLRICH & NAUMANN**; Luisenstrasse 14, 69115 Heidelberg (DE).

(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, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: METHOD FOR CODED DATA TRANSMISSION AND APPARATUS FOR DECODING CODED DATA

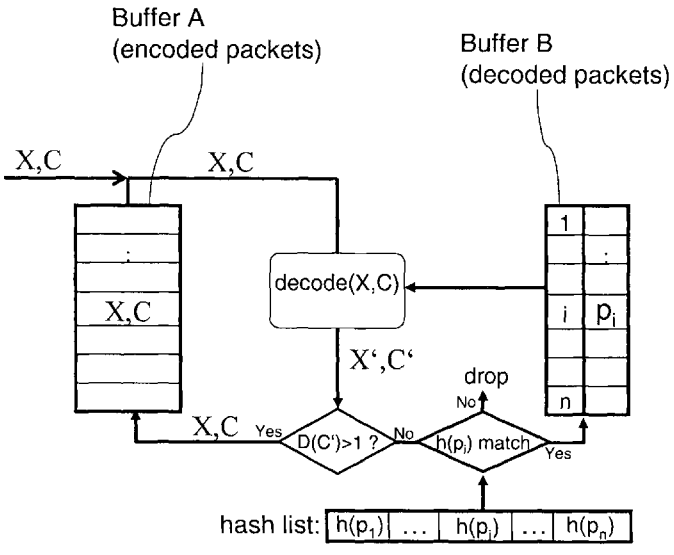

Fig. 2

(57) Abstract: A method for coded data transmission from a sender to at least one receiver, wherein the transmission data is divided into single source packets p_i, the method comprising the steps of generating encoded packets from said source packets p_i by applying rateless erasure codes, transmitting said encoded packets to said at least one receiver, and decoding said encoded packets in order to obtain said source packets p_i by applying a decoding process with on-the-fly decodability functionality, wherein already decoded source packets p_i are employed by the decoding process for decoding remaining encoded packets, the method being characterized in that an integrity check is performed on source packets p_i obtained in the decoding process before accepting said source packets p_i for being employed by the decoding process for further decoding. Furthermore, an apparatus for decoding coded data is disclosed.

WO 2010/020264 A1

METHOD FOR CODED DATA TRANSMISSION AND APPARATUS FOR DECODING CODED DATA

The present invention relates to a method for coded data transmission from a sender to at least one receiver, wherein the transmission data is divided into single source packets p_i, the method comprising the steps of generating encoded packets from said source packets p_i by applying rateless erasure codes, transmitting said encoded packets to said at least one receiver, and decoding said encoded packets in order to obtain said source packets p_i by applying a decoding process with on-the-fly decodability functionality, wherein already decoded source packets p_i are employed by the decoding process for decoding remaining encoded packets.

Furthermore, the invention relates to an apparatus for decoding coded data, wherein said coded data has been created by dividing source data into single source packets p_i and by generating encoded packets from said source packets p_i by applying rateless erasure codes, comprising reception means for receiving said encoded packets from a sender of said encoded packets, and a decoding device with on-the-fly decodability functionality for decoding said encoded packets in order to obtain said source packets p_i, the decoding device being configured to employ already decoded source packets p_i for decoding remaining encoded packets.

Rateless Erasure Codes, which are also known as Fountain Codes, enable a sender to generate a potentially unlimited sequence of encoding symbols from a given set of source symbols allowing the receiver to recover the original source symbols from any subset of the encoding symbols of size equal or slightly larger than the number of source symbols. Fountain Codes are beneficial since they do not require the receivers to inform the sender about individual missing packets. If due to an unreliable (e.g. wireless) medium, over which packets are transmitted, packets get lost, it is not required at the sender side to resubmit the missing packet, say X_j. Instead, due to the "fountain" characteristic of the code applied at the sender side, just another random linear combination X_{j+k} can be transmitted.

Consequently, rateless erasure codes provide an essential unidirectional communication protocol as no acknowledgment in case of unsuccessful decoding is

required. Application is therefore particularly beneficial in e.g. wireless cellular broadcasting networks which do not have a feedback channel.

However, rateless erasure codes typically do not consider malicious modification of coded data, e.g. during transmission. Depending on the application area this can be an important issue. Particularly in open, public, untrusted, or even hostile environments, a secure coded data transmission is essential. Potentially, a single malicious packet can destroy the whole data set during decoding, and, if not recognized, the error can spread and prevent successful decoding for all receivers. This allows even the limited adversary by inserting few malicious packets, to poison a sufficient amount of data to let the decoding process fail. In principle, a single bogus packet can stop the whole decoding process what is known as the so-called poisoning attack.

State of the art solutions for the poisoning attack involve public-key primitives such as homomorphic hash functions or homomorphic signatures. However, those methods based on public-key cryptography have a high data overhead and a high computational complexity and are thus not applicable in resource restricted devices, e.g. in wireless sensor networks.

It is therefore an object of the present invention to improve and further develop a method and an apparatus of the initially described type in such a way that, by employing mechanisms that are readily to implement, an efficient security enhancement is brought into Fountain Codes in an energy-efficient way.

In accordance with the invention the aforementioned object is accomplished by a method comprising the features of claim 1. According to this claim, such a method is characterized in that an integrity check is performed on source packets p_i obtained in the decoding process before accepting said source packets p_i for being employed by the decoding process for further decoding.

Furthermore, the aforementioned object is accomplished by an apparatus for decoding coded data comprising the features of independent claim 23. According to this claim, such an apparatus for decoding coded data is characterised in that the

decoding device is configured to perform an integrity check on source packets p_i obtained in the decoding process before accepting said source packets p_i for being employed for further decoding.

According to the invention it has first been recognised that in the context of securing coded data transmission existing mechanisms using public-key primitives are rather complex and are thus not applicable in various scenarios. Furthermore, it has been recognized that due to the fact that even a single malicious source packet p_i included in the decoding process can potentially prevent the whole source information to be obtained correctly, a solution on a per packet basis is valuable.

In contrast to asymmetric approaches of employing complex public-key primitives, the present invention rather pursues symmetric cryptographic primitives using a decoding process with on-the-fly decodability functionality. According to the invention an integrity check is performed on source packets obtained in the decoding process, working on a single-packet basis. More specifically, an individual verification of source packets by means of the integrity check is performed before accepting the source packets for being employed for further decoding. Consequently, only verified information is used to decode a yet unverified encoded packet. If the decoding fails, the error can be localised to be in exactly the currently decoded packet. Poisoning attacks are therefore effectively prevented.

A further advantage of the present invention is that the method and the apparatus are usable in multi-hop network-coding scenarios and, due to the simple operations being involved in the secured decoding process, can also be applied in wireless sensor networks with weak nodes functioning as sender and receiver, respectively, e.g. for secure code update.

In a specific embodiment the rateless erasure code employed for encoding transmission data is a Luby-Transform (LT)-Code, which is the first class of practical fountain codes that is a near optimal erasure correcting code. LT-Codes were first described by Michael Luby in "LT Codes, In Foundations of Computer Science, FOCS 2002, pages 271–282, IEEE, 2002". LT-Codes are beneficial due a provided

Code Design with the least possible amount of packets which can be decoded on-the-fly, i.e. being decoded in the first cycle, without being stored in buffer A.

As regards the encoded packets, each packet may include a code symbol X_n generated as a linear combination of source packets p_i. The generation of linear combinations of source packets p_i enables the sender to produce a potentially unlimited sequence of code symbols in an economic fashion.

The encoding process may comprise two steps. In a first step the code symbols X_n are generated by randomly choosing a packet degree d according to a given degree distribution $\rho(d)$. In which d is the number of source packets p_i that are integrated into an encoded packet. In a second step d is to be chosen uniformly randomly out of a total of K source packets p_i and successively XORing them. This is done for at least $N > K$ encoded packets X_n. The efficiency and quality of LT Codes depends largely on the degree distribution $\rho(d)$ of encoded packets.

According to a preferred embodiment, the encoded packets each include a coefficient vector C_n describing the linear combination, which has been used to compute the respective code symbol X_n. More specifically, in case of the encoded packets being generated in form of linear combinations of source packets p_i, the coefficient vector C_n may indicate the source packets p_i, which have been employed for calculating the respective linear combination X_n. The coefficient vectors C_n can be appended to each encoded packet transmitted from the sender to one or more receivers. Alternatively, the coefficient vectors C_n may be computed simultaneously by sender and receiver using a pseudo-random generator with the same seed.

As regards efficient decoding with on-the-fly characteristic or nearly on-the-fly characteristic, it may be provided that the decoding device, which analyses incoming encoded packets, is configured as to start the decoding process with encoded packets with a coefficient vector C_n of weight $D(C_n)=1$. Encoded packets for which holds $D(C_n)=1$ are actually source packets and can thus be verified in the decoding process directly when received.

Advantageously, the integrity check performed on source packets p_i obtained in the decoding process includes the application of a message authentication code. According to a preferred embodiment, the message authentication code applied to source packets p_i may be based on a hash function. More specifically, the integrity check may include the steps of computing a hash value h of a source packet p_i, and comparing the computed hash value $h(p_i)$ with a corresponding hash value $h(p_i')$, which has been pre-calculated on the side of said sender. If both hash values are consistent with each other, it is ensured that the respective source packet p_i is no bogus packet that might have been inserted by a malicious participant.

To allow for a facile comparison of the two hash values, according to a specific embodiment a hash list containing the pre-calculated hash values $h(p_i)$ of all source packets p_i is generated on the side of the sender, e.g. by the sender itself, and is transmitted to the one or more receivers. Preferably, this transmission is executed before transmitting said source packets p_i.

Advantageously, source packets p_i obtained in said decoding process, which have passed said integrity check successfully, are inserted into a first buffer B. When all source packets p_i have been successfully recovered and verified by means of the integrity check, the source packets p_i can be retrieved from buffer B and can be put together again to form the original transmission data. Contrary, source packets p_i obtained in the decoding process, which have failed to pass the integrity check successfully, may be dropped.

According to a preferred embodiment, the decoding process employs source packets p_i from the first buffer B to decode remaining encoded packets. As only source packets p_i that have been successfully verified by means of the integrity check are inserted into the first Buffer B, it is ensured that the remaining decoding process will be always performed on correct source packets p_i such that the correctness of the decoding process is approved. Consequently, the poisoning problem is effectively avoided.

With respect to an efficient continuation of the decoding process, it may be provided that encoded packets, which failed to be decoded in the decoding process, are

inserted into a second buffer A. According to a preferred embodiment a buffer management is applied on the second buffer A, which aims at avoiding an overflow of that buffer. An overflow could result from a situation in which an attacker changes coefficient vectors of encoded packets. For instance, the introduction of coefficient vectors C_i' with $D(C_i')>D(C_i)$ constitutes a DoS (Denial of Service) attack as these packets will circulate several times between the second buffer A and the decoding process before the final verification drops the packet and prevents it from being stored in the first buffer B.

Advantageously, buffer management includes the steps of monitoring the second buffer A, and in case monitoring yields that the second buffer A is full, dropping randomly encoded packets from the second buffer A. Alternatively, instead of a random dropping, a buffer management may be provided according to which the encoded packet with the highest weight coefficient vector C_i is dropped. Such dropping is most efficient as these high weighted encoded packets would circulate with a high probability unacceptably long between the second buffer A and the decoding process. According to still another embodiment, the selection criterion for packet dropping management may be an adjustment of the packets buffered in the second buffer A to an expected weight distribution. The expected weight distribution will typically correspond with the specific weight distribution employed as input parameter for the rateless erasure code. In any case it proves to be advantageous that the applied selection criterion itself, once known to an attacker, should not give additional attack possibilities.

As regards rateless erasure codes, they provide an essentially one-way communication protocol as no acknowledgment in case of unsuccessful decoding is required. Thus application is beneficial in e.g. wireless cellular broadcasting networks which do not have a feedback channel. Therefore, a wireless broadcast channel is employed for coded data transmission from sender to at least one receiver.

In a particularly preferred embodiment the sender of the encoded data is a sensor node of a wireless sensor network (WSN), in particular a sink node. In such a scenario, the transmission data advantageously includes program data, which is encoded and transmitted from the sink node to other sensor nodes of the sensor

network that function as receivers. By applying the method according to the present invention in such a way, a secure remote code image update mechanism for wireless sensor networks to update code on sensor networks that are not easily accessible is provided. The possibility of securely reprogramming the sensor nodes after the development greatly reduces maintenance costs by offering a cheap way of updating the program. In particular, it may be provided that in a multi-hop scenario, in which certain sensor nodes retransmit received information to further sensor nodes of the network, the secured decoding process (and subsequent re-encoding) is applied several times, in order to benefit from the advantages of Fountain Codes not only at the first hop and to efficiently avoid poisoning attacks during the entire distribution process.

Furthermore, in case of applying LT-Codes, the process can be considered to be extremely efficient, since the decoding is mainly based on XOR operations. Consequently, when applying the LT decoding algorithm to the area of sensor networks, the limiting factor on a sensor node is the necessary data overhead, and the buffer size in the buffers A and B, and not so much the computational complexity.

There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end, it is to be referred to the patent claims subordinate to patent claims 1 and 23 and to the following explanation of a preferred example of an embodiment of the invention, illustrated by the figure. In connection with the explanation of the preferred example of an embodiment of the invention by the aid of the figure, generally preferred embodiments and further developments of the teaching will be explained.

In the drawings:

Fig. 1 is a schematic view of a decoding process of an LT-Code with on-the-fly characteristic,

Fig. 2 is a schematic view of a decoding process of an LT-Code with on-the-fly characteristic, including an integrity check according to an embodiment of the present invention,

Fig. 3 is a schematic view of the decoding process of Fig. 2, further including a buffer management for avoiding DoS attacks, and

Fig. 4 is a schematic view of a preferable packet structure for distributing a hash list for performing an integrity check.

Fig. 1 illustrates – schematically – the principil decoding process of an LT-Code with on-the-fly characteristic. LT Codes are the first full realization of the idea of rateless erasure codes, which are also known as Fountain Codes. The basic principle of LT Codes for data transmission over a wireless interface can be described as follows. The data block to be transmitted is separated into K packets each of size l. An encoded packet is computed in two steps:

1) A packet degree d is randomly chosen according to a given distribution. The choice of the weight distribution is the key parameter with respect to the performance and the efficiency of the coding scheme $\rho(d)$.

2) The encoded packet is obtained by choosing uniformly randomly d out of the K source packets, namely $\{p_{l1}, \ldots, p_{ld}\}$, and successively XORing them to compute

$$X = \bigoplus_{i=1}^{d} p_{l_i}$$

This is done for at least $N > K$ encoded packets X. The information which packets p_{li} have been considered for a concrete encoded packet X_j is represented in a coding vector C_j of size $1 \times K$. This information is necessarily needed for the decoding process. The coding vector can be appended to each packet, or computed simultaneously by sender and receiver using a pseudo-random generator with the same seed.

Subsequent to the encoding process, $N > K$ packets are sent. The packets contain the encoded vector X and potentially information about the coefficient vector C. In any case, the receiver extracts the pair (X, C) of size $l + K$ from the packet.

The LT decoding process includes two buffers. More detailed, the decoding process uses a buffer A, where not yet decoded packets are stored, and a buffer B for decoded information. Encoded packets X, C received over the wireless are decoded in accordance to the information in the coding vector C. The decoding device, which analyses incoming encoded packets, is configured as to start the decoding process with encoded packets with a coefficient vector C_j of weight $D(C_j) = 1$. Encoded packets for which holds $D(C_j) = 1$ are actually source packets p_j and are put in the buffer B and used to decode further packets in buffer A. If the packet still has a coefficient vector C_j with weight $D(C_j) > 1$ it is saved in the buffer A of encoded packets.

Those plaintext packets p_i which are already stored in the buffer B, and which are relevant according to the currently processed packet (X, C), will be applied to the actual decoding. The exact processing of the box 'Decode' is as follows: Incoming packets X_j, C_j are partially decoded for all known source packets p_i, i.e.

$$X'_j = X_j \oplus p_i$$
$$C'_j = C_j - e_i,$$

where e_i is the i-th unit vector.

The buffer B stores all the actually decoded plaintext packets p_i. If a new element is added to B, all (X, C) in buffer A will again be applied to the decoding process.

Fig. 2 illustrates the LT decoding process described in Fig. 1, including an integrity check according to an embodiment of the present invention. More precisely, the integrity check includes the application of a message authentication code. In the specific case shown in Fig. 2, the message authentication code is based on the application of a hash function.

More specifically, the security enhancements during the decoding process at the receiver side are as follows: Before a decoded packet p_i is written into the buffer B, the relevant hash value $h(p_i)$ of a previously sent hashlist is compared with the hash value computed by hashing $h(X)$:

$$h(p_i) = h(X),$$

with $D(C) = 1$ and $C[i] = 1$. This ensures that no bogus packet is stored in the buffer B ensuring that the remaining decoding process can be performed on correct packets p_i, such that the correctness of the decoding process is approved.

Fig. 3 illustrates the decoding process of Fig. 2, further including a buffer management for avoiding DoS attacks. The application of a message authentication code, e.g. performing the comparison $h(p_i) = h(X)$ as described above ensures that no incoming bogus packet X_i, C_i can negatively effect the decoding process of subsequently incoming packets; however, since the initially sent hashlist contains hash values $h(p_i)$, which are purely computed on plaintext packets p_i and in particular not on C_i – which is also sent over the wireless –, the DoS attack described in the following is still possible: a malicious participant can modify a packet X_i, C_i to $X_i, \hat{C}_i$ by modifying the coefficient vector itself. To perform a DoS attack, which substantially delays the whole decoding process, an attacker could choose a $\hat{C}_i$ with $D(\hat{C}_i) > D(C_i)$. Such an attack results in a situation in which $X_i, \hat{C}_i$ will circulate several times between buffer A and the decoding process before the final hash verification drops the packet and prevents it from the storage in B. Even worse, if the attacker sends several of such corrupted packets $X_i, \hat{C}_i$ with high degrees $D(\hat{C}_i)$, the buffer B will not be able to store enough correct packets with the consequence that the decoding process is delayed or finally fails.

To overcome this situation, another decision point before storing an already partly decoded packet X_k, C_k into the buffer A is introduced. A selection criterion to overwrite a packet in case of a full buffer A ideally has to fulfill two requirements: Firstly, it should be smart enough to skip such an encoded packet which anyhow would circulate with a high probability unacceptable long between buffer A and the decoding process; Secondly, the selection criterion itself, once known by an attacker,

should not give additional attack possibilities. Possible selection criterions may be *random*, *maximum weight*, and *adjust to expected weight distribution*.

To conclude, the mechanism verifying decoded packets on basis of the initially sent hash value list ensures that only correct packets will be considered for the page reconstruction and the remaining decoding process; packet filtering based on criteria as mentioned above ensures that bogus packets will not interfere the on-the-fly decoding process. An attacker aiming at a DoS attack will ideally drop packets X_i,C_i with $D(C_i) = 1$ during the starting phase of the transmission, whereas he will modify packets X_i,C_i to $X_i,\hat{C}_i$ with $D(\hat{C}_i) = n$ (with n high) or at least $D(\hat{C}_i) > D(C_i)$.

Fig. 4 illustrates a preferable packet structure to distribute the hash list for performing the integrity check on source packets p_i obtained in the decoding process. For realizing the shown packet structure, the sender first has to encode the transmission data, wherein the full transmission data is distributed into pages, depending on the memory availability at receiver side. It is to be noted that only one page is required if the receiver has full power. Every page is split up into m packets depending on the network's payload size.

The data structure for authentication is a combination of a *hash chain* with h_1 securing the pages and a *hash tree of hash chains* with h_2 securing the individual packets. The root of the hash tree and the hash chain are signed with a *public-key signature* scheme. The first page (*P0*) of the encoded data contains the signature and the hash tree, to set up the prerequisites for the receivers to setup the enhanced decoding algorithm. The first page can therefore not be transmitted via fountain codes, but e.g. by standard Deluge (as described e.g. in Jonathan W. Hui and David Culler, The dynamic behaviour of a data dissemination protocol for network programming at scale. In *Embedded networked sensor systems, SenSys '04*, pages 81–94. ACM, 2004.) Actually, considering the hash tree, the transmission of the first page follows closely Seluge (as described e.g. in An Liu, Young-Hyun Oh, and Peng Ning, Secure and dos-resistant code dissemination in wireless sensor networks using seluge. In *Information Processing in Sensor Networks, IPSN 2008*, pages 561–562. IEEE, 2008.). This leads to the packet structure as depicted in Fig. 4. The dark colored blocks represent a hash chain containing hashes (h_1) of the full pages, while

the hatched blocks are a hash tree of hash chains that contain hashes (h_2) of the individual packets.

As shown in Fig. 4, the first page (*P0*) is a hash tree of hash chains with a signed root that facilitate the authentication of the second page and thus via the hash chain the complete code image. Let *w* be the number of hash values that can fit a single packet. The hash tree in page *P*0 is created such that a packet at level *i* comprises hash values of *w* packets at level *i + 1*, where the hash values of the packets of the page *P1* compose the packets at the last level. When the construction of the hash tree is completed, the sender signs the root packet together with the hash of the first data page *h*1(*P1*) and code related header information such as the version number. Page *P0* is constructed in a way, that after the signature is received, every following packet can be checked against a hash value.

As regard the encoding strategy, the sender prepares the packet structure prior to the start of the code image transmission as described above. The protocol starts transmitting the first page with Deluge (respectively Seluge), i.e. no fountain coding techniques are applied yet. Once the receivers received successfully page *P*0, the values for the next page's packets are set up and the transmission of the next page starts using Fountain Codes. The sender transmits now page by page encoded. Therefore, the sender builds randomly encoded packets according to the weight distribution and sends $N > K$ packets. The value $\alpha = N - K$ depends on the channel quality and adversarial interference and can by adapted dynamically. If after *N* transmitted packets, receivers still have not successfully decoded the actual transmitted page, they send a NACK and the sender continuous sending another β encoded packets. Receivers that have already decoded stop listening.

Many modifications and other embodiments of the invention set forth herein will come to mind the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are

employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. Method for coded data transmission from a sender to at least one receiver, wherein the transmission data is divided into single source packets p_i, the method comprising the steps of

generating encoded packets from said source packets p_i by applying rateless erasure codes,

transmitting said encoded packets to said at least one receiver, and

decoding said encoded packets in order to obtain said source packets p_i by applying a decoding process with on-the-fly decodability functionality, wherein already decoded source packets p_i are employed by the decoding process for decoding remaining encoded packets,

characterized in that an integrity check is performed on source packets p_i obtained in the decoding process before accepting said source packets p_i for being employed by the decoding process for further decoding.

2. Method according to claim 1, wherein said rateless erasure code employed for encoding said transmission data is a Luby-Transform (LT)-Code.

3. Method according to claim 1 or 2, wherein said encoded packets each include a code symbol X_n generated as a linear combination of said source packets p_i.

4. Method according to claim 3, wherein said code symbols X_n are generated by

randomly choosing a packet degree d according to a given degree distribution $\rho(d)$, and

choosing uniformly randomly d out of a total of K source packets p_i and successively XORing them.

5. Method according to any of claims 1 to 4, wherein said encoded packets each have associated a coefficient vector C_n describing the linear combination, which has been used to compute respective code symbol X_n.

6. Method according to claim 5, wherein said coefficient vectors C_n are appended to said encoded packets transmitted from said senders to said at least one receiver.

7. Method according to claim 5, wherein said coefficient vectors C_n are computed by said sender and by said at least one receiver using a pseudo-random generator with the same seed.

8. Method according to any of claims 5 to 7, wherein incoming encoded packets are analysed by said decoding process, wherein the decoding of encoded packets starts with encoded packets with a coefficient vector C_n of weight $D(C_n)=1$.

9. Method according to any of claims 1 to 8, wherein said integrity check performed on source packets p_i obtained in said decoding process includes the application of a message authentication code.

10. Method according to any of claims 1 to 9, wherein said integrity check performed on source packets p_i obtained in said decoding process includes the steps of

computing a hash value h of said source packet p_i, and

comparing said computed hash value $h(p_i)$ with a corresponding hash value $h(p_i)$, which has been pre-calculated on the side of said sender.

11. Method according to claim 10, wherein a hash list containing said pre-calculated hash values $h(p_i)$ of all source packets p_i is generated by said sender and is transmitted to said at least one receiver, preferably before transmitting said source packets p_i.

12. Method according to any of claims 1 to 11, wherein source packets p_i obtained in said decoding process, which have passed said integrity check successfully, are inserted into a first buffer (B).

13. Method according to any of claims 1 to 12, wherein source packets p_i obtained in said decoding process, which have failed to pass said integrity check successfully, are dropped.

14. Method according to claim 12, wherein said decoding process employs source packets p_i from said first buffer (B) to decode remaining encoded packets.

15. Method according to any of claims 1 to 14, wherein encoded packets, which fail to be decoded in the decoding process, are inserted into a second buffer (A).

16. Method according to claim 15, wherein a buffer management is applied on said second buffer (A), which aims at avoiding an overflow of said second buffer A.

17. Method according to claim 16, wherein said buffer management includes the steps of

monitoring said second buffer (A), and

in case said monitoring yields that said second buffer (A) is full, dropping randomly encoded packets from said second buffer (A).

18. Method according to claim 16, wherein said buffer management includes the steps of

monitoring said second buffer (A), and

in case said monitoring yields that said second buffer (A) is full, dropping the encoded packet with the highest weight of said coefficient vector C.

19. Method according to claim 16, wherein said buffer management includes the steps of

monitoring said second buffer (A), and

in case said monitoring yields that said second buffer (A) is full, dropping encoded packets in such a way that the packets buffered in said second buffer (A) become adjusted to an expected weight distribution.

20. Method according to any of claims 1 to 19, wherein a wireless broadcast channel is employed for said coded data transmission from said sender to said at least one receiver.

21. Method according to any of claims 1 to 20, wherein said sender is a sensor node of a sensor network, in particular a sink node or a base station.

22. Method according to claim 21, wherein said transmission data includes program data, which is encoded and transmitted from said sink node to other sensor nodes of the sensor network.

23. Apparatus for decoding coded data, wherein said coded data has been created by dividing source data into single source packets p_i and by generating encoded packets from said source packets p_i by applying rateless erasure codes, comprising

reception means for receiving said encoded packets from a sender of said encoded packets, and

a decoding device with on-the-fly decodability functionality for decoding said encoded packets in order to obtain said source packets p_i, the decoding device being configured to employ already decoded source packets p_i for decoding remaining encoded packets,

characterized in that the decoding device is configured to perform an integrity check on source packets p_i obtained in the decoding process before accepting said source packets p_i for being employed for further decoding.

24. Apparatus according to claim 23, wherein said decoding device comprises a Luby-Transform (LT)-Decoder.

25. Apparatus according to claim 23 or 24, wherein said decoding device is configured to perform said integrity check on source packets p_i obtained in the decoding process by applying a message authentication code.

26. Apparatus according to any of claims 23 to 25, wherein said decoding device is configured to perform said integrity check on source packets p_i obtained in the decoding process by

computing a hash value h of said source packet p_i, and

comparing said computed hash value $h(p_i)$ with a corresponding hash value $h(p_i)$, which has been pre-calculated on the side of said sender.

27. Apparatus according to any of claims 23 to 26, comprising a first buffer (B) for buffering source packets p_i obtained in said decoding process, which have passed said integrity check successfully.

28. Apparatus according to any of claims 23 to 27, comprising a second buffer (A) for buffering encoded packets, which failed to be decoded in said decoding process.

29. Apparatus according to any of claims 23 to 28, the apparatus being designed as a part of a sensor node's hardware.

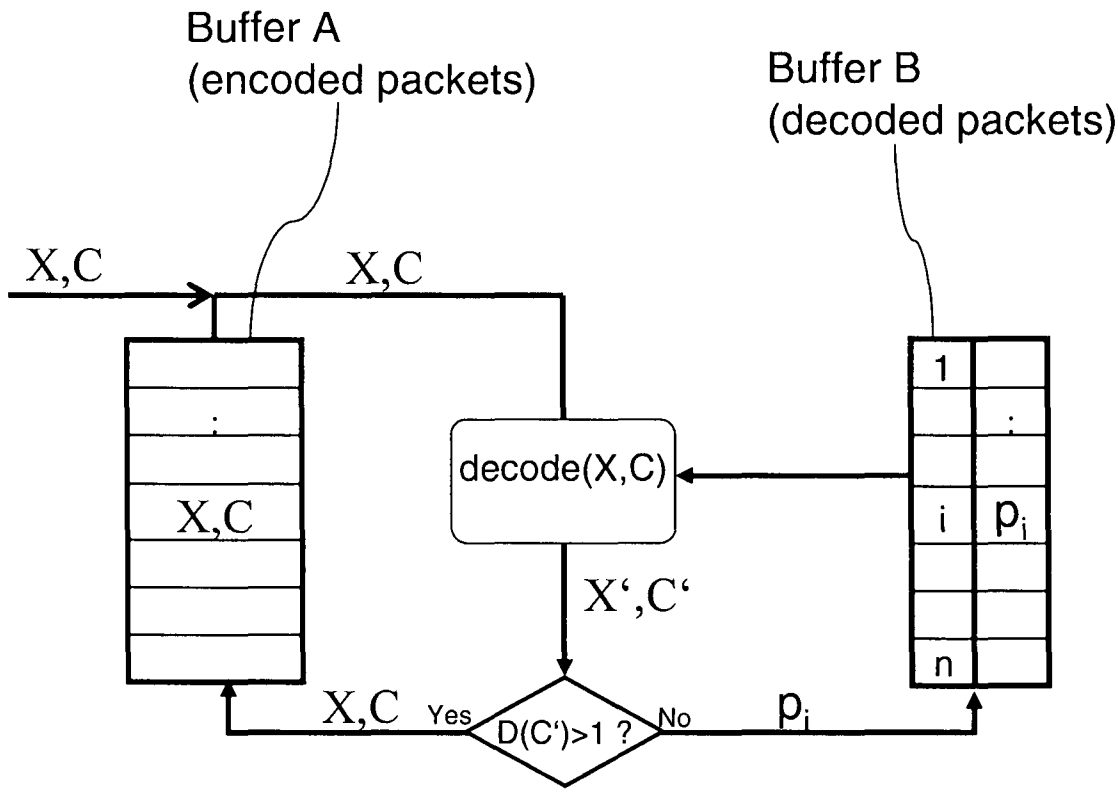

Fig. 1
(State of the Art)

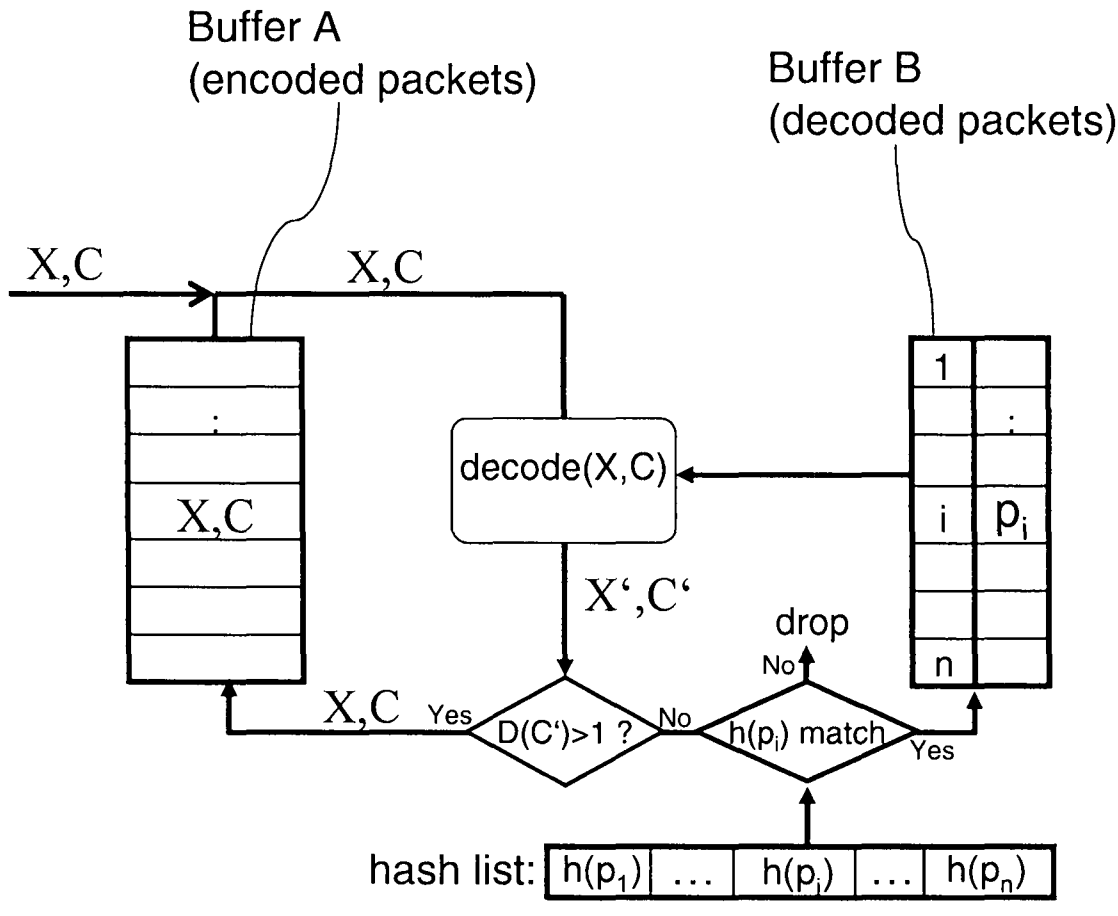
Buffer A
(encoded packets)
Buffer B
(decoded packets)
X,C
X,C
X,C
decode(X,C)
X‘,C‘
drop
No
X,C
Yes
D(C‘)>1 ?
No
h(p_i) match
Yes
1
i
p_i
n
hash list:
h(p_1)
...
h(p_i)
...
h(p_n)

Fig. 2

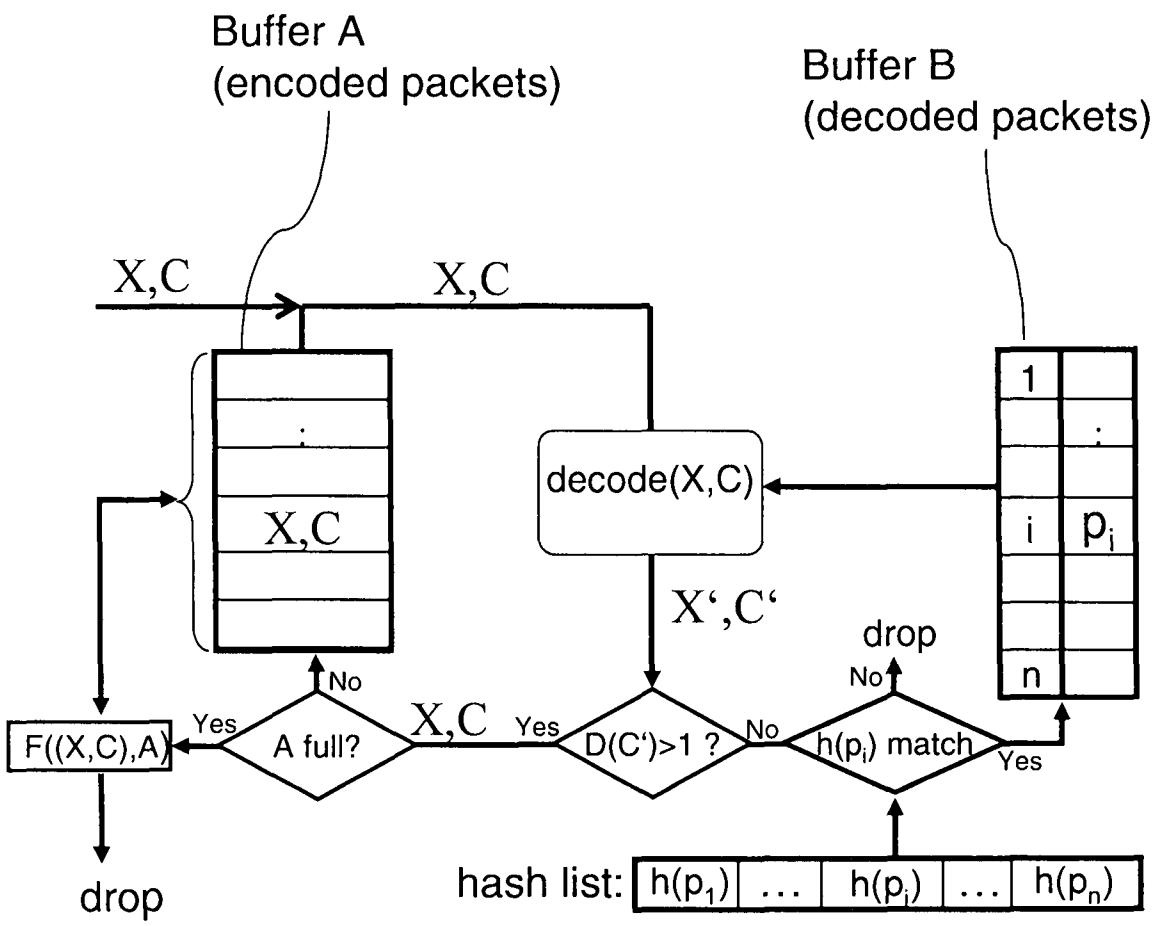
Buffer A
(encoded packets)
Buffer B
(decoded packets)
X,C
X,C
X,C
decode(X,C)
X‘,C‘
1
i
p_i
n
No
Yes
A full?
F((X,C),A)
drop
X,C
Yes
D(C‘)>1 ?
No
drop
No
$h(p_i)$ match
Yes
hash list:
$h(p_1)$
...
$h(p_i)$
...
$h(p_n)$

Fig. 3

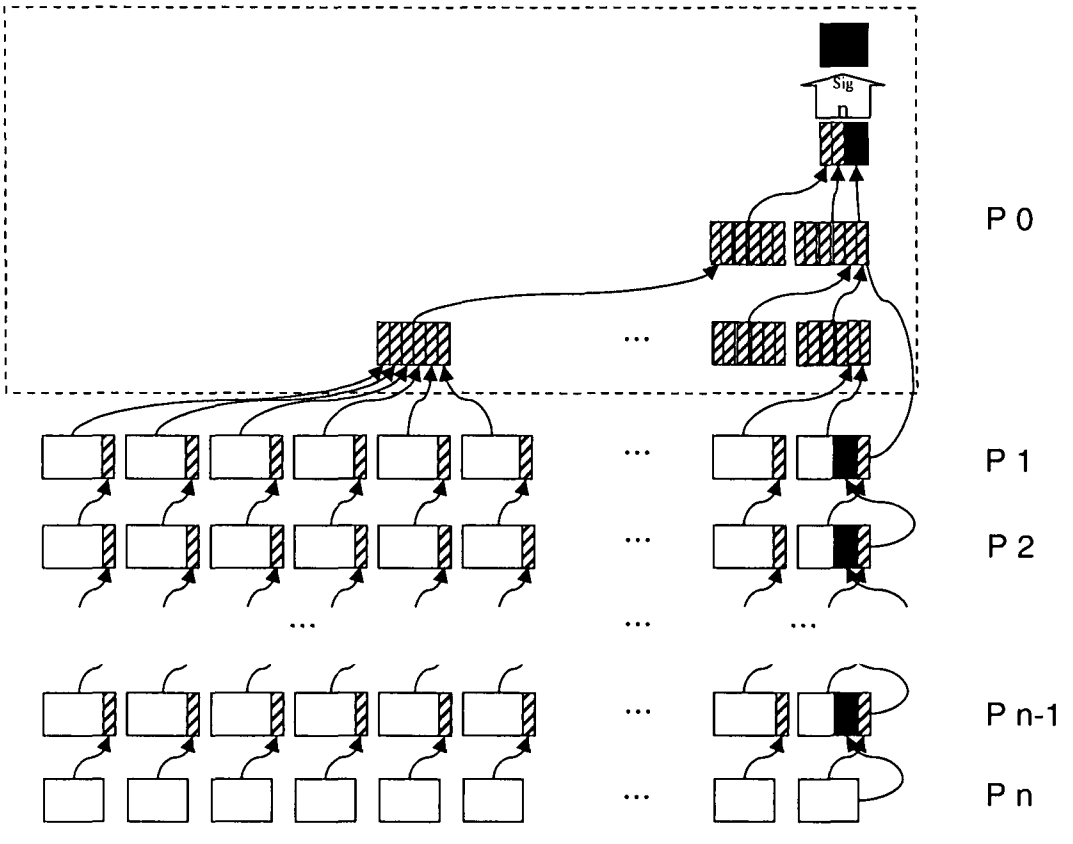

Fig. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/006832

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L1/00 H04L29/06

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)
H04L

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KROHN M N ET AL: "On-the-fly verification of rateless erasure codes for efficient content distribution" SECURITY AND PRIVACY, 2004. PROCEEDINGS. 2004 IEEE SYMPOSIUM ON BERKELEY, CA, USA 9-12 MAY 2004, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 9 May 2004 (2004-05-09), pages 226-240, XP010768048 ISBN: 978-0-7695-2136-7	1-14, 20-27,29
A	abstract * sections I,II,III * figure 1	15-19,28

-/--

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" 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)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" 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

"X" 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

"Y" 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	Date of mailing of the international search report
24 July 2009	31/07/2009

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Toumpoulidis, T

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/006832

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SANGWON HYUN ET AL: "Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks" INFORMATION PROCESSING IN SENSOR NETWORKS, 2008. IPSN '08. INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 22 April 2008 (2008-04-22), pages 445-456, XP031246731 ISBN: 978-0-7695-3157-1 the whole document	1-29

Form PCT/ISA/210 (continuation of second sheet) (April 2005)