Multilevel Network Steganography in Fountain Codes

Jörg Keller
FernUniversität in Hagen
Faculty of Mathematics and Computer Science
Hagen, Germany
joerg.keller@fernuni-hagen.de

ABSTRACT

We present a method to establish a network storage covert channel in a fountain code, which is used to provide reliable communication over lossy network with low overhead and without acknowledgment. As also parts of the secret message get lost when a carrier packet is lost, reliable transmission of the secret message is provided by using a second fountain code. Thus, our proposal opens the possibility for a multilevel steganographic method. We evaluate a proof-of-concept implementation that uses LT-codes and demonstrate that activation of the covert channel does not deteriorate the reliability of the carrier. We also discuss countermeasures that limit the possibilities for covert channels in fountain codes.

CCS CONCEPTS

• **Security and privacy** → **Hash functions and message authentication codes**; Mathematical foundations of cryptography; • **Theory of computation** → *Cryptographic primitives.*

KEYWORDS

Network Steganography, Fountain Code, Network Storage Covert Channel, Multi-level Steganography

ACM Reference Format:
Jörg Keller. 2021. Multilevel Network Steganography in Fountain Codes. In *European Interdisciplinary Cybersecurity Conference (EICC), November 10–11, 2021, Virtual Event, Romania.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487405.3487420

1 INTRODUCTION

Fountain codes [3] are used to encode and transmit source packet data over unreliable networks with notable packet loss such as wireless sensor networks. Their name is derived from the property that they can produce a potentially infinite sequence of encoded packets from the source packets. Their advantages are low overhead, i.e., if p packets must be sent, then receiving any p (or slightly more) encoded packets is sufficient to reconstruct the original packets. Thus, they also do not need a back channel for acknowledgment packets. Fountain codes are also used in other domains such as encoding page blocks in SSD storage, but we will focus on their use in networks.

EICC, November 10–11, 2021, Virtual Event, Romania

ACM ISBN 978-1-4503-9049-1/21/11.
https://doi.org/10.1145/3487405.3487420

As wireless sensor networks are used also for sensitive data, an attacker may have an interest in compromising one sensor node (or several), and exfiltrating some data. A typical means to do this is to use a network covert channel in the sensor node's communication with the network sink [10]. To this end, we present a network covert channel that hides within an LT code [7] as a concrete (and easy to explain) implementation of a fountain code. The covert channel uses the random state/value modulation pattern [17] to inject covert data. As the network covert channel will be affected by packet loss in the carrier network, it should use some means to account for this. As a typical network protocol with acknowledgment scheme is out of reach, it can use the same approach as the carrier: using a fountain code. This in turn opens up an interesting possibility for a multilevel steganography [4], as a second covert channel can be used in that fountain code. As the multilevel scheme is homogeneous, there are only practical restrictions on the number of levels, i.e., bandwidth at some point will be too small to implement a further covert channel with fountain code. We evaluate our proposal with respect to its influence on the performance of the carrier communication, and also discuss possible countermeasures to avoid or at least restrict the use of this type of covert channel.

The remainder of this article is structured as follows. In Section 2 we summarize information on fountain codes and network covert channels, and present related work. In Section 3 we present our proposal for a network covert channel embedded in a fountain code, and extend it to multilevel steganography. Section 4 reports on a prototype implementation and preliminary experiments, while Section 5 analyzes detectability by sketching countermeasures and Section 6 concludes with an outlook on future work.

2 BACKGROUND

2.1 Fountain codes

Fountain codes are rateless erasure codes [3]. To transmit (or store) a fixed set of p source packets of equal size over a channel with notable packet loss, the source packets are encoded by combining packet data into a possibly infinite sequence of encoded packets, and the receiver decodes the encoded packets again into source packets. The encoding is such that from *any* p (or slightly more) encoded packets, the set of source packets can be recovered completely. Thus, there is no need of a control channel from the receiver to the sender for acknowledgment packets.

Luby presented his transform codes (LT codes) [7] as a first realization of fountain codes, and we will use LT codes in the remainder of this paper because their ease of presentation, although there exist more efficient fountain codes, e.g., Raptor codes [13].

Table 1: Degree distributions and subset counts in LT codes with different source packet count

	$p = 16$		$p = 64$	
d	$\rho(d)$	$\binom{p}{d}$	$\rho(d)$	$\binom{p}{d}$
1	0.221	16	0.161	64
2	0.457	120	0.400	2,016
4	0.188	1,820	0.256	635,376
8	0.134	12,870	0.101	4,426,165,368
16	—	—	0.045	488,526,937,079,580
32	—	—	0.037	$\approx 1.83 \cdot 10^{18}$

Rossi et al. and Ugus [12, 15] used LT codes in wireless sensor networks, and Rossi et al. give optimal sparse degree distributions that we will use later on.

Assume that a sender wants to transmit a set of p source packets p_i, where $i \in P$, with $|P| = p$. For simplicity, we will assume $P = \{0, 1, \ldots, p-1\}$ in the following. All packets are bitstrings of equal size. The sender encodes the source packets into a sequence of encoded packets as follows. For encoded packet e_j, a degree d_j is chosen according to a degree distribution $\Omega(d)$. Then, a subset $D_j \subseteq P$ of size $|D_j| = d_j$ is chosen randomly (all $\binom{p}{d_j}$ subsets of size d_j are equally likely). The encoded packet contains data $\bigoplus_{i \in D_j} p_i$, the bitwise or of the source packet data, plus a representation of subset D_j (see below.)

The decoder knows the number of source packets to be transmitted. Upon receiving an encoded packet e_j with subset D_j, the receiver performs the following steps [15]:

- **Step 1:** The receiver stores e_j if it has not yet seen that packet before, i.e. if the receiver has not yet stored packet $e_{j'}$ with subset $D_{j'} = D_j$.
- **Step 2:** The receiver checks if it already possesses an encoded packet $e_{j'}$ with $D_{j'}$, where $D_{j'}$ and D_j only differ by one element $\{k\}$.
- **Step 3:** If yes, then source packet p_k can be recovered by combining the data $\bigoplus_{i \in D_j} p_i$ and $\bigoplus_{i \in D_{j'}} p_i$ from both encoded packets because of $p_i \oplus p_i = 0$. Continue with p_k (encoded with a subset of size 1) from step 1.
- **Step 4:** The receiver checks if it can generate new encoded packets from e_j and packets it has stored, so that the degree of such a new packet is lower than the degrees of the packets it originated from. For each newly generated packet, the receiver starts again with step 1.

The decoder stops if all source packets have been recovered, which is after receiving $(1 + \epsilon)p$ packets with very high probability, for suitable degree distribution Ω. For details about decoder implementations, see e.g. [15].

Table 1 shows degree distributions from [12] and number of possible subsets for $p = 16$ and $p = 64$.

To represent subset D_j, either a bitvector of length p can be used, where bit i is set if and only if $i \in D_j$. We will denote this representation as BV. Alternatively, the degree d_j can be transmitted as a binary number in $\log_2 p$ bits[1], and the subset can be represented explicitly. Either, the index i of each element $i \in D_j$ is given as a binary number, resulting in $d_{max} \cdot \log_2 p$ bits (denoted as SUB), where d_{max} is the maximum possible degree. Alternatively, a suitable enumeration of the subsets of size d_j can be transmitted as a binary number (denoted as ENUM), resulting in $\log_2 \binom{p}{d_j} \leq (p \log_2 p)/2$ bits, as $\binom{p}{d_j} \leq \binom{p}{p/2} < (p/2)^{p/2}$. Please note that this bound is not tight. For example $\binom{64}{32} \approx 1.83 \cdot 10^{18} < 2^{65}$ while $32^{32} = 2^{160}$. Furthermore, if the maximum degree is less than $p/2$, the number of necessary bits is even smaller. Finally, all $s_p = \sum_{d \in D} \binom{p}{d}$ possible subsets can be enumerated as a binary number (denoted as ENUMALL) in the range 0 to $s_p - 1$, i.e., with $\log_2 s_p$ bits, where D is the set of possible degrees.

The choice of representation depends on the circumstances. If the source packets themselves are large compared to their number p, e.g., more than 1000 bits for $p = 64$, then even $p \log p$ bits to represent D_j seem a small overhead. If however each p_i only comprises 128 bits, then an overhead of 64 bits is large. Figure 1 shows the structure and bit size of different encodings of subsets for $p = 16$ and $p = 64$ source packets, respectively. The width of the fields was derived from the values in Table 1 and the explanations above. For example, for $p = 64$ and SUB, each field has a width of $\log_2 p = 6$ bits, except the degree field which has $\log_2(1 + \log_2 p) = 3$ bits. For ENUM, the index field has a width of $\log_2 \binom{64}{32} = 61$ bits, as most subsets are of degree $d = 32$. The offset 1,956 for ENUMALL with $p = 16$ is the sum $\sum_{d=1,2,4} \binom{16}{d}$.

In Section 3 we will introduce a new type of representation of D_j that is slightly larger than the ones before, but has an additional property: if the subsets of each degree are chosen equally likely, then all representations of sets D_j are equally likely.

2.2 Network Covert channels

Steganography is the art of concealing the existence of a secret message in an innocent carrier [10]. For example, the carrier can be a network connection. The secret message can be transmitted via a so-called covert channel, either a storage covert channel, where the bits of the message are represented explicitly in the network packets (e.g., in unused header bits), or a timing covert channel, where the bits of the message are represented by certain temporal relations between network packets (e.g., a gap of 1 second between two successive network packets signals a 1, while a gap of 0.1 seconds signals a 0.) In addition to using steganography, which only hides existence, also the content of the secret message can be protected (and made looking random) by using encryption [11] prior to transmission.

The sender and receiver of the secret message and of the network communication used as a carrier, called covert and overt sender and receiver, respectively, need not be identical, but we will consider them as identical in this work.

A covert channel can be assessed according to its bandwidth, its stealthiness, i.e., the difficulty to detect its existence, and its negative influence on the carrier.

[1] Fewer bits suffice if not all degrees are used. For example, Rossi et al. [12] only use degrees that are powers of two, so that $\log_2(1 + \log_2 p)$ bits suffice. For $p = 64$, the number of bits reduces from 6 to 3.

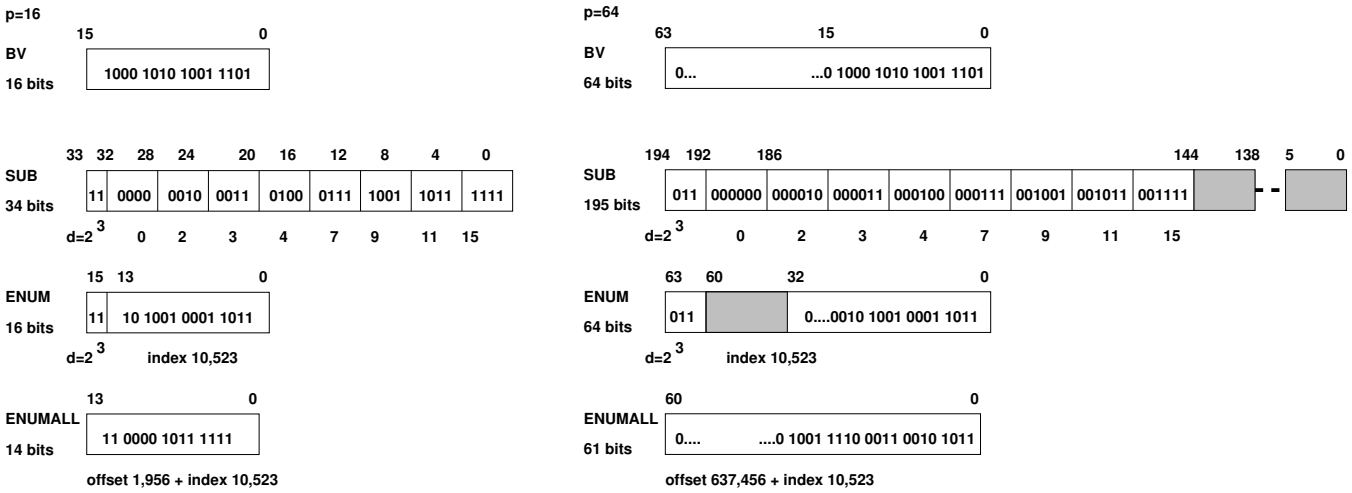

Figure 1: Structure and bit size of subset encodings for $p = 16$ and $p = 64$. As example, the subset $D_j = \{0, 2, 3, 4, 7, 9, 11, 15\}$ of size $d = 8$ is used, with index 10,523 among the subsets of that size. A gray field indicates that its value is not used in this example, but will be necessary to encode a subset of larger size.

There are many proposals for covert channels at all layers of the network stack. To categorize the existing body of work, patterns are used [16, 17]. In the covert channel presented in the next section, a (pseudo-)random value will be replaced by a piece of the secret message, which corresponds to pattern EN4.2 *random state/value modulation* [16], a generalization of the *random (value) modulation* pattern from [17]:

"A (pseudo-)random value or (pseudo-)random state is replaced with a secret message (that is also following a pseudo-random appearance)."

A particular form of steganography, with very few examples in practice, is multilevel steganography [4], where a first covert channel serves as the carrier for a second covert channel.

2.3 Related Work

An example of replacing the pseudo-random content of a network header field with encrypted covert content is given by [8]. They modify the hop count value in IPv6.

To the best of our knowledge, covert channels in fountain codes have not yet been reported in the literature. On the contrary, fountain codes have been mentioned as a means against packet loss within both network storage covert channels [9] and network timing covert channels [1, 6], avoiding usual protocol schemes with acknowledgement packets and re-send of packets. Also, fountain codes have been used in digital media steganography [5].

3 COVERT CHANNELS IN FOUNTAIN CODES

We assume that we are given an LT-code with degree distribution Ω choosing degree $d \in D$ with probability $\rho(d)$, where $D = \{d_1, \ldots, d_{|D|}\}$ is the set of possible degrees d, with $d_1 < d_2 < \cdots < d_{|D|}$. Then the number of possible subsets of size d from the set P of packets with size $p = |P|$ is $n_d = \binom{p}{d}$. We assume representation ENUM, i.e., the degree d is given as a binary number with $\log_2 |D|$ bits, and the subset used is given as a number between 0 and $\binom{p}{d} - 1$ in binary representation with $\log_2 \binom{p}{d_{|D|}}$ bits — if we assume $d_{|D|} \leq p/2$, then this is the maximum size and all descriptions have equal length.

If the subsets of one size are chosen randomly, independently and with equal probabilities, then we can replace the subset indices for one degree by the encrypted content of a secret message. Block encryption typically uses a fixed number of bits, therefore we only use one degree d^*. As the n_d are not powers of 2, we might either use block encryption for a power of 2 close by, or might interpret the whole encrypted secret message as a number to the base n_{d^*}.

Please note that using representation ENUMALL, i.e., enumerating all subsets in the range 0 to $\sum_{d \in D} \binom{p}{d}$, will not allow to use the whole range but still only one degree. While the numbers for a particular degree are random with uniform distribution, this is not true over the whole range because in general $\rho(d) \neq n_d/n$.

If the original message can be reconstructed from any $p(1 + \epsilon)$ received packets, then on average a fraction $\rho(d^*)$ of those will have degree d^* and thus carry a part of the secret message. Hence, the length of the secret message must be chosen accordingly, so that it needs fewer packets, and the secret message must be able to deal with packet loss as well. The latter can be achieved by encoding the secret message with an LT code as well. However, as the encoded secret message must look random, the LT code used must use a representation that looks random as well, see below.

In addition, to choose d^*, we can consider the average number of usable bits per packet. For each packet with degree d^*, we can use $\log_2 n_{d^*}$ bits, and such packets appear with probability $\rho(d^*)$. One can try to maximize the product $\rho(d^*) \cdot \log_2 n_{d^*}$; in many cases, d^* might be $d_{|D|}$, i.e., the highest degree, but the probability $\rho(d)$ will be very low.

If we start with $p = 64$ and representation ENUM, we can choose $d^* = 4$, so that we can use a secret message of 16 packets, because $\rho(4) > 0.25$, and have 19 bits per packet, including the space for the LT encoding.

Table 2: Heuristic factors f_d for $p = 16$ achieving $n^* = 65,522$, i.e., an encoding of 16 bit.

d	$\rho(d)$	n_d	f_d	n_d^*	n_d^*/n^*
1	0.221	16	807	12,912	0.197
2	0.457	120	225	27,000	0.412
4	0.188	1,820	7	12,740	0.194
8	0.134	12,870	1	12,870	0.196

For such an encoding we use a variant of ENUMALL with integral factors f_d so that each subset of degree d has f_d representations, resulting in $n_d^* = f_d \cdot n_d$ and $n^* = \sum_{d \in D} n_d^*$, such that $n_d^*/n^* \approx \rho(d)$. Such an encoding might use slightly more bits, i.e., $\log_2 n^*$, than the most compact encoding, but only if the factor for $d_{|D|}$, that makes up the largest part of the range from 0 to n^*, is much larger than 1. On the other hand, all representations are equally likely if the f_d different possible representations for each subset of degree d are chosen with equal probabilities $1/f_d$.

Table 2 shows factors that we found heuristically for $p = 16$ to achieve an encoding in 16 bits, i.e., not more than ENUM would use, cf. Figure 1. For an encoding with 18 bits, the values of $\rho(d)$ would be met even more closely. As we have 19 bits per packet of the secret message, 3 bits remain for the payload, so that the secret message can have 48 bits in total.

In this second LT code, we might well implement a second covert channel with $p = 8$ and protected by a third LT code with representation ENUMALL that needs 12 bits, i.e., 4 bits are left as payload in each packet, so that the total message size for this second covert channel is 32 bit.

Hence, our approach presents an example of multilevel steganography [4], and even allows more than two levels. The maximum nesting is only restricted by the size reduction: the LT encoding on each level is the packet size on the next level, and thus the packet sizes are reduced from level to level. However, the number of packets are only reduced at the first level, as in the following levels, an encoding can be used where packets of all degrees can be used to carry a part of the secret message from the next level.

4 PRELIMINARY EXPERIMENTS

We have implemented an LT-code encoder and decoder with degree distributions according to Table 1. We have transmitted $p = 16$ source packets via this LT-code. We repeated this experiment 100 times. On average, 39.94 encoded packets must be transmitted to recover the original 16 packets completely. The standard deviation was 13.69.

Then, we placed a covert channel into this LT-code by replacing the random indices of the subsets of degree 4 by indices generated from an encrypted message. As encrypted message, we use the encrypted version of the first validation vector of Advanced Encryption Standard (AES) in Counter Mode, supplied by National Institute of Standards and Technology (NIST) [2]. As there are $\binom{16}{4} = 1,820$ possible subsets of that size, we represented the encrypted message as a number to that base, and generated the indices accordingly. To recover the overt message of $p = 16$ packets completely, on average (100 repetitions) 39.72 encoded packets were transmitted, which is even slightly less than without covert channel. Thus, the performance of the overt channel is not reduced. Yet, the standard deviation was slightly higher with 17.89.

The work involved with encoding the covert channel is low. If the covert message is assumed to be present in encrypted form, then generating a pseudo-random index from the covert message only involves an integer division with remainder, but saves one call to the normal pseudo-random number generator.

The work involved with decoding the covert channel message from the overt message is higher, however, it occurs either on the sink node which has higher performance than the sensor nodes, or it occurs on a separate machine that eavesdrops on the link to the sink node. Hence, we have neglected that effort.

5 DETECTABILITY AND COUNTERMEASURES

Already Luby [7] mentions the possibility to use sequence numbers instead of explicit descriptions of subsets D_j. However, this requires that covert sender and receiver agree on a seed prior to a transmission to initialize synchronized pseudo-random number generators on both sides, that the covert receiver can use to reconstruct the random choices of the covert sender. If random values are transmitted in a fountain code implementation, then detection of the proposed covert channel is difficult because it replaces those random values by other random values, if we consider encrypted parts of the secret message as such. A small advantage for detection could be that the size of the range of the random values is a binomial, i.e., not a power of 2, while encryption algorithms often produce cipher text blocks which are of fixed bit length, i.e., with a range that is a power of 2. However, RSA is a notable exception because it uses a modulus which is a product of two primes, and if those primes can be freely chosen by the covert channel to have a product as close as possible to the original range, then it will be very difficult to detect the difference, especially if the missing values are not placed at one or other end of the range, but somewhere in the middle or distributed over the range. For example, $\binom{16}{8} = 12,870$, while the product of the primes 101 and 127 is 12,827. As only a small fraction of the possible values will occur during transmission of one overt message, the difference will not show. Also, if the sequence of encrypted blocks is seen as a large number which is represented to the base of the binomial, then there is only a difference in either the first or the last block.

Another possible approach for countermeasure would be to apply a form of de-randomization [14] to the original fountain code, so that the transmission of a random value can be replaced by transmission of a deterministic value. The use of sequence numbers mentioned above could be considered as a straightforward form of such de-randomization. As de-randomization of algorithms has progressed much in the last 20 years, more advanced approaches might be possible, but they are outside the scope of the current research.

Finally, a possible means to restrict the bandwidth of such type of covert channel might be to use a sub-optimal degree distribution, e.g., a distribution where only degrees 1 and 2 are used. A low degree d means that the possible number of subsets $\binom{p}{d}$ is small, too, so that its binary representation is shorter and fewer covert data

can be transported. However, as also the probability for the largest degree will change, one has to check if the average number of usable bits per packet, i.e., the product of largest degree's probability and binary representation size of the corresponding subset, will get smaller in the end.

6 CONCLUSIONS

We have presented a network storage covert channel that uses an LT-code as a carrier. This allows to exfiltrate data, e.g., via wireless sensor networks over lossy connections without the necessity of acknowledgment packets. The covert channel uses the random value pattern, and to our knowledge is the first covert channel within a fountain code. As fountain codes have already been proposed to protect secret message transfer against packet loss in the overt channel, we adapt this proposal which allows multilevel steganography, i.e., a second covert channel, and even further levels whose number is only restricted by the shrinking bandwidth in each further level. We have performed preliminary experiments to evaluate the influence of a covert channel on carrier performance.

Future work will comprise to increase the bandwidth of the covert channel, and further experiments that include multilevel steganography to check if this still does not influence performance of the overt fountain code, and investigation of further countermeasures, e.g., based on advanced forms of de-randomization.

Acknowledgment

This work is supported by project SIMARGL (Secure intelligent methods for advanced recognition of malware, stegomalware & information hiding methods, https://simargl.eu), which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833042.

REFERENCES

[1] Rennie Archibald and Dipak Ghosal. 2012. A Covert Timing Channel Based on Fountain Codes. In *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25-27, 2012*, Geyong Min, Yulei Wu, Lei (Chris) Liu, Xiaolong Jin, Stephen A. Jarvis, and Ahmed Yassin Al-Dubai (Eds.). IEEE Computer Society, Los Alamitos, CA, 970–977. https://doi.org/10.1109/TrustCom.2012.21

[2] Lawrence E. Bassham III. 2002. *The Advanced Encryption Standard Algorithm Validation Suite (AESAVS).* Technical Report. National Institute of Standards and Technology, Gaithersburg, MD.

[3] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. 1998. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In *Proceedings of the ACM SIGCOMM 1998 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 31 - September 4, 1998, Vancouver, B.C., Canada*, Gerald Neufeld, Gary S. Delp, Jonathan Smith, and Martha Steenstrup (Eds.). ACM, New York, NY, 56–67. https://doi.org/10.1145/285237.285258

[4] Wojciech Fraczek, Wojciech Mazurczyk, and Krzysztof Szczypiorski. 2012. Multilevel Steganography: Improving Hidden Communication in Networks. *J. Univers. Comput. Sci.* 18, 14 (2012), 1967–1986. https://doi.org/10.3217/jucs-018-14-1967

[5] Jessica Fridrich, Miroslav Goljan, and David Soukal. 2005. Perturbed quantization steganography. *Multimedia Systems* 11, 2 (01 Dec 2005), 98–107. https://doi.org/10.1007/s00530-005-0194-3

[6] Weiwei Liu, Guangjie Liu, Jiangtao Zhai, Yuewei Dai, and Dipak Ghosal. 2016. Designing Analog Fountain Timing Channels: Undetectability, Robustness, and Model-Adaptation. *IEEE Trans. Inf. Forensics Secur.* 11, 4 (2016), 677–690. https://doi.org/10.1109/TIFS.2015.2505688

[7] Michael Luby. 2002. LT codes. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE, New York, NY, 271–280. https://doi.org/10.1109/SFCS.2002.1181950

[8] Norka B. Lucena, Grzegorz Lewandowski, and Steve J. Chapin. 2005. Covert Channels in IPv6. In *Privacy Enhancing Technologies, 5th International Workshop, PET 2005, Cavtat, Croatia, May 30-June 1, 2005, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 3856)*, George Danezis and David M. Martin Jr. (Eds.). Springer, Berlin, 147–166. https://doi.org/10.1007/11767831_10

[9] Wojciech Mazurczyk, Steffen Wendzel, Mehdi Chourib, and Jörg Keller. 2019. Countering adaptive network covert communication with dynamic wardens. *Future Gener. Comput. Syst.* 94 (2019), 712–725. https://doi.org/10.1016/j.future.2018.12.047

[10] Wojciech Mazurczyk, Steffen Wendzel, Sebastian Zander, Amir Houmansadr, and Krzysztof Szczypiorski. 2016. *Information Hiding in Communication Networks.* Wiley-IEEE, Hoboken, NJ.

[11] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. 1996. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, FL.

[12] Michele Rossi, Giovanni Zanca, Luca Stabellini, Riccardo Crepaldi, Albert F. Harris III, and Michele Zorzi. 2008. SYNAPSE: A Network Reprogramming Protocol for Wireless Sensor Networks Using Fountain Codes. In *Proceedings of the Fifth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2008, June 16-20, 2008, Crowne Plaza, San Francisco International Airport, California, USA.* IEEE, New York, NY, 188–196. https://doi.org/10.1109/SAHCN.2008.32

[13] Amin Shokrollahi. 2006. Raptor codes. *IEEE Trans. Inf. Theory* 52, 6 (2006), 2551–2567. https://doi.org/10.1109/TIT.2006.874390

[14] Shu Tezuka. 1995. Derandomization. In *Uniform Random Numbers.* Springer International Series in Engineering and Computer Science (Discrete Event Dynamic Systems), Vol. 315. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-2317-8_6

[15] Osman Ugus. 2013. *Secure and Reliable Remote Programming in Wireless Sensor Networks.* Ph.D. Dissertation. FernUniversität in Hagen. http://deposit.fernuni-hagen.de/2915/

[16] Steffen Wendzel, Luca Caviglione, Wojciech Mazurczyk, Aleksandra Mileva, Jana Dittmann, Christian Krätzer, Kevin Lamshöft, Claus Vielhauer, Laura Hartmann, Jörg Keller, and Tom Neubert. 2021. A Revised Taxonomy of Steganography Embedding Patterns. In *ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021*, Delphine Reinhardt and Tilo Müller (Eds.). ACM, New York, NY, 67:1–67:12. https://doi.org/10.1145/3465481.3470069

[17] Steffen Wendzel, Sebastian Zander, Bernhard Fechner, and Christian Herdin. 2015. Pattern-Based Survey and Categorization of Network Covert Channel Techniques. *ACM Comput. Surv.* 47, 3 (2015), 50:1–50:26. https://doi.org/10.1145/2684195