Improved Concept and Implementation of a Fountain Code Covert Channel

Jörg Keller* and Ewelina Marciniszyn

FernUniversität in Hagen, 58084 Hagen, Germany
joerg.keller@fernuni-hagen.de, ewelina.marciniszyn@gmail.com

Received: May 4, 2022; Accepted: August 22, 2022; Published: September 30, 2022

Abstract

Fountain codes are used to provide reliable communication over a lossy network with low overhead and without acknowledgment. We present a method for network steganography within a fountain code as carrier, which uses most carrier packets to transmit parts of the secret message, and tries to maximize bandwidth. 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 provide a detailed analysis on the possible bandwidth in multiple levels and a complete implementation for Luby Transform (LT) codes which we evaluate with a focus on detectability, i.e., if random values in a carrier packet can be distinguished from secret message parts. We also discuss countermeasures that limit the possibilities for covert channels in fountain codes.

Keywords: network steganography, storage covert channel, fountain code, multilevel steganography

1 Introduction

Fountain codes [1] 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 n source packets must be sent, then receiving any $(1+\varepsilon)n$ 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 solid state disk (SSD) storage, but we will focus on their use in networks. Fountain codes are used in practice and standardized by 3rd Generation Partnership Project (3GPP) [2] and Internet Engineering Task Force (IETF) [3].

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 [4]. In [5], of which this paper is an extended version, we presented a network covert channel that hides within a Luby Transform (LT) code [6] as a concrete and easy to explain implementation of a fountain code. The covert channel uses the random state/value modulation pattern [7] to inject covert data. As the network covert channel will be affected by packet loss in the carrier network, it uses a second fountain code to

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 13(3):25-36, Sept. 2022
DOI:*10.22667/JOWUA.2022.9.30.025*

*Corresponding author: Faculty of Mathematics and Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany, Tel: +49-2331-987-376 ensure complete reconstruction at the receiver side. This presents an application case for multilevel steganography [8], as a second covert channel can be hidden in that second fountain code. As the multilevel scheme is homogeneous, i.e., the second covert channel is of the same type as the first, further levels are possible. There are only practical restrictions on the number of levels, because bandwidth in each level will be reduced, up to some point where bandwidth will be too small to implement a further covert channel with fountain code.

In the current research, we extend the basic idea in several directions by making the following contributions:

- While our initial approach only used encoded packets of one particular degree to transport secret message parts, we now use encoded packets of all degrees except degree 1 and 2. This increases steganographic bandwidth by allowing more source packets in the second fountain code.
- We present the concept of the "half-degree" channel, where the second fountain code uses degree d when the first fountain code uses degree $2d$. This reduces overhead, as the degree of the second fountain code need not be stored explicitly, and the encoding size of the subset in the second fountain code automatically fits well to the encoding size of subset in the first fountain code, thus leaving more room for secret message data, i.e., improving steganographic bandwidth via larger source packet size in the second fountain code.
- We calculate the maximum number of source packets of the second fountain code that can be sent within the first fountain code and still be reconstructed with high probability. Thus, we improve the steganographic robustness of the covert channel.
- We exactly calculate the bandwidth of second level and third level covert channels to demonstrate that multiple levels are indeed possible.
- We present a complete implementation, which we evaluate experimentally with a focus on detectability.

The remainder of this paper 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 briefly summarize the idea underlying our previous approach and present the half-degree covert channel that optimizes steganographic bandwidth and also improves chances for multilevel steganography. Section 4 reports on the implementation and experiments, while Section 5 discusses possible countermeasures and Section 6 concludes with an outlook on future work.

2 Background

2.1 Fountain codes

Fountain codes are rateless erasure codes [1]. To transmit (or store) a fixed set of n 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 n* (or slightly more: $(1+\varepsilon)n$ for an LT code) encoded packets that arrive at the receiver, 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 [6] presented his transform codes (LT codes) 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 [9]. These Raptor codes (see also technical specification [3])

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

	$n = 16$		$n = 64$	
d	$\rho(d)$	$\binom{n}{d}$	$\rho(d)$	$\binom{n}{d}$
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	$\approx 1.83 \cdot 10^{18}$

first apply an erasure code before sending with an LT code, and thus our approach should work also with raptor codes.

Rossi et al. [10] and Bohli et al. [11] 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 n source packets p_i, where $i \in P$, with $|P| = n$. For simplicity, we will assume $P = \{0, 1, \ldots, n-1\}$ 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 $\binom{n}{d_j}$ subsets of size d_j are equally likely). The encoded packet contains data $\bigoplus_{i \in D_j} p_i$, the bitwise exclusive or of the source packet data, plus a representation of subset D_j. We will not go into the details of how to encode D_j, we are only interested in the number of possible subsets. Possible encodings are discussed in [5]. The encoded packet thus is a tuple $e_j = (d_j, D_j, \bigoplus_{i \in D_j} p_i)$.

The decoder knows the number of source packets to be transmitted. Upon receiving an encoded packet e_j with subset D_j of size d_j, the receiver performs the following steps [12]:

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 $(1+\varepsilon)p$ packets with very high probability, for suitable degree distribution Ω. For details about decoder implementations, see e.g. [12].

Table 1 shows degree distributions from [10] and number of possible subsets for $n = 16$ and 64.

Figure 1 shows an example of how $n = 4$ packets are encoded with degrees 1, 2 and 4, the respective subsets are shown. Most transmitted encoded packets are received, and the receiver stores those and computes further encoded packets if it can. It stops once it has received all packets.

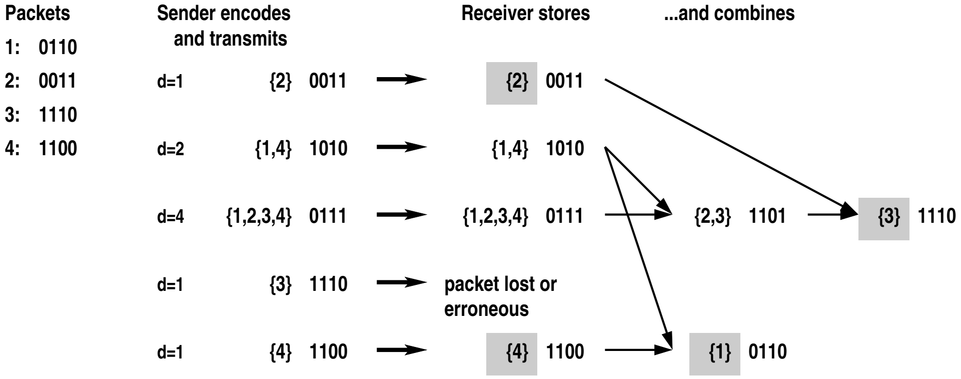

Figure 1: Example fountain code transmission for $n = 4$ packets. When the receiver can combine two encoded packets into a new one, it does a bitwise exor of the payloads. It notices each time a source packet has been received, and stops once all packets, i.e., the complete message, can be decoded.

2.2 Network Covert channels

Steganography is the art of concealing the existence of a secret message in an innocent carrier [4]. 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 [13] 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, its negative influence on the carrier (which also might increase detectability), and its robustness against distortions in the carrier.

There are many proposals for covert channels at all layers of the network stack. To categorize the existing body of work, patterns are used [7, 14]. 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* [14], a generalization of the *random (value) modulation* pattern from [7]:

“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 [8], where a first covert channel serves as the carrier for a second covert channel, which in turn might serve as the carrier for a third covert channel, etc. As bandwidth in deeper levels quickly deteriorates for identical carriers, the additional levels will not have notable influence on bandwidth, and thus not on detectability. Yet, the additional space can be used to transmit an extra key that might go unnoticed even if the first level covert channel is detected.

2.3 Related Work

An example of replacing the pseudo-random content of a network header field with encrypted covert content is given by [15]. They modify the hop count value in IPv6.

To the best of our knowledge, a storage covert channel hidden in data transfer that is protected by a fountain code has not yet been reported in the literature, except for the conference paper [5] that is the basis of the present research. Fountain codes have been mentioned as a means against packet loss within both network storage covert channels [16] and network timing covert channels [17, 18], avoiding usual protocol schemes with acknowledgement packets and re-send of packets. Also, fountain codes have been used in digital media steganography [19].

3 Covert Channel Design

3.1 Notation and Basic Idea

In the following, we will use fountain codes at several *layers*, which we will number consecutively as layer j. $j=0$ is the carrier or overt channel, $j=1$ is the first covert channel in the carrier and at the same time the carrier for the next covert channel, $j=2$ is the second covert channel within the first covert channel, and at the same time the carrier for the third covert channel, etc.

In layer j, we denote by
n_j the number of source packets to be sent in this layer,
l_j the length in bit of each source packet in this layer, i.e., the payload lengthin each encoded packet in this layer,
h_j the length in bit of the header information in each encoded packet, where some encoded packets may not use all of those bits as the length of the header information may depend on the degree of the encoded packet,
$t_j = h_j + l_j$ the total length in bit of each encoded packet sent in this layer,
ρ_j the degree distribution in this layer, and
D_j the set of possible degrees of a packet in this layer, where we assume that the maximum degree $\max D_j \leq n_j/2$.

The header information of an encoded packet comprises the degree d of an encoded packet and the representation of a subset of d indices of source packets that are combined into this encoded packet.

Please note that the number of possible subsets of higher degrees is a binomial and not a power of 2. Therefore, instead of saying that an encoded packet contains $h_j = \lceil \log_2 \binom{n_j}{d} \rceil$ header bits and l_j payload bits, of which only bit combinations are allowed that lead to a valid subset index, we also say that the packet encodes $\binom{n_j}{d} \cdot 2^{l_j}$ possibilities.

In a conventional LT code, the subset above is randomly chosen with uniform distribution among all $\binom{n_j}{d}$ possible subsets of size d. Thus, the basic idea for a covert channel in [5] was to replace the randomly chosen subset by a subset that represents an encrypted secret message, which thus also looks randomly. As block ciphers have been used as basic blocks for pseudo-random number generators, the quality of the (assumed) random numbers should be at least as good as before. An LT code has no cryptographic requirements such as forward secrecy on the pseudo-random numbers, thus these need also not be checked with the covert channel data that act as pseudo-random numbers. To handle packet loss in the carrier, which affects the secret message, also the secret message gets encoded with a fountain code.

The bit capacity $\log_2 \binom{n_j}{d}$ for the secret message depends on the degree d. A smaller degree $d' < d$ needs fewer bits to represent one of the $\binom{n_j}{d'}$ subsets. While the number of bits in the encoded packet available to encode a subset is sufficient to represent all subsets for the maximum degree, and is the same for all degrees, using bits that should be zero at a smaller degree may lead to detectability and could be normalized on the path to disturb a covert channel.

As the size of an encoded packet from the covert channel must be fixed, a design decision in [5] was to use only one degree d^*. We will overcome this limitation in the next subsection with an improved proposal.

3.2 Improved Proposal

In order to increase steganographic bandwidth of our covert channel, encoded packets with as many degrees as possible should be used to transfer steganographic information. In order to avoid the situation where a covert channel packet with high degree must be stuffed into the few possibilities available for a carrier packet of low degree, the concept of *half-degree* covert channel is applied: In a carrier packet of degree $d \geq 2$, we transmit a covert channel packet of degree $d/2$. Thus, the degree must only be stored once, in the carrier, and a low number of subset possibilities in the carrier corresponds to an even lower number of subset possibilities in the covert channel, thus leaving room for steganographic payload. This will change the degree distribution in the covert channel, but our experiments in Section 4 indicate that reconstruction of the covert message is still possible.

Let assume that we have fixed the number n_1 of source packets in the covert channel (see below). Then for each degree $d \geq 2$, $\binom{n_0}{d}/\binom{n_1}{d/2}$ bounds the number of possibilities to encode steganographic payload, i.e. the bit length of the source packet size in the covert channel is

$$l_1 = \min_{d \geq 2} \left\lfloor \log_2 \left(\frac{\binom{n_0}{d}}{\binom{n_1}{d/2}} \right) \right\rfloor . \tag{1}$$

The inner term grows with d, and thus the smallest degree determines l_1. As the number of subset possibilities for degree 2 is still very small and would severely restrict l_1, we exclude carrier packets of this degree from transporting covert information (which would have degree 1). In order to still have packets of degree 1 in the covert channel, which are necessary to ensure reconstructability of the covert message, we use the ample subset space in packets of degree $d \geq 8$ to encode a covert channel packet of degree 1 in addition to the covert channel packet of degree $d/2$. Please note that normally, $\binom{n_0}{d}$ is slightly larger than $\binom{n_1}{d/2} \cdot 2^{l_1}$. To avoid that some subsets in the carrier could not be chosen, which might have a negative impact on carrier performance and thus increase detectability, we use the concept of mappings (see Section 3.3).

To determine the number of source packets in the covert channel, we might simply argue that a fraction of

$$p = \sum_{d \geq 4} \rho_0(d) \tag{2}$$

of the carrier packets contain steganographic information, and thus $n_1 = p \cdot n_0$. If $(1+\varepsilon) \cdot n_0$ received carrier packets suffice to reconstruct the carrier message, then $p \cdot (1+\varepsilon) \cdot n_0 = (1+\varepsilon) \cdot n_1$ packets of these should contain covert channel packets, which should suffice to reconstruct the covert channel message.

However, we are dealing with small numbers here, e.g., the number of received carrier packets will be less than 100. Thus, we may consider the number of received covert channel packets as a random variable X that follows a binomial distribution with $(1+\varepsilon) \cdot n_0$ trials and success probability p [20], and we choose n_1 such that

$$P(X < (1+\varepsilon)n_1) < T \tag{3}$$

where T is a small threshold probability that we can tolerate for non-reconstruction of the covert channel message. Please note that also this is just a guess, as the number of necessary packets to reconstruct the complete message also follows a distribution, and design choices must be validated with experiments.

Construction of a level-2 covert channel within the level-1 covert channel uses the same equations, only with input parameters n_1 and l_1 instead of n_0 and l_0.

In the rest of this section, we will derive accurate parameter values for multiple levels of covert channels. More design space considerations can be found in the thesis [21], cf. Acknowledgments how to access it.

Level 1: According to the right half of Tab. 1 and Eq. (2), $p = 0.431$ of carrier packets have a degree 4 or higher. From the experiments in Section 4, we can derive $\varepsilon \approx 0.5$ to correctly decode carrier messages in more than 99% of the experiments. Not surprisingly, for $n_1 = 32$ source packets in the first covert channel, the chance to decode the covert message correctly according to Eq. (3) is only 10.4%. For $n_1 = 28$, which corresponds to $p \cdot n_0$, the chance increases to 48.8%, and for $n_1 = 24$ to 88.9%. For $n_1 = 16$ the probability grows to $1 - 10^{-9}$.

For $n_1 = 16$, we get $l_1 = \lfloor \log_2 5294.8 \rfloor = 12$ from Eq. (1) when using $d \geq 4$. The minimum is indeed obtained by $d = 4$. For $d = 8$, there is much more room, and 21 bit would be available. However, this is not sufficient to add an encoded covert channel packet of degree 1, which will need 4 bits for the subset and l_1 bits for the payload. From this additional constraint $2l_1 + 4 \leq 21$ we get $l_1 = 8$. For carrier packets with $d = 16$ and $d = 32$, even more covert channel packets of degrees 1 and 2 can be added, to approximate the relative frequencies of the degrees in level 1 to those of Table 1 left part. The total steganographic bandwidth is $n_1 \cdot l_1 = 128$ bit. An alternative calculation with $n_1 = 32$ leads to $l_1 = 5$ bit, i.e., to a steganographic bandwidth of 160 bit at the cost of a reduced reconstructability of the covert message.

Level 2: As carrier packets of degrees 8 and higher make out 0.183 of the fraction 0.439 of carrier packets that carry level 1 covert information, we have $p = 0.417$. Hence, on the second level. If we choose $n_2 = 4$, then the covert message can be decoded in 93.0% of the cases. The payload length is $l_2 = 5$ bit, resulting in a steganographic bandwidth of 20 bit.

Level 3: Here, we can embed $n_3 = 2$ source packets of $l_3 = 2$ bits of a level 3 covert channel with 3 possibilities as header info into the degree-2 header of a level 2 covert channel packet, achieving a steganographic bandwith of 4 bit.

3.3 Mapping of Ranges

Suppose a set $A = \{0, \dots, a-1\}$ shall be mapped as uniformly as possible onto a set $B = \{0, \dots, b-1\}$ where $a < b < 2a$. A mapping, that is as uniformly as possible, is defined via a surjective function $invmap : B \to A$ with the following property: $b - a$ elements of A have exactly two preimages in B, the remaining $a - (b - a) = 2a - b$ elements of A have exactly one preimage in B.

The desired mapping can then be computed as follows: if $x \in A$ has one preimage under *invmap*, then $map(x) = invmap^{-1}(x)$. If $x \in A$ has two preimages under *invmap*, then for each evaluation of $map(x)$, one of the two elements of $invmap^{-1}(x)$ is chosen with uniform probability. As *invmap* is a function, the mapping is uniquely decodable.

In our case, b is a binomial, the number of possibilities for encoding a subset in level j, while a is the product of a binomial and a power of 2, the number of possibilities for encoding a subset and a payload in level $j+1$. It is obvious that $b \geq a$, and that the case $b = a$ is trivial. The case $b \geq 2a$ is either covered by increasing the bit length of the payload, i.e., doubling a until $b < 2a$, or by using the concept of integral factors from [5].

The function invmap can be derived by fixing two disjoint subsets B_1 and B_2 of B, each with size $b - a$. If $y \in B \setminus (B_1 \cup B_2)$, then

$$invmap(y) = y - |\{z \in B_1 \cup B_2 \mid z < y\}| \, .$$

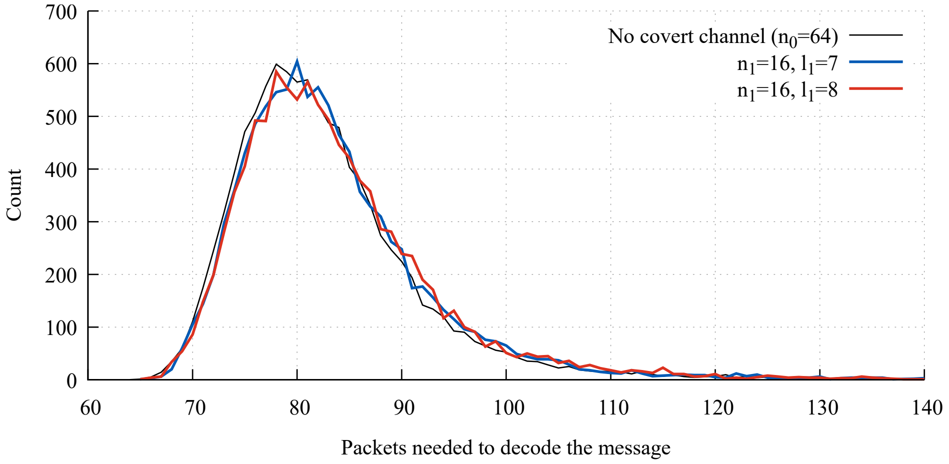

Figure 2: Distribution of required count of received packets to reconstruct message in carrier.

If $y \in B_i$, then

$$invmap(y) = b - a + |\{z \in B_i \mid z < y\}| \, .$$

B_1 and B_2 might be randomly chosen, or might comprise elements with regular distance $b/(b-a)$, starting with elements $b/(3(b-a))$ and $2b/(3(b-a))$, respectively.

4 Experimental Evaluation

We have implemented a simulator for transmissions encoded into LT codes, and extended this simulator to use covert channel data instead of values from the pseudo-random number generator.

We performed two sets of experiments, all with $n_0 = 64$ source packets in the carrier, and a sparse distribution of degrees as powers of 2 as proposed by [10]. In the first set, we checked reconstruction of the carrier message for packet loss rates of 0%, 10%, 20%, both with and without covert channel. We found that the distribution of the number of carrier packets which the overt receiver must receive in order to reconstruct the carrier message remains the same, independent of the loss rate or the presence of a covert channel, see e.g. Fig. 2. A notable exception would be a construction where carrier packets with degree 2 contain covert channel packets with degree 1, which we did not use. Hence, in following experiments we refrained from using packet loss to reduce simulator runtimes. Each experiment was repeated 10,000 times with different seeds of the pseudo-random number generator.

In the second set of experiments, we compared different variants of the half-degree covert channel. With $n_1 = 16$ source packets in the covert channel, each with a length $l_1 = 8$ bit, we distribute covert channel packets as follows using the half-degree system. No covert channel packets are placed in normal headers of degree 1 and 2. The normal header of degree 4 contains a single covert packet of degree 2. The header of normal degree 8 contains a covert packet of degree 4 and also contains a covert packet of degree 1. Normal degree 16 has space for a covert packet of degree 8 and two covert packets of degree 1. The header of normal degree 32 does not contain a covert packet of degree 16, because in this experiment $n_1 = 16$. Instead we fill the available space with five covert packets of degree 1. We make certain that these five covert packets of degree 1 are created from different secret source blocks.

Using this setup, we achieved reconstruction of the secret message by the covert receiver in 99.61% of the experiments, cf. Fig. 2. The total length of the secret message is $n_1 \cdot l_1 = 128$ bit. For $n_1 = 32$ source packets in the covert channel, each with a length $l_1 = 5$ bit, the total length of the secret message is $n_1 \cdot l_1 = 160$ bit, i.e., a bit higher, but a the price that the secret message could only be reconstructed by the covert receiver in 82.34% of the experiments,

Experiments on detectability focused on a Kolmogorov-Smirnov test [20] on the subset encodings in the presence of a covert channel. Yet, even for 20 samples the actual distribution of values could not be distinguished from the expected uniform distribution when range mapping and integral factors are used (cf. Section 3.3), independent of the chosen test parameter, and independent of the simulation run. We concluded that also detection approaches like Shannon entropy or Cabuk's compressibility metric [22] would not detect the presence of such covert channel, yet this might be checked as future work.

More experiments that restricted the design space to the proposal in this paper can be found in the thesis [21], cf. Acknowledgments how to access it.

5 Countermeasures

Already Luby [6] mentions the possibility to use sequence numbers instead of explicit descriptions of subsets D_j. As the sequence numbers do not contain randomness, the covert channel is effectively prevented. However, this requires that overt sender and receiver agree on a seed prior to a transmission to initialize synchronized pseudo-random number generators on both sides, that the overt receiver can use to reconstruct the random choices of the overt sender. Thus, such countermeasure would only be possible in specific circumstances.

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 the value transmitted by the covert channel is a concatenation of another binomial plus the payload, which is a bitvector comprised of the exclusive or of several source packets for the covert channel.

As an example, consider an encoded packet in the carrier with degree 4. If 64 source packets are to be sent, then $\binom{64}{4} = 635,367$ different subsets are possible. If the covert channel can transport 32 source packets via the half-degree channel approach, i.e., with degree 2, then $\binom{32}{2} = 496$ subset combinations are possible. A maximum payload of 10 bit length is possible, which leads to $496 \cdot 2^{10} = 507,904$ different encodings. Thus, almost 30% of the possibilities in the carrier will not occur. If 30% of the encodings in the covert channel are mapped in a random fashion to either of two combinations (cf. Subsection 3.3), this will still change the frequency of appearance, and might be detected by a Kolmogorov-Smirnov-Test [20], even for the small number of samples that we have. Thus, we have a tradeoff here between design for maximum bandwidth and design for maximum stealthyness, If e.g. the degree in the covert channel would be chosen to be 3 instead of 2, then $\binom{32}{3} = 4,960$ combinations and a payload of 7 bits are possible, with a total of $634,880$ encodings, which is much closer to $635,367$ than before, yet at the price of losing 3 bits of payload in every packet of the covert channel.

Another possible approach for countermeasure would be to apply a form of de-randomization [23] 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 $\binom{n}{d}$ 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, uses about half of the carrier packets to transport secret message data, tolerates packet loss by using a second LT code, and optimizes steganographic bandwidth by the use of half-degree technique. Such a covert channel allows to exfiltrate data via wireless sensor networks over lossy connections without the necessity of acknowledgment packets, without impairing the functionality or the performance of the carrier. The covert channel uses the random value pattern, and to our knowledge is the first covert channel within a fountain code. The second LT code allows integration of a second level covert channel, i.e., multilevel steganography. We have demonstrated that multiple levels are feasible for realistic parameter settings.

We have presented an implementation and performed experiments to evaluate the detectability of the covert channel, and the influence of a covert channel on carrier performance. The experiments confirm that the covert channel has no influence on the performance of the carrier and is thus barely detectable for a warden that is not aware of this particular type of covert channel. If a warden searches particularly for this type of covert channel, anomalies from the range difference of a random binomial and the covert message encoding might be detectable via stochastic tests like Kolmogorov-Smirnov. Furthermore, the covert message can be reconstructed by the covert receiver in the overwhelming majority of the cases, and a tradeoff is possible to increase steganographic bandwidth at the cost of reduced robustness.

The degree distribution of the normal fountain code has an impact on the transmission success rate and secret message size of the covert fountain code. Future work could investigate how much the distribution of the normal fountain code can be modified to favor higher degrees, without increasing the number of required normal packets too much. If the normal fountain code is too sub-optimal then this might indicate the existence of a covert channel.

In addition, our experiments have focused on $n_0 = 64$. Higher values could be investigated. The proposed half-degree method lends itself well to being embedded into normal fountain codes with different values of n_0, or could be adapted to work with a dense degree distribution instead of a sparse degree distribution.

Finally, our future work will comprise to transfer our approach to more advanced forms of fountain codes such as raptor codes, and integrating our approach in a real-world use case such as WSN protocol LEACH and its descendants [24].

Acknowledgments

This article is an extended version of [5]. New contributions are in particular: Concept how to use more overt packets to transmit covert information; Extending space for covert payload by the concept of "half-degree" channel; Detailed calculation of bandwidths for 3 levels of covert channels; Implementation and experimental results. Both authors contributed equally to the concepts. E. Marciniszyn additionally did all implementation work as part of her thesis [21], which is available at `https://feu.de/pv/docs/abschlussarbeit_marciniszyn_.pdf`. We thank the anonymous reviewers for their very helpful comments.

References

[1] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *Proc. of the 1998 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'98), Vancouver, British Columbia, Canada*, pages 56–67. ACM, October 1998.

[2] 3rd Generation Partnership Project(3GPP). Multimedia broadcast/multicast service (mbms) protocols and codecs (release 17). Technical Report TS 26.346, 3rd Generation Partnership Project, 2022.

[3] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder. (RaptorQ) Forward Error Correction Scheme for Object Delivery. IETF RFC 6330, August 2011. `https://www.rfc-editor.org/info/rfc6330` [Online; Accessed on September 10, 2022].

[4] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, and K. Szczypiorski. *Information Hiding in Communication Networks*. Wiley-IEEE, 2016.

[5] J. Keller. Multilevel network steganography in fountain codes. In *Proc. of the 5th European Interdisciplinary Cybersecurity Conference (EICC'21), Targu Mures, Romania*, pages 72–76. ACM, November 2021.

[6] M. Luby. Lt codes. In *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02), Vancouver, British Columbia, Canada*, pages 271–280. IEEE, November 2002.

[7] S. Wendzel, S. Zander, B. Fechner, and C. Herdin. Pattern-based survey and categorization of network covert channel techniques. *ACM Computing Surveys*, 47(3):1–26, April 2015.

[8] W. Fraczek, W. Mazurczyk, and K. Szczypiorski. Multilevel steganography: Improving hidden communication in networks. *Journal of Universal Computer Science*, 18(14):1967–1986, July 2012.

[9] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6):2551–2567, June 2006.

[10] M. Rossi, G. Zanca, L. Stabellini, R. Crepaldi, A. F. Harris III, and M. Zorzi. Synapse: A network reprogramming protocol for wireless sensor networks using fountain codes. In *Proc. of the 5th IEEE Annual Communications Society Sensor, Mesh and Ad Hoc Communications and Networks SECON Workshops (SECON'08), San Francisco, California, USA*, pages 188–196. IEEE, June 2008.

[11] J.-M. Bohli, A. Hessler, O. Ugus, and D. Westhoff. Security enhanced multi-hop over the air reprogramming with fountain codes. In *Proc. of the 34th Annual IEEE Conference on Local Computer Networks (LCN'09), Zurich, Switzerland*, pages 850–857. IEEE, October 2009.

[12] O. Ugus. *Secure and Reliable Remote Programming in Wireless Sensor Networks*. PhD thesis, FernUniversität in Hagen, Germany, 2013.

[13] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[14] S. Wendzel, L. Caviglione, W. Mazurczyk, A. Mileva, J. Dittmann, C. Krätzer, K. Lamshöft, C. Vielhauer, L. Hartmann, J. Keller, and T. Neubert. A revised taxonomy of steganography embedding patterns. In *Proc. of the 16th International Conference on Availability, Reliability and Security (ARES'21), Vienna, Austria*, pages 1–12. ACM, August 2021.

[15] N. B. Lucena, G. Lewandowski, and S. J. Chapin. Covert channels in ipv6. In *Proc. of the 5th International Workshop on Privacy Enhancing Technologies (PET'05), Cavtat, Croatia*, volume 3856 of *Lecture Notes in Computer Science*, pages 147–166. Springer Berlin Heidelberg, May-June 2005.

[16] W. Mazurczyk, S. Wendzel, M. Chourib, and J. Keller. Countering adaptive network covert communication with dynamic wardens. *Future Generation Computer Systems*, 94(6):712–725, May 2019.

[17] R. Archibald and D. Ghosal. A covert timing channel based on fountain codes. In *Proc. 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12), Liverpool, United Kingdom*, pages 970–977. IEEE Computer Society, June 2012.

[18] W. Liu, G. Liu, J. Zhai, Y. Dai, and D. Ghosal. Designing analog fountain timing channels: Undetectability, robustness, and model-adaptation. *IEEE Transactions on Information Forensics and Security*, 11(4):677–690, April 2016.

[19] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *Multimedia Systems*, 11(2):98–107, December 2005.

[20] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 1991.

[21] E. Marciniszyn. Fountain codes and covert channels. Master's thesis, FernUniversität in Hagen, Germany,

2022.

[22] S. Cabuk, C. E. Brodley, and C. Shields. IP covert channel detection. *ACM Transactions on Information and System Security*, 12(4):1–29, April 2009.

[23] S. Tezuka. Derandomization. In *Uniform Random Numbers*, pages 143–160. Springer New York, NY, 1995.

[24] S. Varshney and R. Kuma. Variants of LEACH routing protocol in WSN: A comparative analysis. In *Proc. of the 8th International Conference on Cloud Computing, Data Science Engineering (Confluence'18), Noida, India*, pages 199–204. IEEE, January 2018.

Author Biography

Jörg Keller received M.Sc. and Ph.D. degrees and habilitation from Saarland University, Saarbrücken, in 1989, 1992, and 1996, respectively. Since 1996, he is a professor of computer engineering at FernUniversität in Hagen, Germany, where he leads the Parallelism & VLSI research group. His research interests include network steganography, cryptographic primitives for embedded systems, energy-efficient and fault-tolerant parallel computation, and blended and virtual laboratories.

Ewelina Marciniszyn received the bachelor degree in computer science from the University of Hagen in 2022. Before pursuing these studies she worked in the field of law. She holds an LL.M. (Master of Laws) in intellectual property law from London / Dresden, and is a fully qualified German lawyer. Her interests are cybersecurity, programming and intellectual property law.