Security Challenges for Light Emitting Systems

Louiza Hamada, Pascal Lorenz, Marc Gilg

future internet

MDPI

Article

Security Challenges for Light Emitting Systems

Louiza Hamada †, Pascal Lorenz *,† and Marc Gilg †

IRIMAS Institute, University of Haute Alsace, 34 Rue du Grillenbreit, 68008 Colmar, France; louiza.hamada@uha.fr (L.H.); marc.gilg@uha.fr (M.G.)

* Correspondence: lorenz@ieee.org

† These authors contributed equally to this work.

Abstract: Although visible light communication (VLC) channels are more secure than radio frequency channels, the broadcast nature of VLC links renders them open to eavesdropping. As a result, VLC networks must provide security in order to safeguard the user's data from eavesdroppers. In the literature, keyless security techniques have been developed to offer security for VLC. Even though these techniques provide strong security against eavesdroppers, they are difficult to deploy. Key generation algorithms are critical for securing wireless connections. Nonetheless, in many situations, the typical key generation methods may be quite complicated and costly. They consume scarce resources, such as bandwidth. In this paper, we propose a novel key extraction procedure that uses error-correcting coding and one time pad (OTP) to improve the security of VLC networks and the validity of data. This system will not have any interference problems with other devices. We also explain error correction while sending a message across a network, and suggest a change to the Berlekamp–Massey (BM) algorithm for error identification and assessment. Because each OOK signal frame is encrypted by a different key, the proposed protocol provides high physical layer security; it allows for key extraction based on the messages sent, so an intruder can never break the encryption system, even if the latter knows the protocol with which we encrypted the message; our protocol also enables for error transmission rate correction and bit mismatch rates with on-the-fly key fetch. The results presented in this paper were performed using MATLAB.

Keywords: Li-Fi; visible light communication; security; Berlekamp–Massey algorithm; Reed–Solomon; error correcting; OTP

Citation: Hamada, L.; Lorenz, P.; Gilg, M. Security Challenges for Light Emitting Systems. *Future Internet* **2021**, *13*, 276. https://doi.org/10.3390/fi13110276

Academic Editor: Paolo Bellavista

Received: 15 September 2021
Accepted: 27 October 2021
Published: 28 October 2021

1. Introduction

The security of networks has become a major challenge in recent years. First, there is a need to keep information confidential, so that only authorized parties have access to it, and protect the information transmitted over the network [1], which can even cover the backup of files or passwords stored on computers that are connected online, as well as access to computer systems and applications. This problem may be solved by requiring users to utilize keys to access their workstations, password secure critical documents, and digitally sign their emails [1]. The second safety challenge involves the protection of the internet infrastructure. The purposes are to protect against attacks on the configuration of network devices, stealing networking resources and, consequently, maliciously jamming nodes or connections with disingenuous data that prevent legitimate messages from passing [1].

One solution is visible light communication, a new emerging paradigm of wireless technology proposed in the beginning of 2000s [2]. This technology was designed as a point-to-point wireless communication link between an LED light and a receiver, equipped with a photo detection (PD) system. The transfer of rate depends on the digital technology used and, thus, on the light [3]. According to [4], the light communication system (LCS) incorporating white LED illumination has received considerable attention in the last decade. In the paper, the authors propose a handover algorithm for an internal cellular system to extend the bandwidth transmission. There are several studies on security, in many aspects,

namely the physical layer, MAC layer, indoor communication, outdoor communication, inter-channel interference, etc. In [5], the paper identifies the physical characteristics of VLC systems with security relevance. It also summarizes all of the security techniques that have been proposed in the literature for VLCs to date, including the physical layer security (PLS), which is discussed from an information theoretic perspective, as well as availability and integrity issues. The paper also addresses issues of secure localization and key generation [5].

Other articles have proposed solutions based on OFDM modulation, and how to exploit this modulation for the benefit of cryptography and noise during communications in the system [6]. In [6], the authors provide an overview of recent developments in Li-Fi physical layer security (PLS) and explains the main differences between Li-Fi PLS and RF PLS. Furthermore, the authors of [7] present a new key extraction procedure for orthogonal frequency division multiplexing (OFDM) schemes in an indoor setting. The methodology presented extracts keys at the media access control (MAC) and physical levels. Because each OFDM signal frame is encrypted with a distinct key, it provides excellent physical security. The authors of [8] also propose a Li-Fi access point (AP) structure that employs orthogonal frequency division multiplexing (OFDM) and a tunable optical coding/decoding technique based on the optical pulse delay in an optical delay line (ODL) loop vector to allow efficient mapping of OFDM-based data access and transfer. In this paper, we present a generalized BM method for protecting and making our network more secure, especially during this global health crisis. In terms of complexity, our improved method has a complexity of $O(L^2)$ for both encoding and decoding. Another advantage of this approach is that even if the intruder discovers the encryption algorithm, the intruder will be unable to decode a message since he/she needs the value of k to encode and decode it. According to our estimations, even if the approach changes, the error-correcting aspect will not alter.

2. Security Challenges in VLC System

2.1. Security of MAC in Li-Fi

The IEEE 802.15.7 specification describes the MAC security mechanisms that higher protocol levels may necessitate [5]. The standard indicates that cryptographic algorithms should provide data transmission confidentiality and integrity, but their implementation should not be overly complicated and, in particular, should not use excessive quantities of computer power, storage space, and power sources [5]. The need stems from VLC applications in personal area networks and body area networks (PANs and BANs), where computing resources are severely limited. Moreover, because VLC networking is also utilized as LAN technology, the standard's current security measures may be insufficient. For MAC-level security, the IEEE 802.15.7 standard employs symmetric-key cryptography. The standard does not include key generation and management procedures. Frame protection is implemented at the MAC level as follows: (1) A "link key" that is only utilized by peers is used for peer-to-peer communication; (2) a shared "group key" is used for communication between a group of devices. This technique allows for some flexibility and an application-specific trade-off between key storage and key maintenance costs vs. the cryptographic security given. The 128-bit AES algorithm is used for encryption at the MAC layer. A keyframe counter is also used to force key initialization and prevent replay attacks [5]. The data, beacon payload, and control payload frames are encrypted.

2.2. Security in Physical Layer

2.2.1. Theoretical Basis at the Physical Layer Security

The most advanced study subject is PLS in visible light communication (VLC) systems, which may incorporate secure communication zones, artificial noise creation, and modulation-based secret key generation at the physical layer [5]. The MAC layer's security is based on edge-to-edge encryption, cryptography authentication, strong password protection, and control mechanisms based on strong cryptography algorithms: symmetric for edge-to-edge encryption and asymmetric for authentication, key generation, and key trans-

fer. Wyner presented an eavesdropping channel model for physical layer security (PLS) in 1975, which was later extended to various channel models, including those characterizing wireless systems [5].

The principle of security in cryptography systems is: as long as computing the offensive power of the hitters is below the limit, the system is secure. For example, it is assumed that an eavesdropper does not have enough computing power to perform a brute force attack on the private key in a short period of time to decode the encrypted message [5].

PLS is a remarkable complement to cryptography-based protection, as it introduces an additional layer of secrecy that is demonstrably unbreakable, regardless of the attackers computing power, as well as an option for standalone privacy solutions for systems that are limited in hardware and/or power, such as in internet of things (IoT) applications [5].

2.2.2. On–Off Keying Modulation and Graduation Methodology

On–off Keying (OOK) is the easiest modulation patterns for VLC systems, where the LEDs turn on and off depending on the value of "one" or "null" in the data bits. When the on–off Keying modulation is disabled, it means that the light intensity can be easily reduced as long as the network can clearly distinguish between "on" and "off". The diagram of the on–off keying transmitter in Figure 1 shows how the upper layer bits enter forward error correction (FEC) before being encoded with Manchester run-length limited (RLL) [9]. In Manchester coding, which involves a clock, a logical zero is represented by an on–off keying symbol "01" and a logical one is represented by an on–off keying symbol "10", resulting in a direct current (DC) balanced code. We modified the scheme of [8] to obtain the new architecture proposed in Figure 2. The scaling of the on–off modulation is achieved either by setting the "on" or "off" levels of the on–off modulation symbol with reduced intensity, or by changing the average duty cycle of the pulse by inserting a "clearing/compensation" time in the modulation pulse. This pulse is obtained by turning the light source on or off completely for the time necessary for scaling [9]. This adds a direct current component to the light source control pulse, which affects the intensity of the light. For example, if the brightness is X percent with the T1 time period and the clearing patterns have an average luminosity of Y percent with the T2 time period, the resultant average luminosity N (percent) is frequently stated as follows.

$$N = \frac{XT_1 + YT_2}{T_1 + T_2} \quad (1)$$

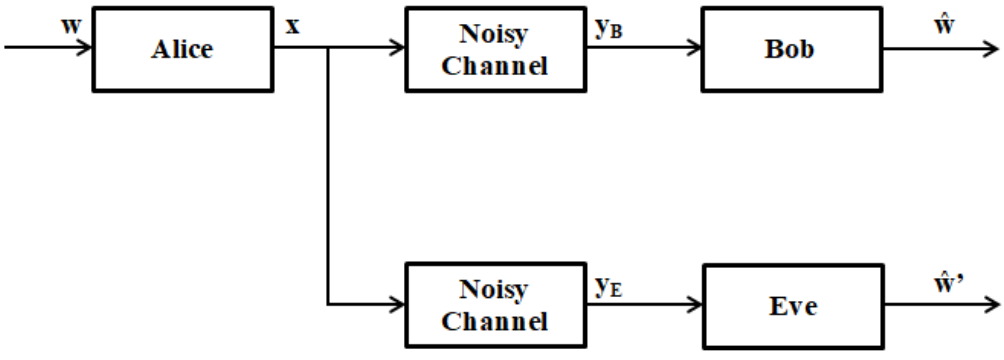


Figure 1. Eavesdropping channel model introduced by Wyner.

If we provide insufficient power to the light emitting diodes (LEDs), the color will change when the on–off keying 'on and off' levels are reset, resulting in an endless binary data rate as the brightness falls. Inserting clearing patterns, on the other hand, results in a lower bit rate when the brightness falls, implying a lower binary data rate at a constant range. However, compression techniques have been employed in similar work to decrease the binary data rate [9].

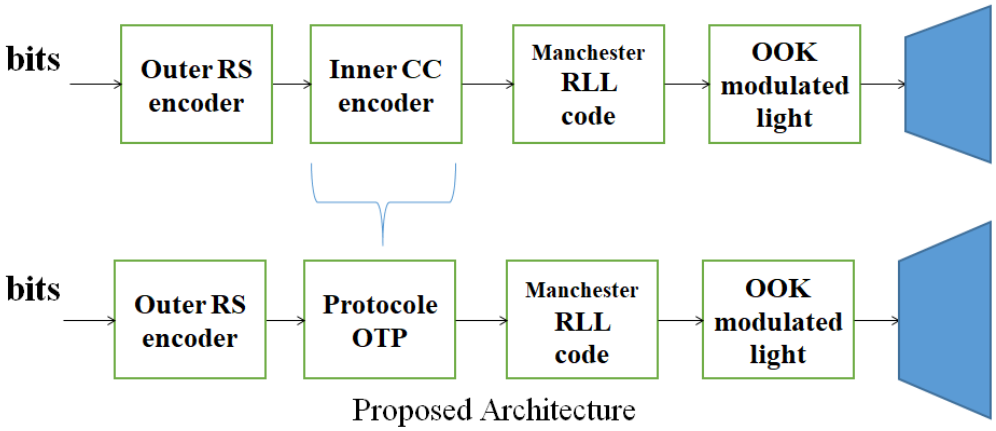


Figure 2. OOK transmitter block diagram.

The architecture of the on–off keying scaling frame illustrated in Figure 3 includes a synchronization prologue (preamble), a PHY header that provides frame details: frame length, modulation technique, coding, and payload framework. If the compensation period is too lengthy, the receiver may lose synchronization because the receiver clock is generally retrieved from the data [9]. Therefore, in the on–off keying scaling frame structure, the information frame is partitioned into subframes, each subframe is usually preceded by a resynchronization field that uses an infinite sequence of the 1010 transition pattern used to reset the data clock during the clearing time. The information frame is fragmented into subframes of the appropriate length during the computation of the banking system and, thus, FEC is applied. We have shown an example of the on–off keying scaling to increase luminosity by adding clearing patterns in Figure 3. The typical luminosity (XY) of N per cent is realized by adjusting the luminosity of the information and, thus, the clearing patterns. When the on–off keying modulation is used for transfer of data, the information inherently includes a duty cycle of 50 per cent due to the Manchester run-length limited secret letter. To control the duty cycle within the frame, timing and brightness clearing patterns (as defined in the equation) must be applied to support the XY of N p. c. Outside the information communication frame, inactive patterns of N per cent average luminosity are sent to ensure that the typical luminosity of the network lighting source remains constant [9].

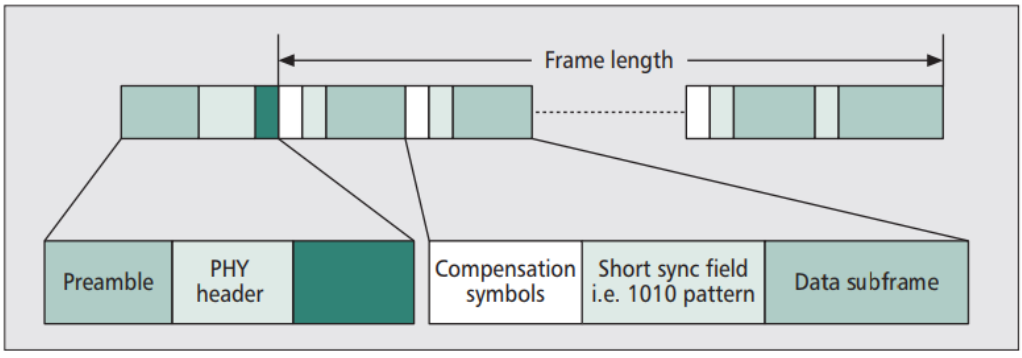


Figure 3. OOK gradation structure.

3. Application and Overview

The Reed–Solomon error correcting codes that employ the Berlekamp–Massey method were referenced by the author of [9]. We are particularly interested in the Berlekamp–Massey approach, and based on the publication of [9], we proposed an improved BM solution that can both encrypt a message and correct transmission issues during the network connection.

Reed–Solomon (RS) error correcting codes operate on a dataset that is represented as a set of finite field elements known as symbols. Multiple symbol problems can be identified and resolved using RS codes. Denoted by RS (N,K), where N is the number of bits in the code word and K is the number of bits in the information word. We specify two variables m that determine the length of the symbols and t that limit the maximum number of symbols, which may be rectified for each pair RS (N,K) [10].

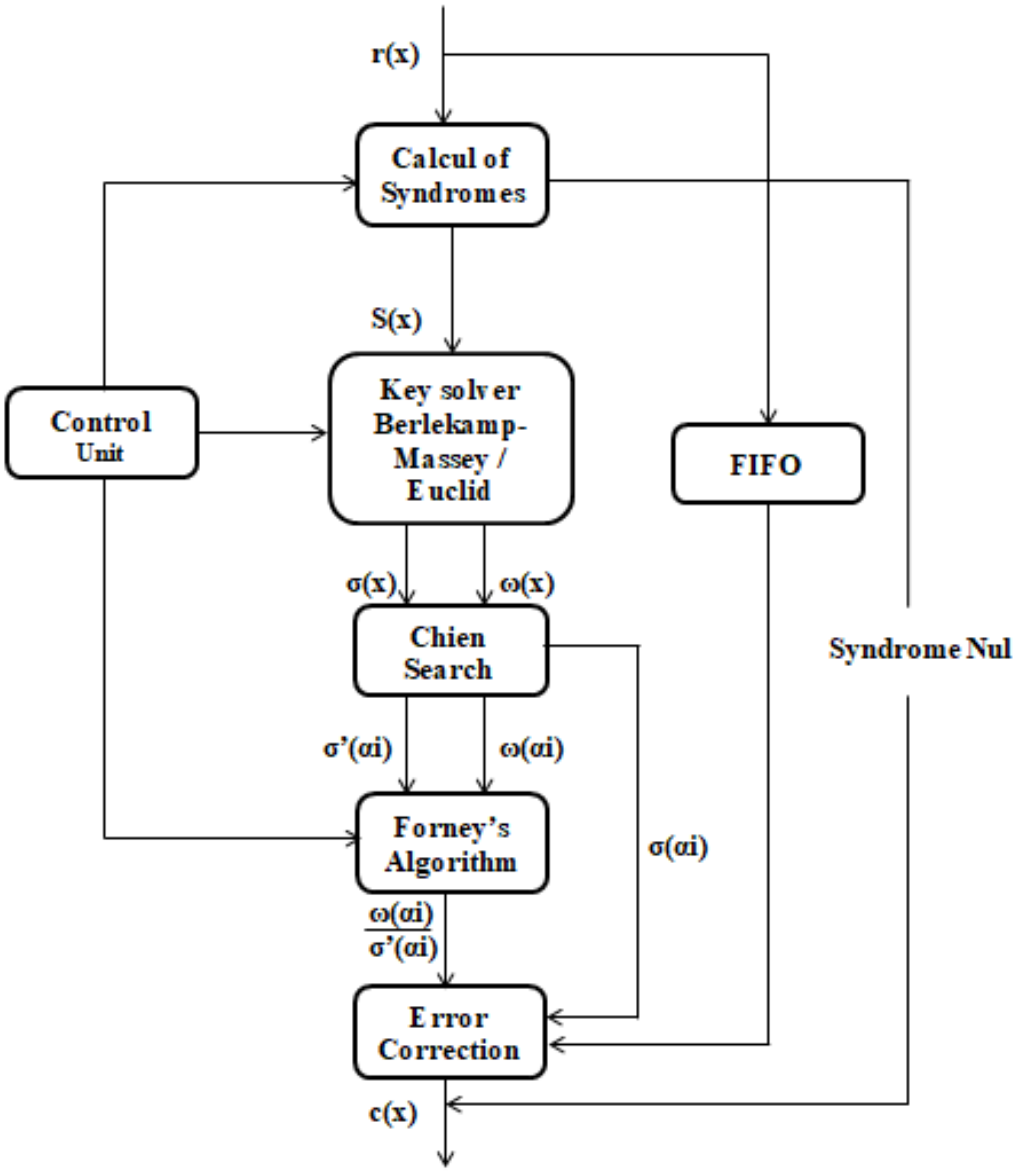


Figure 4. Decoding steps of Reed–Solomon.

The encoder architecture shows that each number is associated with a degree input, which is a constant of the polynomial g(x). For a given unit, the polynomial information i(x) is specified component-by-component in the encoder [11]. These components check at the encoder issue once desired latitude is reached, where the management logic returns them via an associated degree adder to provide the appropriate parity. This process continues until all k symbols (elements) of i(x) have been input to the encoder [11]. At this point, the output control logic only allows the path of the input file, while the equality path remains disabled. With an output latitude of about one clock cycle, the encoder issues the last data frame with the (k + 1)th clock [12]. In addition, the feedback control logic feeds the output of the adder into the bus during the first k clock cycles. A period of at least n − k timing

of the clock after the entry of the last frame in the encoder, i.e., at the kth timing of the clock [12]. Meanwhile, the feed-forward controller disables the adder output so that there is no feed-forward and continuously supplies zero symbols to the bus. Similarly, the output logic command disables the input information path and allows the encoder to output equality symbols (from the k + 2th to the n + 1th timing of the clock). Thus, a new unit starts at the n + 1th timing of the clock. The basic idea of the Reed–Solomon decoder is to detect an erroneous sequence with few terms, which, summed to the received data, results in a valid code word. Several steps are necessary for the decoding of these codes [11]; Figure 4. Because of the small number of symbols that Reed–Solomon coding can correct, this coding is very poor, with an impulsive noise of long duration, or regular random noise. The linear complexity of the encoding of RS codes is O(nlogn) and O(nlog²n), which pushed us to propose another algorithm of error correction while keeping the concept of RS codes in Figure 5 [12].

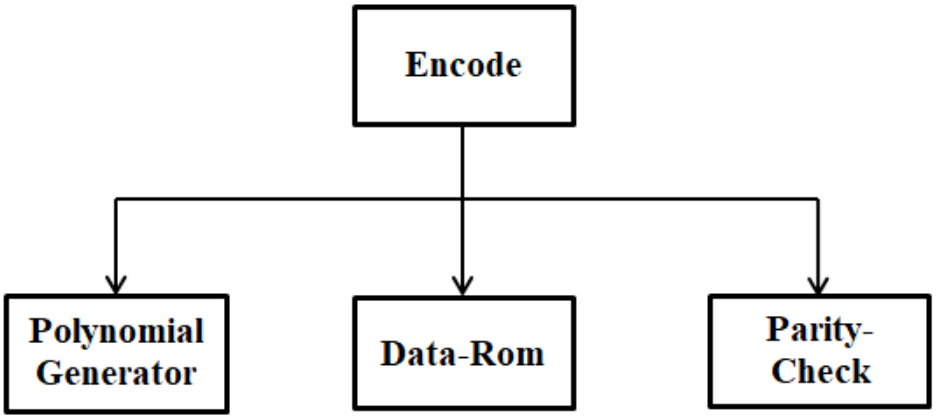


Figure 5. Encoding steps for Reed–Solomon.

The RS code decoding steps are summarized in the diagram of Figure 4; the identification of the variables is clarified in Table 1.

The code word r(x) = c(x) + e(x) [11]. RS decoder can detect the position and length of up to t errors (or 2t deletions) and correct them [10].

Table 1. Decoding Variable.

Variable	Role
r(x)	Received code word
S(x)	Syndromes
$\omega(x)$	Error magnitude polynomial
ω(αi)	Error amplitude polynomial evaluated for all elements included in GF (2^m)
σ(x)	Error locator polynomial
σ(αi)	Error location polynomial evaluated for all elements in GF (2^m)
σ′(αi)	Derivative of the error location polynomial evaluated for all elements in GF (2^m)
c(x)	Recovered code word

Steps to decode RS Code [11]:

(1) Determine the syndrome generator;

(2) Use Berlekamp's algorithm or Euclid's algorithm to form the polynomial error;

(3) Find the roots of this polynomial; usually this is done using the Chein search algorithm;

(4) Determine the error type, Forney's algorithm, or any other matrix inversion algorithm calculates this;

(5) Correct the faulty symbols by overlaying the mask and the data word, and inverting all the bits that are corrupted one-by-one using the XOR operation [11].

4. Results and Discussion

The idea is to review the Berlekamp–Massey, and at the end modify the result and adapt an error correction to the transmitted message. Let us take the message to be sent m = 110,111,000 with a length of 9 bits, we will unroll the modified Berlekamp–Massey; results are presented in Table 2. From Table 2, N is the number of bits in the sent sequence, S_N denotes the number of bits in the sent message, L represents the number of errors, and M denotes the size of the LFSR produced by Berlekamp–Massey. We will consider d, the key created by our method. Each sent message creates a secret key, and f(x) and g(x) are the error locator polynomials, which determine the erroneous symbols that occur during message transmission, and the error estimator polynomial, which corrects the symbols that contain errors, respectively.

To encode our previous message (m), we apply an XOR between m and the key obtained by the BM algorithm that we modified; this key will be on 3 bits, for the digits obtained, different from "0" and "1". The encrypted message is d(x). Now, to decode d(x), we reapply an XOR between it and k in binary. The decoded message d(x), then with errors, to correct these errors, we will go through (02) steps:

First step: we apply an XOR between the initial message (m) and the key this time on 4 bits for the digits different from 0 and 1, we will call the result obtained p(x).

Second step: an XOR will be applied between the d'(x) and p(x).

Table 2. Example of the Berlekamp–Massey algorithm calculation.

N	S_N	d	L	f(x)	m	g(x)
			0	1	−1	1
0	1	1	1	x + 1	0	1
1	1	2	1	x + 1	0	1
2	0	1	2	$x^2 + x + 1$	2	x + 1
3	1	2	2	$x^2 + x + 1$	2	x + 1
4	1	2	2	$x^2 + x + 1$	2	x + 1
5	1	3	2	$x^2 + x + 1$	2	x + 1
6	0	2	2	$x^2 + x + 1$	2	x + 1
7	0	1	6	$x^6 + x^5 + x^2 + x + 1$	7	$x^2 + x + 1$
8	0	1	6	$x^6 + x^5 + x^3 + 2x^2 + 2x + 1$	7	$x^2 + x + 1$

Demonstration:

Encode: m ⊕ key (3 bits): [1 1 0 1 1 1 0 0 0] ⊕ [1 0 1 0 1 0 1 0 0] = [0 1 1 1 0 1 1 0 0]: d(x).

Decode: d(x) ⊕ k (9 for this example): [0 1 1 1 0 1 1 0 0] ⊕ [1 0 0 1 0 0 0 0 0] = [1 1 1 0 0 1 1 0 0]: d'(x).

Correction errors:

First step: m ⊕ key (4 bits) = [1 1 0 1 1 1 0 0 0] ⊕ [1 0 0 1 0 1 0 0 1] = [0 1 0 0 1 0 0 0 1]: p(x).

Second step: d'(x) ⊕ p(x) = [1 1 1 0 0 1 1 0 0] ⊕ [0 1 0 0 1 0 0 0 1] = [1 1 0 1 1 1 0 0 0].

The rule $\sum_{i=1}^{L} c_i s_{N-i} = 0$ ensures that the Berlekamp–Massey algorithm runs correctly. The rule is assured for the results in Table 2.

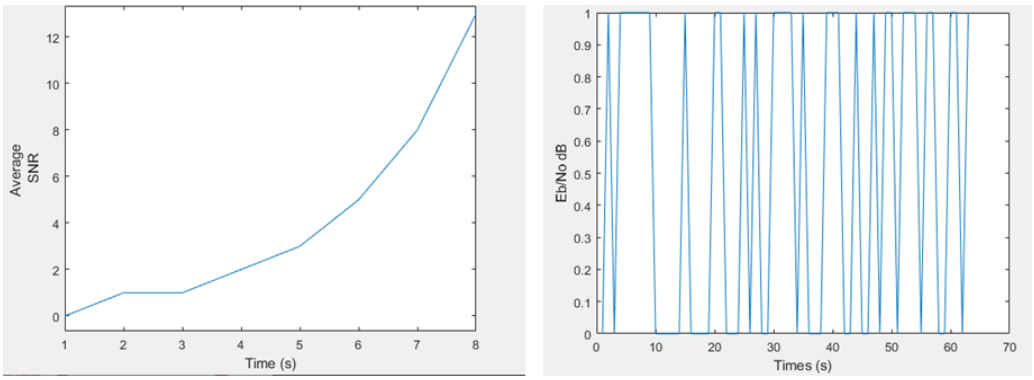
Berlekamp–Massey is a variant to the Reed–Solomon corrective codes, which consist, on the one hand, in constructing for successive values of N, an LFSR of length LN, and feedback polynomial f_N, which generates the first N bits of the sequences [12]. The sequences sent (encoded) must be a multiple of a polynomial S(x), called the generator polynomial, known in advance to the sender and the receiver. We can show that, for N = 2L, the algorithm returns the feedback polynomial of the starting LFSR. On the other hand, to generate a key from the sent message, which will be of the same size of it, each message will have its own key; we cannot have two different messages with the same key. The receiver will receive the coded message with the key for decrypting the message, once authenticated; the user receives a secret code on his smartphone for the second authentication.

As mentioned before, the objective of Berlekamp–Massey is to find the minimal level of errors (L) and f(x) that conducts to all syndromes Sn + $C^1S_{n-1} + \ldots + C^LS_{n-L}$, at every iteration, the algorithm calculates the value d, which we will consider at the end of our algorithm the key for each message. If d = 0, this means that f(x) and L are correct; we increment m and continue. If d0, the algorithm continues to run and recalculates each time f(x) until d = 0. [13].

The algorithm must also decrease the number of errors (L) if necessary [14]. If L is equal to the current error number, the gap during the iteration process becomes zero before n becomes larger than or equal to 2L. Or, the algorithm will update the value of L and g(x), decrease L, and put "m = 1". "L = (n + 1 − L)" indicates the number of valid syndromes to calculate and subsequently correct errors, and also handles the case where L decreases by more than 1 [14].

The change made is to add a variable k that is used to decode the message sent via the light signal. This variable is selected by the user from the odd numbers and must be different from 1.

Figure 6 below shows the simulation results of the Berlekamp–Massey algorithm and its modified variant; the images represent the noise resistance during communication over a Li-Fi network. We notice that the existence of noise is quite regular for the modified BM, unlike for the noise in (Figure 6b). This represents an advantage to adapt the modified BM algorithm.



(a) (b)

Figure 6. Noise measurement in a Li-Fi network. (**a**) according to modified BM, (**b**) according to BM.

Figure 7 represents an example of the RS code flow for error correction: (15,11), (31,27), (63,59), and (127,123). The above graph shows that the error-correcting Reed–Solomon codes become more efficient as the code size grows, since the noise impact decreases with a larger code size. Furthermore, for bigger code words, the noise duration must represent a relatively small proportion, and the received noise must be averaged over a lengthy period of time. For large block sizes, the Reed–Solomon codes are favored. This indicates that the ideal big size code word represents a step forward in terms of performance. On the other side, a very high size code word will complicate transmission implementation.

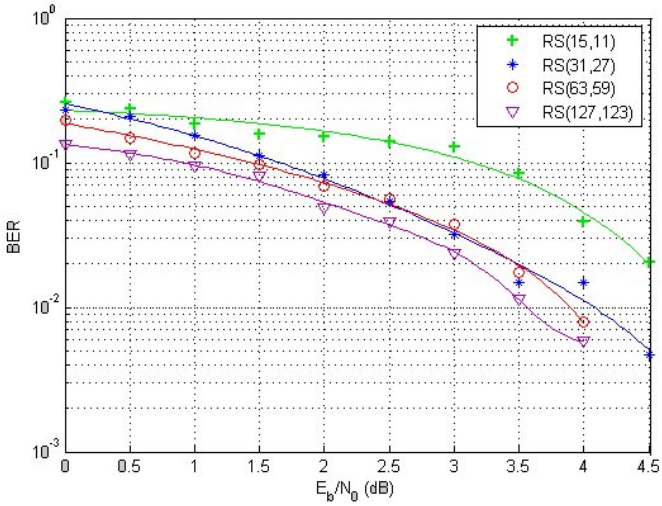


Figure 7. Performance of the Reed–Solomon code for similar error correction.

Choice of Reed–Solomon

A comparison with other corrector codes was made [15]; the results of this comparison are presented in Table 3.

The comparison is based on the speed of encoding/decoding of these codes; which is an important argument as these codes are often used to transmit information in real-time, on the efficiency to the noise resistance, and the ability to correct a larger number of errors.

The security of encoding and decoding is also an essential consideration in the selection of Reed–Solomon codes. Other corrective codes, such as the Hamming code, are considerably more difficult to identify and fix transmission faults than Reed–Solomon codes, which detect problems the first time they occur.

According to the table (Table 3), RS codes have a very high resistance to noise, which represents a major factor for our study; moreover, RS codes have a great capacity to correct the greatest number of transmission errors. These codes are also used in on–off keying modulation in the proposed architecture; hence, the choice of these codes for error correction. The variable p is an integer representing the probability of the existence of noise.

Table 3. Comparison between some correction codes.

Codes	Coding	Decoding without Errors	Decoding with Errors	Noise Resistance
Reed–Solomon	15 ko/s	4.5 ko/s	4.5–1 ko/s	p = 50
Hamming	80	3	1	
CIRC	118	181	180–90 ko/s	p = 100

5. Conclusions and Future Works

The idea cited in this paper is to use the Berlekamp–Massey algorithm for the one time pad for the generation of pseudo-random keys, and whose algorithm has been modified by adding a degree of security. These pseudo-random keys will depend on the value of k, which represents an odd number different from 1 and is chosen by the sender of a message, and we will use this value thereafter to encode and decode the message. We will be able to tweak the algorithm such that the value of k varies with each communication and key creation, ensuring that the method will never be broken on the network because our notion is dependent on the value of k. The sender and receiver will use the same protocol to encode and decode the message during transmission. Although Li-Fi systems are more secure than RF systems, our approach is introduced as a security accomplishment for this system, and allows information to be exchanged securely across the network and to protect users, as long as each message has its own key, the sharing of these secret keys will be

conducted in a secure way. In terms of complexity, our method has a complexity of $O(L^2)$ for both encoding and decoding. Another advantage of this technique is that, even if the intruder understands the encryption algorithm, the intruder will be unable to decode a message, since he/she lacks the value of k required to encode and decode the message. According to our estimates, even if the method is changed, the error correcting aspect will remain unchanged. The suggested technique addresses the shortcomings of various current approaches based on generating secret keys, the most significant of which is the requirement to transmit the shared key. This must be done with extreme caution to avoid revealing the key to unwanted users. There might also be an issue with the amount of keys utilized. When you have a big number of keys, it might be tough to keep track of them all. This new approach allows a more secure communication through the network. Our motivation to propose this idea is to be able to disseminate this concept in hospitals, especially in this period of a global health crisis. The issue determines how to monitor each patient while maintaining the confidentiality of their information.

In future work, we will review the Manchester coding at the physical frame level and develop it to be able to code the outgoing signal. We also want to work more on OFDM modulation for secret key generation.

Author Contributions: Visualization, M.G.; Writing—original draft, L.H. and P.L.; Writing—review & editing, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

5G	5th generation
AP	access point
BAN	body area network
BM	Berlekamp–Massey
DC	direct current
FEC	forward error correction
Inner cc	internal counter encoder
IoT	internet of things
LAN	local area network
LCS	light communication systems
LED	light emitting diode
LFSR	linear feedback shift register
Li-Fi	light fidelity
MAC	medium access control
ODL	optical delay line
OFDM	orthogonal frequency division multiplex
OOK	on off keying
OTP	one time pad
PAN	personal area network
PD	photo-detection
PHY	physical layer
PLS	physical layer security
RLL	run-length limited
RS	Reed–Solomon
VLC	visible light communication

References

1. Farrel, A. Concepts in IP Security. In *The Internet and Its Protocols: A Comparative Approach*; Morgan Kaufmann Publishers: San Francisco, CA, USA, 2004; pp. 677–681.
2. Lorenz, P.; Hamada, L. LiFi Towards 5G: Concepts Challenges Applications in Telemedecine. In Proceedings of the 2020 Second International Conference on Embedded & Distributed Systems (EDiS), Oran, Algeria, 3 November 2020.
3. Haas, H. LiFi is a paradigm-shifting 5G technology. *Physics* **2017**, *3*, 26–31. [CrossRef]
4. Karunatilaka, D.; Zafar, F.; Kalavally, V.; Parthiban, R. LED Based Indoor Visible Light Communications: State of the Art. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 1649–1678. [CrossRef]
5. Blinowski, G. Security of Visible Light Communication systems—A survey. *Phys. Commun.* **2019**, *34*, 246–260. [CrossRef]
6. Hamada, L.; Lorenz, P. Li-Fi: A Revolution in Wireless Networking for Smart Communication Through Illumination. *Acta Sci. Comput. Sci.* **2021**, *3*, 53–58.
7. Al-Moliki, Y.M.; Alresheedi, M.T.; Al-Harthi, Y. Secret Key Generation Protocol for Optical OFDM Systems in Indoor VLC Networks. *IEEE Photonics J.* **2017**, *9*, 1–15. [CrossRef]
8. Zhang, Z.; Chaaban, A.; Lampe, L. Physical Layer Security in LiFi Systems. *Philos. Trans. A* **2020**. Available online: https://royalsocietypublishing.org/doi/full/10.1098/rsta.2019.0193 (accessed on 26 October 2021). [CrossRef] [PubMed]
9. Abdallah, W.; Krichen, D.; Boudriga, N. An optical backhaul solution for LiFi-based access networks. *Opt. Commun.* **2020**, *454*, 124473. [CrossRef]
10. Rajagopal, S.; Roberts, R.D.; Lim, S. IEEE 802.15.7 visible light communication: Modulation schemes and dimming support. *IEEE Commun. Mag.* **2012**, *50*, 72–82. [CrossRef]
11. Kwon, J.K. Inverse Source Coding for Dimming in Visible Light Communications Using NRZ-OOK on Reliable Links. *IEEE Photonics Technol. Lett.* **2010**, *22*, 1455–1457. [CrossRef]
12. Sonawane, S.; Baste, V.S. Implementation of RS-CC Encoder and Decoder using MATLAB. *IJSTE* **2019**, *5*, 22–30.
13. Shrivastava, P.; Singh, U.P. Error Detection and Correction Using Reed–Solomon Codes. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2013**, *3*. Available online: https://www.semanticscholar.org/paper/Error-Detection-and-Correction-Using-Reed-Solomon-Shrivastava-Singh/9a572d759e0d960355d6e2541d2a80d2e38df975 (accessed on 26 October 2021).
14. Wikipedia. Available online: https://en.wikipedia.org/wiki/One-timepad (accessed on 12 September 2021).
15. Taveneaux, A. Comparaison de l'efficacité de codes correcteurs d'erreurs. Available online: https://docplayer.fr/8544869-Comparaison-de-l-efficacite-de-codes-correcteurs-d-erreurs.html (accessed on 20 September 2021).