Secure Optical Communications Based on Fast Cryptography

Luis Velasco*, Masab Iqbal, and Marc Ruiz
Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
**e-mail: luis.velasco@upc.edu*

2023 23rd International Conference on Transparent Optical Networks (ICTON) | 979-8-3503-0303-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICTON59386.2023.10207201

ABSTRACT

Although security solutions, like Advanced Encryption Standard (AES) and ChaCha, are common at the packet layer, secure transmission at the optical layer is still not implemented. The reason is that such cryptographic methods are not fast enough to support high-speed optical transmission and might introduce significant delay. Moreover, methods for key exchange, key generation and key expansion need to be implemented on standard coherent transponders. In this paper, we summarize a secure cryptographic solution for optical connections named Light Path SECurity (LPsec), which involves fast data encryption using stream ciphers and key exchange using Diffie-Hellman (DH) protocol through the optical channel. To support encryption of high-speed data streams, a fast, general purpose Pseudo-Random Number Generator (PRNG) is used. Moreover, to make the scheme more secure against exhaustive search attacks, an additional substitution cipher is proposed. In contrast to the limited encryption speeds that standard stream ciphers can support, LPsec can support high-speed rates.
Keywords: secure optical communications; lightpath security.

1 INTRODUCTION

Optical networks are vulnerable to a variety of attacks such as eavesdropping, physical infrastructure attacks, interception, and jamming (refer to [1] for a survey on the topic). Designing a security solution involves combining multiple technologies for key distribution and data encryption and decryption, among others. For the former, although Quantum Key Distribution (QKD) provides secure key agreement, it is not expected to be available in the short term. With regard to encryption, there are several solutions, from *stream ciphers* (each incoming plaintext digit is encrypted sequentially) to *block ciphers* (where plaintext digits are grouped into blocks and then encrypted together). Salsa20 and ChaCha [2] are examples of stream ciphers and Advanced Encryption Standard (*AES*) [3] is the most extended block cipher. However, one of the main challenges for securing the optical layer is that encryption and decryption need to operate at line speeds of 100s of Gb/s and should not introduce any meaningful delay to the optical transmission. Some vendors implement AES at the Optical Transport Network (OTN) layer [4], where key exchange can be carried out using header bytes of the Optical Data Unit (ODU) frame. In this case, AES is used as a block cipher that encrypts blocks of 128 bits. Note that such a solution brings the additional requirement of implementing OTN, in addition to the intrinsic complexity of AES.

This paper summarizes our proposal for LPsec [5], an approach that includes tailored solutions for both key exchange and encryption/decryption so they can be easily implemented at the optical layer. For the key exchange, we design a mechanism based on the Diffie-Hellman (DH) key exchange [6], where the initial public keys of the two end parties, i.e., the Transmitter (Tx) and the Receiver (Rx), are exchanged via the Software Defined Networking (SDN) controller and are periodically updated through the optical channel to enhance the security level. For encryption, we rely on two ciphers: *i*) a traditional stream cipher that uses a symmetrical key and *ii*) permutations of symbols. Each cipher has its drawbacks, but when combined they provide the required security level to encrypt data at 100s of Gb/s. Besides, LPsec exhibits negligible transmission delay.

2 SECURE OPTICAL LAYER

This section first introduces some basic cryptographic concepts, such as ciphers, key extension, and key exchange [7], that are used in the rest of the paper. Next, the cryptographic techniques that we are proposing to implement for securing the optical layer are described.

A. Background on Cryptography

In cryptography, a cipher is an algorithm that transforms a plaintext message (m) into a ciphertext (c) (encryption), and vice versa (decryption). There are several types of ciphers, e.g., a *substitution cipher* encrypts units (e.g., each letter in a text) of plaintext by replacing them with the ciphertext with the help of a key. The receiver performs the inverse process to recover the original plaintext. Although the number of substitution alphabets might be large, substitution ciphers can be broken by frequency analysis.

To show that a cryptosystem is secure, mathematical modeling and proofs are used to verify that it satisfies a set of security properties. In particular, the One-Time Pad (OTP) cipher shows *perfect security*, as proved by Shannon in [8]. OTP encrypts m using a key (k) with the same length of m (denoted as n), by just implementing a bitwise XOR operation. Two important properties of XOR are: *i*) if k is uniformly distributed on $\{0,1\}^n$ then, the ciphered message $m \oplus k$ is also uniformly distributed; and *ii*) the inverse operation (decryption) consists on applying the XOR function with the same (symmetric) key k, i.e., $m \oplus k \oplus k = m$. Although OTP shows perfect

security, it does not fit well for stream ciphers, where the length of the messages tends to infinite. Semantic security provides a weaker notion of security that allows to build secure ciphers that use reasonably short keys. That entails splitting the data stream into chunks of data of predefined size. However, k cannot be reused from one data chuck to another, as that would reduce the security level. Nonetheless, k can be extended using a cryptographically secure PRNG to generate a sequence of stream keys (k_s).

Salsa and ChaCha are fast and secure stream ciphers that are appropriate for practical use and variants of them are being used in widely deployed protocols, such as Transport Layer Security (TLS) [9]. The PRNGs use a 256-bit seed, a 64-bit nonce, and a 64-bit counter to form a 512-bit block to create up to 2^{64} 512-bit pseudo random blocks. The design of these stream ciphers is highly parallelizable to speed-up encryption [2]. Block ciphers can be used as well to build a stream cipher; one popular block cipher is AES. In AES, an input block of 128 bits is processed as a 4×4-byte matrix. The AES algorithm performs 10, 12 or 14 rounds depending on the size of the cipher key (128, 192 or 256 bits). The process begins with the expansion of the initial key to produce a series of keys used in each round. At every round, the encryption begins by adding the round key as a XOR cipher followed by a non-linear byte substitution through a predetermined substitution table. Then, rows shifting followed by a mixing of columns and round key addition are performed. The procedure is repeated until completing the required number of rounds. Decryption is performed in the inverse manner.

The DH key exchange is a solution to exchange keys between two parties that want to establish a secure communication channel. Both parties generate private (integer) keys k_p (i.e., k_a and k_b) and their related public keys k_P (i.e., k_A and k_B). The public keys are shared over the insecure channel and each party computes the symmetric key k used for data encryption using their own private key k_p and their counterpart's public key k_P.

B. Implementing LPSec in an Optical Coherent System

LPsec requires extending the standard coherent transponder with optical encryption and decryption blocks, as well as with some key management functionalities. In addition, cryptographic blocks need to operate at line speeds and should not introduce any significant delay to data transmission. To achieve such an objective, optical encryption should be based on simple operations performed on the input bit stream. The main design aspects of the cryptographic techniques proposed in this paper are analyzed hereafter.

The encryption is based on two nested ciphers that provide a high security level. The outer cipher is a substitution cipher that relies on a Lookup Table (LUT) used for the substitution of bits before sending it to the modulator. This creates a ciphered gray map constellation through LUT permutations of incoming bits as suggested in Fig. 1a. Note that there are $M!$ permutations in an M- Quadrature Amplitude Modulation (QAM) system (e.g., there exist more than 2^{44} permutations in a 16-QAM system) and thus, we can use a random key (k_l) of the appropriate length (i.e., 44 bits in the example), to select the permutation of the LUT. The inner cipher is a stream cipher that encrypts data chunks of predefined size based on a cryptographically secure PRNG to generate a sequence of stream keys (k_s). The proposed encryption system is sketched in Fig. 1b, where output ciphertext c_2 is produced by the combination of the inner stream cipher E_1 and the outer substitution cipher E_2.

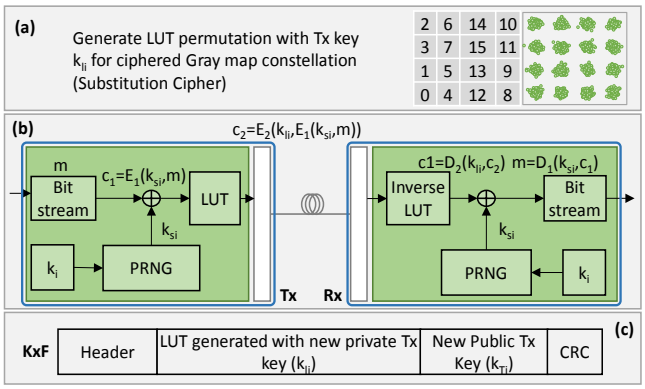

Figure 1. Overview of LPsec. Example of 16-QAM LUT for encoding (a), encryption / decryption (b), and frame structure for periodical LUT and key synchronization (c).

Note, however, that the sequence of stream keys $k_s = [k_{sj}]$ generated by the PRNG from a given key k cannot be infinite as this would reduce the security level of E_1. In addition, the LUT should be periodically regenerated to minimize the vulnerability of E_2. In consequence, we limit the lifetime of keys k and k_l, e.g., to 1 sec., which entails new keys being periodically generated at the Tx and exchanged with the Rx. Specifically, the DH key exchange method is used to generate the symmetric key k. The Tx and Rx generate a random private/public pair of keys ($\{<k_r, k_R>\}$ and $\{<k_t, k_T>\}$) and exchange their public keys (k_T, k_R) with the other party. The initial key exchange can be facilitated by the SDN controller, which, once the optical connection is computed and established in the network, can collect the public keys and send them to the counterpart. However, symmetric keys should have a short lifetime and need to be frequently updated, which makes the SDN not a suitable option.

In our approach, we perform a partial key exchange, where only the Tx generates a new pair of keys $<k_{ti}, k_{Ti}>$, as well as a new key k_{li} for the next period i. Next, both the public key k_{Ti} and the new permutation LUT(k_{li}) are sent to the Rx through the optical channel. We propose to use a special frame (henceforth called Key exchange Frame, KxF) for the key exchange, (see Fig. 1c). A KxF is generated by the Tx and sent to the Rx periodically. The KxF includes a header of a fixed size that allows the Rx to detect its arrival. Because the Rx is not synchronized with the Tx for key exchange, any occurrence of the header pattern in the data stream must be

prevented at the Tx side. Otherwise, the Rx would follow an erroneous key exchange procedure that will stop data transmission. The solution is to add escape bit sequences to break any KxF header pattern in the input data. To this end, two *Finite State Machines* (FSM) at the Tx and the Rx sides add and remove such escape bit sequences to/from the plain bit stream. Note that the occurrence of errors in the KxF is critical. In case errors still remain after the FEC decoding stage and impact the KxF, the decryption process should be restarted. For this very reason, the KxF includes a cyclic redundancy check (CRC).

3 KEYS AND SECURITY LEVEL

This section first details the key exchange process, including the initial exchange and the periodical updates. The generation and expansion of symmetric keys is detailed next, after which security level of the system is studied.

A. Key Exchange

As introduced in Section B, an initial key exchange is performed through the SDN controller and then, the Tx updates the Rx with the keys to be used through the optical channel. Figure 2 presents a sequence diagram detailing the computation performed by the Tx and Rx, as well as the messages exchanged through the control plane and the data and messages sent over the encrypted optical channel.

The initial key exchange is carried out at connection set-up through the SDN controller (messages 1-5 in Fig. 2), which collects the public key of the Rx (1) and sends it to the Tx (2). The Tx generates a pair of private and public keys and a random key k_{l0}, which is used to generate the initial LUT permutation. In addition, the Tx generates the symmetric key k_0 with its private key and Rx's public key. Key k_0 is used at this time to generate the particular KxF header pattern that will be used for key exchange on the optical channel. Both FSM_{Tx} and FSM_{Rx} must be generated for that specific pattern. Before sharing the LUT, it is encrypted with symmetric key k_0 and sent together with the Tx public key to the SDN controller (3), which shares them with the Rx (4). Upon the reception, the Rx generates the symmetric key k_0 with its private key and Tx public key, generates the FSM, and decrypts the LUT. The Rx replies to the SDN controller when it is ready to start the secure communication and the SDN controller notifies the Tx (5), which generates a new set of private and public keys, the LUT, and the symmetric key for the next time interval. Then, the Tx replies to the controller that it is also ready, and the initialization phase concludes. At this time, the secure optical connection is established.

Once the initialization phase ends, the secure optical transmission phase begins and continues until the connection is torn down. At the starting time, the Tx updates the LUT and public key (6). Note that such exchange is encrypted using the nested encryption used for data transmission; in this case, keys k_{s0} (extended from symmetric key k_0) and k_{l0} are used for ciphers E_1 and E_2, respectively (6).

We denote with subindex i the keys that participate in key exchange at the starting of time period i. This entails that during every time period i-1, the Tx generates the set of public and private keys ($<k_{ti}$, $k_{Ti}>$), the symmetric key k_i, and the random key k_{li} (and the permutation of the LUT). Then, at the start of time interval i, the Tx updates the Rx, which computes symmetric key k_i. The exchanged keys will be in place during time period i+1. In particular, key exchange i and all data transmitted during time period i are encrypted using keys i-1.

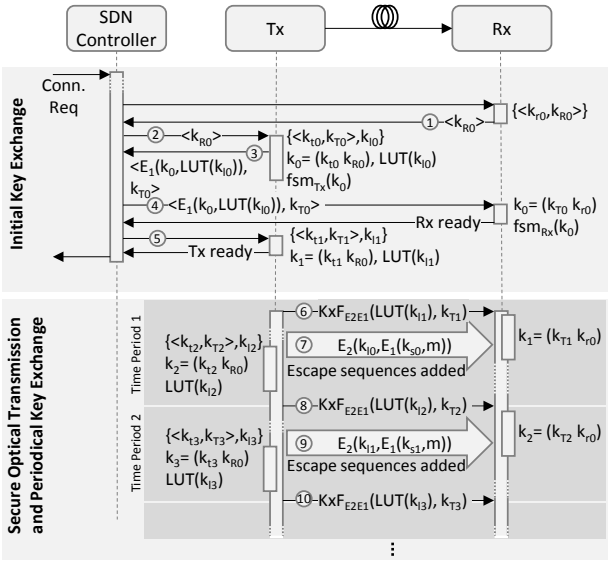

Figure 2. Connection set-up and secure optical transmission.

B. Symmetric Key Generation and Expansion

In the standard DH key exchange, two large prime numbers (p and g) are publicly selected. When Alice and Bob want to setup a secure communication channel, they generate private keys k_p, which are used to compute their public keys k_P (eq. (1)). After the public keys are exchanged, each party computes the symmetric key k for data encryption/decryption (eq. (2)). Similarly, in LPsec, Tx and Rx exchange their public keys k_{T0} and k_{R0} through the SDN controller during the initial key exchange phase. Next, the Tx computes a new pair of keys every period and updates the Rx with the new public key k_{Ti}. (eq. (3),(4)). Note that the Rx does not compute new public and private keys afterwards, and the initial pair $<k_{r0}$, $k_{R0}>$ is used along the lifetime of the optical connection. Thus, the symmetric key k that is used for the stream cipher is updated periodically (e.g., every 1 sec.) (eq. (5)).

$$k_P = g^{k_p} \bmod p \quad (1) \qquad k = {k_B}^{k_a} \bmod p = {k_A}^{k_b} \bmod p \quad (2)$$

$$k_{R0} = g^{k_{r0}} \bmod p \quad (3) \qquad k_{Ti} = g^{k_{ti}} \bmod p \quad \forall i \quad (4)$$

$$k_i = {k_{R0}}^{k_{ti}} \, mod \, p = {k_{Ti}}^{k_{ro}} \, mod \, p \quad (5) \qquad J = \frac{B}{U} \quad (6)$$

Once the symmetric key is computed, it is expanded using a cryptographically secure PRNG to produce keys long enough for the stream cipher to encrypt / decrypt a chunk of data. Therefore, if the size of each chunk of data is U [b] and the transmission speed is B [b/s], then the number of chunks per second, J, can be computed as eq. (6). Hence, each symmetric key k_i generated for the time interval i, is expanded into J k_{sij} keys that the stream cipher will use for chunks j in [0 .. J-1]; keys k_{sij} are U bits long. For example, assuming that the size of data chunks is U=64 bits, the transmission speed is B=100 Gb/s, a new key is generated every 1 second, the PRNG needs to expand the symmetric key k into J=$2^{30.5}$ keys k_{sij} per second (i.e., one key every 0.64 ns), each U bits long. Therefore, the PRNG must be fast enough to work at 100s of Gb/s line speeds, in order not to introduce meaningful delay to data transmission.

C. Security Level and Encryption Speed

Let us assume that, under the DH protocol, an eavesdropper (i.e., Eve) knows the value of p, g, and the public keys of Alice and Bob. To compute the symmetric key, Eve still needs to know the private keys of either Alice or Bob, or to solve the discrete logarithm problem [10], which is considered computationally hard when p is large. However, the security level of a stream cipher depends on the randomness of the PRNG for key expansion [7]. Recall that OTP shows perfect security since ciphertexts do not reveal any information of the related plaintext, so an adversary cannot distinguish between two ciphertexts m_i and m_j encrypted with key k selected at random. However, stream ciphers cannot attain perfect security because PRNGs are utilized, and the length of the generated keys are shorter than those of the messages.

A PRNG is secure if an adversary cannot distinguish between a truly random sequence and the pseudo random sequence generated by the PRNG with a significant advantage. This is related to the computational feasibility of adversaries to perform predictions with a reasonable amount of time and memory. In practice, although standard stream ciphers based on Salsa20 and ChaCha produce high-quality PRNGs, they are not fast enough to be applied to optical transmission. In contrast, we use a general-purpose PRNG because of its high speed. To mitigate the impact of using a general-purpose PRNG only, an optical constellation-based substitution cipher is added. This approach is still not perfectly secure, as the distribution of the encrypted data is not uniform. Therefore, if the characteristics of the plain text are known, an adversary can apply frequency analysis and break the cipher. However, since the substitution cipher is fed with data encrypted using the XOR operation, frequency analysis will not provide useful information. As a consequence, this symbiotic relationship between the stream cipher and the substitution cipher results in a fast and secure cryptographic system.

4 CONCLUSION

A complete solution to add encryption at the optical connection level has been presented in this paper. The solution includes a mechanism for key distribution from the Tx to the Rx and two ciphers that, when combined, can provide the required security level and are able to work on 100s of Gb/s data flows. The key distribution is performed using a Key exchange Frame that is sent periodically from the Tx to the Rx. Note that the proposed key exchange enables implementing security at the optical layer, although can be substituted as soon as other key exchange mechanisms, like QKD, become available. The Xoshiro256+ PRNG is used for symmetric key expansion, so as to provide keys that are used by the first XOR-based cipher. A second cipher based on LUT substitution improves the security level. The design of LPsec has been presented and is easily implementable on current coherent optical systems.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Commission HORIZON ALLEGRO (G.A. 101092766) and the AEI IBON (PID2020-114135RB-I00) projects and from the ICREA institution.

REFERENCES

[1] M. Fok, Z. Wang, Y. Deng, and P. Prucnal, “Optical layer security in fiber-optic networks,” *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 725-736, 2011.
[2] D. Bernstein, “ChaCha, a variant of Salsa20,” in Workshop Record of SASC, 2008.
[3] “Specification for the Advanced Encryption Standard (AES),” FIPS-197, NIST, 2001.
[4] ADVA Layer 1 security. [On-line] https://www.adva.com/en/innovation/ network-security/layer-1-security.
[5] M. Iqbal, L. Velasco, N. Costa, A. Napoli, J. Pedro, and M. Ruiz, “LPsec: A fast and secure cryptographic system for optical connections,” *IEEE/OPTICA J. of Optical Comm. and Networking (JOCN)*, vol. 14, pp. 278-288, 2022.
[6] D. Neuenschwander, “Diffie-Hellman key exchange,” *Probabilistic and Statistical Methods in Cryptology*, 2004.
[7] D. Boneh and V. Shoup, A Graduate Course in Applied Cryptography, [On-line] http://toc.cryptobook.us, 2020.
[8] C. Shannon, “Communication theory of secrecy systems,” *Bell Labs Tech. J.*, vol. 28, pp. 656-715, 1949.
[9] Y. Nir and A. Langley, “ChaCha20 and Poly1305 for IETF Protocols,” IRTF RFC-8439, 2018.
[10] H. Corrigan-Gibbs and D. Kogan “The discrete-logarithm problem with preprocessing,” in Proc. *EUROCRYPT 2018*.