Cryptographic Schemes for Secret Long-Distance Underwater Communications

Michel Barbeau
School of Computer Science, Carleton University, Ottawa K1S 5B6, Canada
Email: barbeau@scs.carleton.ca (M.B.)

***Abstract*—Due to small packet sizes, classical data protection schemes are unsuitable for underwater communications. This article addresses this problem and contains two main results. As a first result, a new symmetric-key encryption protocol adaptable to small message sizes is introduced. The encryption scheme leverages the flexible Quantum Permutation Pad (QPP) symmetric key block cipher. It combines QPP with the block cipher counter mode and a random number generator seeded with a shared secret to adapt QPP to the short underwater protocol data units. Encryption and decryption algorithms are defined, building on QPP in counter mode. The algorithms are analyzed. The analysis demonstrates that the scheme does not achieve perfect indistinguishability. However, the analysis also demonstrates that the message collision probability can be very low. The scheme is generic and adaptable. As a second result, the new symmetric encryption scheme is adapted to the long-range underwater communication protocol (Pronounced you Whisper) UWSPR. The design is analyzed consistently with the theory. Related relevant issues are also addressed, such as key sizes and key generation with the challenges specific to the underwater environment.**

***Keywords*—underwater communications, underwater network, security, confidentiality, encryption, quantum permutation pad, (Pronounced you Whisper) UWSPR**

I. Introduction

Underwater communications find applications in many sectors of human activities, including sensor networks monitoring aquatic life and maritime traffic. Acoustic waves are the data carriers underwater, unlike the classical free-space wireless environment where electromagnetic waves are used. Acoustic waves propagate much farther but at narrow bandwidths and low data rates. Another important observation is that acoustic waves travel much slower than electromagnetic waves. Slow speed and narrow bandwidth result in small data rates and small packets. For instance, a underwater protocol named after the Roman God of openings and Gateways (JANUS) baseline packet is 64 bits [1, 2], while a UWSPR frame is 50 bits [3].

Undersea long-distance acoustic communication, with ranges of tens or hundreds of kilometers, is a hazardous environment. Underwater communication nodes operate on their own in an open environment. Eavesdroppers, injectors of noise and false messages, and rogue nodes may access the environment. In long-distance underwater communications, processing time performance carries a much lower weight than transmission time. The data rate of JANUS, at a center frequency of 11,520 Hz, is 80 bits per second (bps). For UWSPR, the data rate goes from 0.05 to 0.9 bps, according to the mode. At these rates, the available processing time enables the use of time-consuming signal search techniques and, eventually, data protection methods.

While most of the past studies on underwater communications emphasized increasing performance [4], this paper focuses on ensuring the confidentiality of data transmitted in the underwater environment. Due to small packet sizes, classical data protection schemes, such as the Advanced Encryption Standard (AES) [5], are not adapted to underwater communications. AES is a 128-bit block cipher scheme. Schemes that can handle smaller blocks are needed for the underwater context.

Building on a universal encryption scheme called Quantum Permutation Pad (QPP) [6] and data block encryption in counter mode [7], a symmetric-key encryption scheme for small blocks adapted to underwater protocol data units is proposed. The new scheme is analyzed. A use case for the UWSPR protocol is developed. UWSPR has been specifically designed to support long-range undersea communications. The article comprises two main results. Firstly, generic symmetric-key encryption and decryption algorithms are defined using the QPP block cipher in counter mode. The scheme does not achieve perfect indistinguishability but has a very low probability of message collision. The scheme is particularly well adapted to small message sizes of underwater communication protocols. Secondly, the decryption and decryption algorithms are customized to fit the constraints of the UWSPR protocol. It is a challenge because the frame size is limited to 50 bits.

The major contribution of this article is a new symmetric-key data encryption protocol adapted to the small frame size format required for long-range underwater communications. Confidentiality is achieved by defining a block encryption scheme using a binary-value permutation and the counter mode.

The rest of the paper is structured as follows. Section II reviews related work. Section III describes the new

Manuscript received February 6, 2023; revised June 5, 2023 accepted August 10, 2023; published April 8, 2024.

encryption scheme. Use cases are developed in Section IV. We conclude with Section V.

II. Related Work

Research efforts on underwater communications have addressed physical, link, and network layer issues. Much research has examined increasing data rates and reliability [8, 9]. This paper focuses on an equally important issue: the confidentiality of data transmitted using underwater media.

A. Secure Underwater Communications

In the underwater environment, payloads are short. Encryption schemes for classical data do not work well with underwater protocols because they are designed for relatively large payloads. For example, JANUS's 64-bit small baseline packet size does not work well with contemporary cryptographic schemes, such as AES, with its minimum 128-bit block size [5]. Stream ciphers may be considered, considering that plaintexts must be encrypted with different keystreams. This requires frequent key updates; every message needs a new key or an Initialization Vector (IV) with a unique value for each message. Frequent key updates are hard to conduct in a low-bandwidth underwater environment. On the other hand, using an IV field augments the amount of information to be transmitted in a situation where there is little room for overhead.

North Atlantic Treaty Organization (NATO) has created a standard underwater communication architecture named JANUS [1, 2]. The goal is to create a common protocol enabling interoperability between equipment from different equipment vendors. JANUS has two ways to send user data: the Application Data Block (limited to 34 bits) and the optional cargo payload (user-defined size). To encrypt the Application Data Block, JANUS has been augmented with Venilia [10]. Venilia builds on symmetric cryptography, i.e., a cryptosystem named Tiny Underwater Block cipher [11]. To encrypt JANUS's optional cargo payload, an encryption scheme called the AES-Galois/Counter Mode (AES-GCM) has been proposed [12, 13]. The scheme is a 128-bit block cipher. It uses a 256-bit symmetric key together with a 128-bit IV. This IV comprises a 112-bit static pre-configured part and a 16-bit dynamic part. The dynamic part comprises a four-bit node ID and a 12-bit counter. The size of the node ID limits the number of participants to 16, while the size of the counter restricts each participant from sending no more than 4,096 messages. Hamilton et al. tested the approach in a sea trial with cargo payloads of up to 60 bytes [14]. This approach protects packet payloads. Hamilton et al. emphasized the need for methods protecting the entire packets. Indeed, Venilia does not protect the data outside the Application Data Block, while AES-GCM only protects the optional cargo payload with the 128-bit block size constraint.

For their underwater communication system, Caiti *et al*. [15] and Dini and Duca [16] used ciphertext stealing [17]. This scheme mitigates ciphertext expansion caused by plaintext padding in block cipher schemes. Peng *et al*. [18] created an encryption scheme building on chaos theory for underwater communications. Much emphasis had been put on low processing complexity. Their block size is 64 bits.

B. Quantum Threat to Cybersecurity

The quantum threat to cybersecurity is twofold. It menaces modern cryptographic protocols aiming to achieve confidentiality, key establishment, and digital signature. Firstly, concerning cryptographic protocols, the threat comes from Grover's algorithm, which accelerates the search in a list of $\lceil n \rceil$ items from the order of $\lceil n \rceil$, with classical computing, down to the square of $\lceil n \rceil$, with a quantum computer [19]. Grover's quantum algorithm can accelerate the adversarial search for a cryptographic key. To mitigate that threat, the National Institute of Standards and Technology (NIST) recommends using keys longer than or equal to 256 bits [20]. No quantum threat other than Grover's algorithm is known for symmetric encryption. It can be addressed with long key sizes.

Secondly, concerning key establishment and digital signature protocols, several cryptographic schemes rely on the NP-hardness of factoring a large integer into prime numbers. Shor's quantum algorithm can resolve that problem efficiently on a quantum computer [21]. Several experts agree that RSA-2048 is likely to be broken by quantum computing within the next 15 years [22]. To address this threat, post-quantum cryptography aims to create schemes that build on problems that cannot be solved efficiently by known quantum algorithms. Therefore, to ensure quantum resistance, long keys are an essential requirement of symmetric encryption [23, 24]. Quantum Permutation Pad (QPP) can support very long keys.

C. Long-Distance Underwater Communications

Note that JANUS has not been designed for long-distance communications. We focus on UWSPR, designed for long-range underwater communications [3]. It is a two-layer protocol architecture: physical and link. Four-tone frequency-shift keying and non-coherent demodulation are employed at the physical layer. Phase information recovery is not needed for demodulation. Every channel bit is paired with a synchronization bit to make the frame search by a receiver easier. The link layer supports six frame formats, making the protocol adaptable to channel conditions. The frame formats share high-constraint, convolutional-coding Forward Error Correction (FEC). The receiver uses soft symbols and sequential decoding, which can correct very weak and noisy signals. Every frame contains 50 data bits. A frame can take 1, 2, 4, 8, 10, or 20 minutes to transmit, depending on the format. The frame format can be chosen according to the communication conditions and target distances. Robust communications are achieved using long-lasting channel symbols, containing energy, and mitigating the effect of underwater multipath propagation. The system has been tested in several underwater sea trials on the Canadian west coast, Canadian Arctic, and the Greenland Sea. Low frame error rate communications have been achieved underwater using acoustic waves across distances well above 30 kilometers. This performance is achieved at the expense of low data rates and small frames. For instance, a 64 bits

JANUS frame takes one second two send. While a UWSPR 50 bits frame takes at least one minute.

III. Encryption Scheme

An encryption scheme for confidential underwater communications is introduced. It comprises two aspects: a block cipher and a mode of operation. The notation and terminology of Bellare and Rogaway are used [7]. This section develops the foundations that are applied to UWSPR in the next section.

A. Quantum Permutation Pad

The QPP symmetric key block cipher is used [6]. A plaintext M consists of m words $\omega_0, \omega_1, \dots, \omega_{m-1}$, each of size n bits. Every word is an n qubits block. The plaintext is encrypted with m randomly selected permutations $P_0, P_1, \dots, P_{m-1}$ in the symmetric group S_{2^n}. A permutation P in the symmetric group S_{2^n} is a bijective function with the domain and co-domain $D = \{0,1, \dots, 2^n - 1\}$. There are $2^n!$ permutations in S_{2^n}. The encryption key is the sequence π of permutations equals to $P_0, P_1, \dots, P_{m-1}$. The decryption key is the sequence of inverse permutations $P_0^T, P_1^T, \dots, P_{m-1}^T$. The encryption of message M is equal to $E_\pi(M)$=$P_0(\omega_0), P_1(\omega_1), \dots, P_{m-1}(\omega_{m-1})$. The decryption of a ciphertext message C is denoted as $D_\pi(C) = P_0^T(\omega_0), P_1^T(\omega_1), \dots, P_{m-1}^T(\omega_{m-1})$, where C consists of m words $\omega_0, \omega_1, \dots, \omega_{m-1}$, each with a size of n bits.

Definition 1. [Shannon perfect secrecy] A cryptographic scheme is perfectly secure when the encryption key is single-use and selected randomly. For any pair of plaintexts, M_1 and M_2, a ciphertext C is equally likely to be the encryption of M_1 or M_2.

Theorem 1. *QPP is perfectly secure*

Proof. For a fixed ciphertext C, a random single-use key $\pi = P_0, P_1, \dots, P_{m-1}$, and for any message M, we have

$$Pr[E_\pi(M) = C] = Pr[P_0(\omega_0), P_1(\omega_1), \dots, P_{m-1}(\omega_{m-1}) = C]$$

$$= \frac{|\{\pi \in (S_{2^n})^m : P_0(\omega_0), P_1(\omega_1), \dots, P_{m-1}(\omega_{m-1}) = C\}|}{|S_{2^n}|^m}$$

$$= \frac{|S_{2^{n-1}}|^m}{|S_{2^n}|^m} = \frac{(2^{n-1}!)^m}{(2^n!)^m}$$

Probabilities are uniform for all plaintexts M.

Note that QPP is neither a bit permutation nor a substitution cipher [7]; both leak information. Furthermore, the key size ($m \log_2 2^n!$) is larger than the message size (mn).

Theorem 1 is a theoretical result. To make QPP practical and adapted to small underwater protocol data units, we combine it with the block cipher counter mode, a random number generator seeded with a shared secret, and a key establishment protocol.

B. QPP Block Cipher in Counter Mode

The mode of operation defines how the encryption is performed using a block cipher. The available modes of operation are reviewed in detail by Bellare and Rogaway [7]. There are four main options: Electronic Code Book (ECB), Cipher-Block Chaining (CBC) with random IV, Counter-Based version of CBC (CBCC), and Counter (CTR). The ECB and CBCC modes can be ruled out due to the significant risk of information leakage. With the former, ciphertexts can be associated with plaintexts. With the latter, the value of the counter is predictable. CBC with random IV is secure, assuming a secure pseudo-random function for the block cipher. The CTR mode comes in two flavors: random and stateful. Because of a random starting point, the CTR random mode has a risk of collision. The CTR Stateful (CTRC) mode achieves perfect indistinguishability. Both the CTR random and stateful modes are parallelizable. Bellare and Rogaway highlight that encryption modes must be probabilistic and state information dependent [7]. Every plaintext should have several possible ciphertexts. CBC with random IV and the two CTR modes fulfill these conditions.

Algorithm 1 illustrates QPP encryption in the CTRC mode. The encryption and decryption algorithms share a secret sequence of permutations π, one permutation P_i for every value of i in the range $0, \dots, 2^n! - 1$. All permutations in the symmetric group S_{2^n} are used, in a secret random order. The static counter variable (i) is initialized to zero. The counter is an index on the $2^n!$ permutations in the sequence π. Every permutation is used once, index reuse is not allowed. The encryption of a m words message M consumes m permutations. The result of encryption comprises the value of the counter and ciphertext sent together to the decryption algorithm. The reuse of a permutation is not possible. Encryption is unsuccessful when all permutations in the sequence π have been used.

Algorithm 1. QPP Encryption in CTRC Mode

static $i \leftarrow 0$

$E\pi$(M)

if $i + m - 1 \geq 2^n!$ **then return** unsuccessful

$\leftarrow P_i(\omega_0), P_{i+1}(\omega_1), \dots, P_{i+m-1}(\omega_{m-1})$

$oldi \leftarrow i$

$i \leftarrow i + 1$

return $\langle oldi, C \rangle$

Algorithm 2 defines the behavior of the decryption algorithm. It receives a pair comprising a counter (i) and a ciphertext $C = \omega_0, \omega_1, \dots, \omega_{m-1}$. The counter makes it possible to retrieve the right m permutations and decrypt using their transposes into the plaintext M, which is returned.

Algorithm 2. QPP Decryption in CTRC Mode

$D\pi(\langle i, C \rangle)$

if $i + m - 1 \geq 2^n!$ **then return** unsuccessful
$\leftarrow P_i^T(\omega_0), P_{i+1}^T(\omega_1), \dots, P_{i+m-1}^T(\omega_{m-1})$
return M

Definition 2. [Indistinguishability] With indistinguishability under a chosen-plaintext attack, an adversary picks two same-length plaintext messages. They

are both submitted to an encryption oracle. One of them is randomly chosen and encrypted by the oracle. The corresponding ciphertext is returned to the adversary. The encryption scheme used to produce the ciphertext is considered secure when the adversary cannot make better than a random guess to determine which of the two messages is encapsulated in the ciphertext.

Theorem 2. *QPP encryption in the CTRC mode achieves perfect indistinguishability*.

Proof. See Ref. [7] for a complete analysis of the CTRC mode. Mainly, it follows from the fact QPP in the CTRC mode uses the family of all permutations on 2^n bits numbers in random order with no repetitions.

QPP in the Counter Stateful (CTRC) mode has a very strong security property but requires a way to store $2^n!$ secret shared permutations of 2^n numbers of n bits numbers, which in the worst case takes $2^n!\, 2^n n$ bits. It is realistic for a small block size, such as n equals three. $2^n!\, 2^n n$ is 967,680 bits, or 120,960 bytes. It can also be made smaller by generating the permutations from their index using the factoradic method. $2^3!$, or 40,320, blocks can be encrypted with the same key. For n greater than three, it becomes a memory challenge. For instance, $2^4!$, or $16!$, is 20,922,789,888,000.

Unlike bitwise XOR-based ciphers, QPP does not leak information when a key is reused for two messages. This is because given two permutations $P_1, P_2 \in S_{2^n}$, their commutator $[P_1, P_2]$ is in general non-null. A version of QPP, not requiring all permutations and permitting the reuse of permutations, is shown in Algorithms 3 and 4. It is called QPP encryption in pseudo-counter mode. The encryption and decryption algorithms share a secret arbitrary long sequence r of random numbers, modulo d, and d secret randomly selected permutations $P_0, \dots, P_{d-1}$, with $0 < d \ll 2^n!$. The counter is used as an index over the permutations, selecting m permutations during encryption and selecting the same permutations for decryption. The value of the counter is sent together with the ciphertext. The decryption algorithm receives both.

Algorithm 3. QPP Encryption in Pseudo-counter Mode

static $i \leftarrow 0$
$E\pi$(M)
$C \leftarrow P_{r(i)}(\omega_0), P_{r(i+1)}(\omega_1) \dots, P_{r(i+m-1)}(\omega_{m-1})$
$oldi \leftarrow i$
$i \leftarrow i + 1$
return$\langle oldi, C\rangle$
if $i + m - 1 \geq 2^n!$ **then return** unsuccessful
$\leftarrow P_i^T(\omega_0), P_{i+1}^T(\omega_1), \dots, P_{i+m-1}^T(\omega_{m-1})$
return M

Algorithm 4. QPP Decryption in Pseudo-counter Mode

$D_\pi(\langle i, C\rangle)$
$M \leftarrow P_{r(i)}^T(\omega_0), P_{r(i+1)}^T(\omega_1) \dots, P_{r(i+m-1)}^T(\omega_{m-1})$
return M

This version requires a random number generator, a secret seed, and secret sharing of d permutations, which requires a maximum of $d2^n n$ bits. The indices of the secret permutations can also be generated from the shared number generator seed and a random number generator. In that case, the secret key consists of the random number generator seed and the d permutations. QPP in pseudo-counter mode does not achieve perfect indistinguishability because of the possibility of collisions.

Definition 3. [Collision] A collision occurs when at least one message value M consisting of m words $\omega_0, \omega_1, \dots, \omega_{m-1}$, each of size n bits, is re-encrypted with the same permutation sequence $P_0, \dots, P_{m-1}$ chosen in the symmetric group S_{2^n}. The same permutations, in the same order, are picked twice to encrypt a given message value.

Collisions are undesirable because they enable the identification of traffic patterns that can leak information and eventually be exploited to break the encryption scheme. The number of mn bits message value, m permutations sequence unique pairs is $2^{mn}d^m$. When more than $2^{mn}d^m$ messages have been encrypted with the same key, the probability that at least one collision occurred is one. We determine this probability when i, the number of messages of size nm bits encrypted with m permutations chosen among d different permutations, is less than equal to $\sqrt{2^{mn}\, d^m}$.

Theorem 3. Let i be the number of messages of m words of size n bits encrypted using QPP in pseudo-counter mode with a sequence of m permutations chosen among d permutations in the symmetric group S_{2^n}. Let i be greater than zero but less than equal to $\sqrt{2^{mn}\, d^m}$. The probability of at least one collision, when i messages of m words of size n bits are encrypted with m permutations chosen among d denoted as $C(d, m, n, i)$, is at least $0.6\frac{i(i-1)}{2^{mn+1}d^m}$ and at most $\frac{i(i-1)}{2^{mn+1}d^m}$.

Proof. Let us assume that all mn bits message values are equally likely, and that the selection of permutations is perfectly uniform and random. Moreover, the probabilities are independent across messages. We first prove the lower bound. Let f_i be the event denoting the absence of collisions after the encryption of i messages. When no collisions occurred after encrypting i messages, then i unique mn bits message value, m permutations sequence pairs have been used. Among the available $2^{mn}d^m$ message value, permutations sequence pairs, solely $2^{mn}d^m - i$ pairs have not been used. Hence, the probability of no collision when making the $i+1$ th encryption is

$$Pr[f_{i+1}|f_i] = \frac{2^{mn}d^m - i}{2^{mn}d^m} = 1 - \frac{i}{2^{mn}d^m}.$$

Therefore, the probability of no collisions after completing the encryption of i messages is

$$1 - C(d, m, n, i) = Pr[f_i] = Pr[f_i|f_{i-1}] \times Pr[f_{i-1}] =$$

$$\dots$$

$$= \prod_{k=1}^{i-1} Pr[f_{k+1}|f_k] = \prod_{k=1}^{i-1}\left(1 - \frac{k}{2^{mn}d^m}\right).$$

Using the inequality $1 - x \leq e^{-x} \leq x$, when $0 \leq x \leq 1$, the fact that $0 < \frac{k}{2^{mn}d^m} \leq 1$, and equality $\sum_{k=1}^{i-1} k = \frac{i(i-1)}{2}$, the above product is less than equal to

$$\prod_{k=1}^{i-1} e^{-\frac{k}{2^{mn}d^m}} = e^{-\frac{i(i-1)}{2^{mn+1}d^m}}.$$

Which means that

$$C(d,m,n,i) \geq 1 - e^{-\frac{i(i-1)}{2^{mn+1}d^m}}.$$

Using the inequality $(1-e^{-x}) \geq \left(1-\frac{1}{e}\right)x$, the above is greater than

$$\left(1-\frac{1}{e}\right) \times \frac{i(i-1)}{2^{mn+1}d^m} \geq 0.6 \times \frac{i(i-1)}{2^{mn+1}d^m}.$$

We may conclude that $C(d,m,n,i) \geq 0.6 \times \frac{i(i-1)}{2^{mn+1}d^m}$.

We now prove the upper bound. Let e_i denote the event that the i-th block encryption is a collision. Among the available $2^{mn}d^m$ message value, permutations sequence pairs, only $i-1$ have been used. We have that the probability of e_i, that is, $Pr[e_i]$, is at most $\frac{i-1}{2^{mn}d^m}$.

Furthermore, there are $2^{mn}d^m$ message value, permutations sequence pair combinations. Hence, we have that

$$\begin{aligned} C(d,m,n,i) &= Pr[e_o \vee e_1 \vee \ldots \vee e_{i-1}] \\ &\leq Pr[e_o] + Pr[e_1] + \cdots + Pr[e_{i-1}] \\ &\leq \frac{1}{2^{mn}}\left[\frac{0}{d^m} + \frac{1}{d^m} + \cdots + \frac{i-1}{dd^m}\right] = \frac{i(i-1)}{2^{mn+1}d^m}. \end{aligned}$$

$C(d,m,n,i)$ represents the probability of at least one collision, when i messages of m words of size n bits are encrypted with m permutations chosen among d permutations in the symmetric group S_{2^n}. The operator $\vee$ denotes the logical or. The term $Pr[e_o \vee e_1 \vee \ldots \vee e_{i-1}]$ means the probability of the independent events $e_o, e_1, \ldots, e_{i-1}$. Not surprisingly, the larger the block size (n), the larger the number of blocks (m), and the larger the number of permutations (d), the lower the risk of collision when the i-th block is encrypted.

QPP in pseudo-counter mode assumes that the plaintext is random. When not, a diffusion phase before encryption, on the encryption algorithm side, and an assembly phase after decryption, on the decryption algorithm side, can be inserted to remove any statistical bias in the ciphertext. This diffusion and assembly can also be driven by the random number generator producing a bit stream xored with the text.

C. Key Size and Key Generation

For QPP in pseudo-counter mode, the secret key consists of d permutations over n bits binary blocks and a value for seeding the random number generator. The $2^n!$ permutations can be indexed. Stirling's approximation says that $m!$ is approximately equal to $\sqrt{2\pi m}\left(\frac{m}{e}\right)^m$. Hence, to represent a number between one and $m!$, at most

$$\left(m+\frac{1}{2}\right)log_2 m + \frac{1}{2}log_2\left(\frac{2\pi}{e}\right)$$

bits are needed. Therefore, with $m = 2^n$, the d secret permutations occupy

$$d \times \left[\left(2^n + \tfrac{1}{2}\right)n + \tfrac{1}{2}log_2\left(\tfrac{2\pi}{e}\right)\right] \text{ bits.}$$

The parameters d and n must be chosen such that the secret is at least 256 bits to meet the 256-bit minimum key size requirement for quantum safety [20]. Recent works by He *et al.* [25] and Lou *et al.* [26] suggested values such as four or eight for n, with the corresponding values eight and 64 for d. However, in the underwater environment, the network is slower than the processors, unlike standard computing, where the network can often absorb more data than a single computer can produce. Hence, processing time is available in the underwater environment relative to transmission time. Furthermore, relative to the classical environment, the available computer memory is large with respect to protocol data unit sizes. Relatively large values for d and/or n can be used.

The permutations can be generated with Fisher and Yates algorithm [27]. The algorithm relies on random number generation. A truly random source of numbers must be used to make key generation reliable.

The establishment and renewal of long keys are challenging in the underwater environment. The payload of UWSPR frames cannot be greater than 50 bits. The application data block of JANUS is limited to no more than 34 bits. Key establishment and renewal are problems when secrets are long relative to the protocol data unit size. To bootstrap security, network participants can be pre-configured with master keys used to derive session keys. Section III highlighted the importance of key renewal to mitigate the risk of information leakage. Generating new session keys can be done in three different ways, which can be used individually or in combination. First, session keys can be updated using material already obtained and new material fitting in the payload of the small protocol data units. Second, clock data can be used as input material to the key generation procedures, assuming that network participants are synchronized. Data can also be derived from unique environment characteristics shared by network participants. For the first approach, a solution has been proposed by Beaupré *et al.* [28] used a small number of packets to refresh a symmetric secret encryption key. All nodes of a subnetwork share a master key. It is used to encrypt and distribute limited-lifetime session keys, which are used for data traffic encryption in the subnetwork. The key establishment protocol adapts the SSLv2 (Secure Sockets Layer version 2) export cipher scheme [29]. Let the master and session keys be m bits long, $m > 0$. A session key is derived from the master key, a secret n bits key piece, and a secret eight bits index j, with $0 \leq j < m - (n+1)$. Let us denote the master key as the sequence $b_0 b_1, \ldots, b_{m-1}$. Let the n-bit key piece be denoted as $c_0 c_1, \ldots, c_{n-1}$, $0 < n \leq m$. The resulting session key is the bit sequence $b_0, \ldots, b_{j-1} c_0 c_1, \ldots, c_{n-1} b_{j+n}, \ldots, b_{m-1}$. A session key established with this method is confidential to the subnetwork members and other nodes that have the master key of this network. The same is true for the scope of the traffic encrypted with the session key. A key piece value is randomly picked among the available 2^n binary values for every session key. An index is randomly picked among the $m-n$ possibilities for a total of $m2^n$ session keys available.

Venilia is an example where the second approach is used [10]. A time-dependent epoch value is combined with other items, namely a session key, an IV, and an index, as inputs to a hash function deriving block encryption keys.

For the third approach, two participants communicate through a channel. At each end of the channel, they make measurements that are assumed to be correlated. Luo et al.

propose a Received Signal Strength (RSS)-based key generation protocol for a pair of network participants [30]. The participants measure the RSS at their end of the channel linking them. The key generation comprises three steps. First, the RSS measurements are quantized into bits. They may not perfectly agree on these bits, due to the imperfect symmetry of the channel. Second, an information reconciliation protocol addresses imperfect symmetric RSS measurements to arrive at identical bit sequences at both ends. Third, a privacy amplification step hashes the bit sequence into a shorter secret resulting in a key with higher entropy. Luo et al. tested their idea in a sea trial across 556 meters. The approach has also been tested by Huang *et al.* [31] across similar distances. Pelekanakis *et al.* [32] developed a key establishment protocol of that type, measuring the channel's frequency response. They tested their idea in a lake. Adapting and evaluating these ideas across long underwater distances remain open issues.

IV. Secret Communications in UWSPR

We adapt the generic encryption scheme developed in the previous section for UWSPR. A generic usage model showing the placement of QPP, in the counter mode, in a middle-security layer is suggested. Then, we discuss a generic approach for encapsulating packets into the security layer protocol data units, consisting of encrypted data blocks. Specific design choices are made for UWSPR.

Fig. 1 shows a generic flow of data blocks suffixed with the counter value (i). In the sequel, this generic data model is applied to UWSPR.

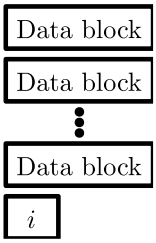

Fig. 1. Sequence of data blocks and counter suffix (i).

Fig. 2 shows the pictures the placement of QPP in the counter mode in a protocol architecture. The network layer provides the packet routing function. The link does the node-to-node transmission of frames. Using QPP in counter mode, the middle-security layer ensures the confidentiality of packets before their transmission.

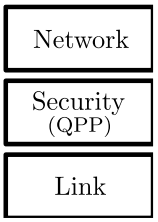

Fig. 2. Placement of a security layer in a protocol architecture.

Fig. 3 shows the encapsulation of a network packet into a security layer data unit consisting of two data blocks with a counter suffix (i). Afterward, the security layer data unit is encapsulated in a link layer frame.

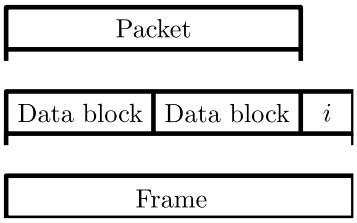

Fig. 3. Encapsulation of a network packet into a security layer data unit in a link-layer frame.

For UWSPR, everything must fit within a 50 bits frame. In a previous publication [33], we proposed and analyzed a solution with a block size (n) of 10 bits and 10 bits for the counter field, a frame containing four blocks. In this article, we propose an alternative, original solution. First, the QPP parameters suggested by He *et al.* [24] and Lou *et al.* [25] are used, which are four or eight for n, with the corresponding values eight and 64 for the number of permutations (d). Second, the counter suffix value is not sent explicitly, saving the few available bits, but rather derived from a common time reference.

A. Block Encryption

He *et al.* [25] and Lou *et al.* [26] proposed for the QPP parameters, the values of four or eight for the block size (n), with the corresponding values of eight or 64 for the number of permutations (d). Neither four nor eight is a divisor of 50, the frame size of UWSPR. Padding can be used to match a multiple of four or eight to the UWSPR frame size. This is, however, a loss of precious UWSPR frame bits. Padding is not an acceptable solution. Instead, we adapt the block size to the frame size using ciphertext stealing [17]. Ciphertext stealing solves the problem and does not require lengthening plaintext with padding.

Let a plaintext M consists of m words $\omega_0, \omega_1, \ldots, \omega_{m-2}, \omega_{m-1}$. Every word is an n bits block, except the last word ω_{m-1} of size k, which is longer than zero but shorter than n bits. All blocks are encrypted and transmitted normally, but the last two. The full block ω_{m-2} is encrypted into ciphertext γ_{m-2}. Let $\gamma_{m-2,n-k} \cdots \gamma_{m-2,n-1}$ be the rightmost k bits of γ_{m-2}. These bits are stolen and used as a suffix to block ω_{m-1} to create a new full block $\omega_{m-1}\gamma_{m-2,n-k} \cdots \gamma_{m-2,n-1}$. This block is encrypted into ciphertext γ_{m-1}. Ciphertexts $\gamma_{m-2,0} \cdots, \gamma_{m-2,n-(k+1)}$ and γ_{m-1} are sent to the destination. The destination decrypts γ_{m-1} into $\omega_{m-1}\gamma_{m-2,n-k} \cdots \gamma_{m-2,n-1}$. Then, it decrypts $\gamma_{m-2,0} \cdots \gamma_{m-2,n-(k+1)}\gamma_{m-2,n-k} \cdots \gamma_{m-2,n-1}$ into plaintext ω_{m-2}. The tail $\omega_{m-2}, \omega_{m-1}$ of M is recovered.

Let us now consider the specific UWSPR case with a four bits block size. Since the frame size is 50 bits, there are 13 blocks of plaintext $\omega_0, \omega_1, \ldots, \omega_{11}, \omega_{12}$. The last block ($\omega_{12}$) is short of two bits. The first 12 blocks $\omega_0, \omega_1, \ldots, \omega_{11}$ are normally encrypted into the ciphertext blocks $\gamma_0, \gamma_1, \ldots, \gamma_{11}$. The last block encryption is illustrated in Fig. 4. The rightmost two bits of the last block, $\gamma_{11,3-4}$ are stolen. The two-bit block ω_{12} is suffixed with the pair $\gamma_{11,3-4}$. The resulting four bits block is encrypted into γ_{12}. The following ciphertext is sent $\gamma_0, \gamma_1, \ldots, \gamma_{10}\gamma_{11,1-2}\gamma_{12}$, which is of size 50 bits and fits the

UWSPR frame size and where $\gamma_{11,1-2}$ denotes the leftmost two bits of the ciphertext block γ_{11}.

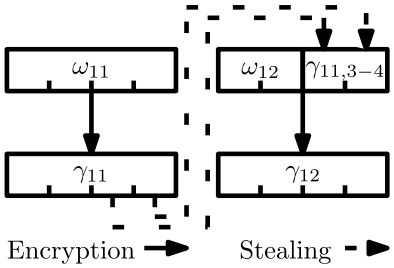

Fig. 4. Encryption of last two blocks of a 50 bits UWSPR frame.

Fig. 5 shows the last two blocks of the decryption of a 50 bits UWSPR frame. The first 11 blocks of ciphertext $\gamma_0, \gamma_1, \ldots, \gamma_{10}$ are normally decrypted into the 11 plaintext blocks $\omega_0, \omega_1, \ldots, \omega_{10}$. The last four bits γ_{12} of the ciphertext are decrypted first into the four bits $\omega_{12}, \gamma_{11,3-4}$. Next, the ciphertex $\gamma_{11,1-2}, \gamma_{11,3-4}$ is decrypted into the plaintext ω_{11}. The full plaintext $\omega_0, \omega_1, \ldots, \omega_{10}, \omega_{11}, \omega_{12}$ is recovered.

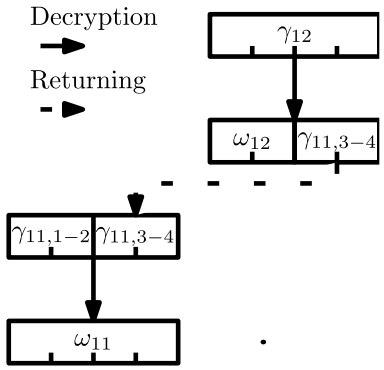

Fig. 5. Decryption of last two blocks of a 50 bits UWSPR frame.

B. Counter Value Deduction

The solution proposed in [33] contains 10 bits for the counter field. This is, again, a loss of precious data bits. We propose an original solution that does not require explicitly transmitting the counter values. They are rather derived by leveraging a common time reference. UWSPR supports six frame formats. They are distinct because their transmission duration is 1, 2, 4, 8, 16, or 20 minutes. Let us denote with the letter F the duration of frame transmission. The UWSPR transmissions are clocked. It is also assumed that the travel time of a frame is always shorter than its duration F. There is a reference start time t_0. Any transmission of a frame of duration F always starts at time $t_o + kF$, where k is equal to 0,1,2,3, ... When a frame is created and scheduled for transmission at time $t_o + kF$, the value of the counter used for encryption is k. When a frame arrives in the interval $t_o + kF$ and $t_o + (k+1)F - \epsilon$, where ϵ is a small value, the counter value used for decryption is k. This technique avoids sending the counter value explicitly and consuming any of the 50 bits of a UWSPR frame.

C. Risk of Collision

Let us examine the risk of collision, as stipulated in Definition 3, using the parameters recommended by He *et al.* [25] and Lou *et al.* [26]. With eight or 64 permutations for d, less than $2^{10} \cdot 2^3 \cdot 2$ or $2^{10} \cdot 2^6 \cdot 2$ bytes, i.e., 16 KB or 128 KB, are required to store the permutations. Figs. 6 and 7 plot the probability of collision for each case.[1] With a maximum number of encrypted blocks (i), $\sqrt{2^{mn}\, d^m}$, greater than 10^{10} in both cases, the collision risk is below 10^{-5} in both cases, which is very low. The UWSPR maximum one frame per minute rate means 10^8 hours of continuous operation with a single session key and low risk of collision, as Defined 3.

D. Integrity Protection

Encryption provides confidential communications. However, secure communications also require integrity protection, not achieved solely with encryption [31]. Encryption does not assure the absence of tampering while a packet travels from its origin to its destination. To achieve integrity protection, each message must embed a digital signature. Some bits among the 50 bits of a UWSPR frame must be used to store the digital signature. The signature can be generated using a one-way hashing function, taking in input the remaining available data bits of the frame.

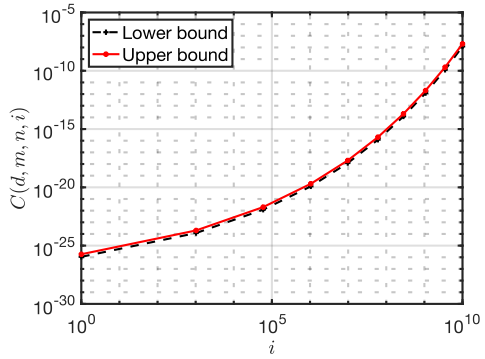

Fig. 6. Probability of collision; the block size (n) is four bits, the number of blocks is 13, and number of permutations (d) is 8.

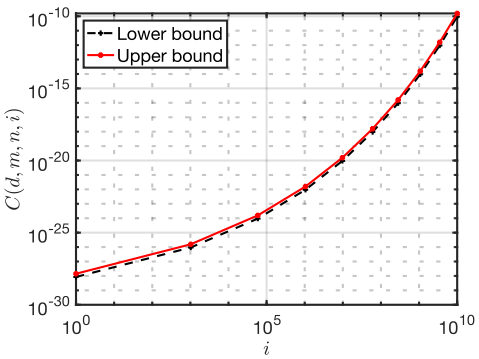

Fig. 7. Probability of collision; the block size (n) is eight bits, number of blocks is seven, and the number of permutations (d) is 64.

[1] The MATLAB Live Script used to produce Figs. 6 and 7 is available online: https://github.com/michelbarbeau/QPP-in-Pseudo-counter-Mode.

E. Comparison with Other Undewater Ciphers

QPP in CTRC or pseudo counter mode is compared with TUBcipher in Table I. QPP in CTRC mode achieves perfect indistinguishability for any block size n. QPP in pseudo counter mode does not achieve perfect indistinguishability due to the risk of collision. However, the analysis of Section IV.C has shown that this risk of collision is very low, even after the encryption of several messages. Finally, TUBcipher with a 27 bits block size does not achieve perfect indistinguishability, but a previous analysis has demonstrated that almost perfect entropy is obtained [28].

TABLE I. COMPARISON OF BLOCK CIPHERS

Scheme	Block size (bits)	Indistinguishability
QPP/CTRC	n	Perfect
QPP/Pseudo	Four or eight	No, but low collision probability
TUBcipher	27	No, but has near max. entropy

V. CONCLUSION

This article introduced a new symmetric-key encryption scheme for underwater communications and, using specific choices for parameters, a new security layer for the UWSPR long-distance underwater acoustic communication protocol. The encryption scheme builds upon QPP. It is combined with the block cipher counter mode to make it practical and adapted to small underwater protocol data units. QPP encryption in pseudo-counter mode has been created. The encryption and decryption algorithms share a secret sequence of random numbers, modulo d, and secret permutations $P_0, \dots, P_{d-1}$. The counter is used as an index over the permutations, selecting m permutations during encryption and selecting the same permutations for the decryption of a message. The value of the counter is sent together with the ciphertext. The indices of the secret permutations can also be generated from a shared number generator seed and a random number generator. QPP in pseudo-counter mode does not achieve perfect indistinguishability because of the collision risk. Nevertheless, we showed that the collision probability is very low and at most $\frac{i(i-1)}{2^{mn+1}d^m}$, where i be the number of messages of m blocks of size n bits encrypted using QPP in pseudo-counter mode with a sequence of m permutations chosen among d permutations in the symmetric group S_{2^n}.

We adapted the generic encryption scheme for UWSPR, designed for long-range underwater communications. Every frame contains 50 data bits. It means that everything must fit within 50 bits. We used four or eight for the block size n, with the corresponding values eight and 64 for the number of permutations (d). The counter suffix value is not sent explicitly, saving the few available bits. It is rather derived from a common time reference. The block size is adapted to the frame using ciphertext stealing. The message collision risk is below 10^{-5}, which is very low. The UWSPR maximum one frame per minute rate means that 10^8 hours of continuous operation is possible with a single session key and a low risk of collision. Encryption provides confidential communications. However, secure communications also require integrity protection. Further research is required to address this issue efficiently in small frames.

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] North Atlantic Treaty Organization (NATO), STANAG 47 48 - Digital Underwater Signalling Standard for Network Node Discovery and Interoperability, Edition 1, NSO/0413 UWWCG/4748, March 2017.

[2] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The JANUS underwater communications standard," in *Proc. 2014 Underwater Communications and Networking (UComms)*, pp. 1–4. IEEE, 2014.

[3] M. Barbeau, S. Blouin, and A. Traboulsi, "Adaptable design for long range underwater communications," *Wireless Networks*, vol. 4, pp. 1–7, July 2022.

[4] M. Jouhari, K. Ibrahimi, H. Tembine, and J. B. Othman, "Underwater wireless sensor networks: a survey on enabling technologies, localization protocols, and Internet of underwater things," *IEEE Access*, vol. 7, pp. 96879–96899, 2019.

[5] National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), [Online]. Available: https://csrc.nist.gov/publications/detail/fips/197/

[6] R. Kuang and M. Barbeau, "Quantum permutation pad for universal quantum-safe cryptography," *Quantum Information Processing*, pp. 1–22, 2022.

[7] M. Bellare and P. Rogaway, "Introduction to modern cryptography," *UCSD CSE*, vol. 207, 2005.

[8] A. Radosevic, R. Ahmed, T. M. Duman, J. G. Proakis, and M. Stojanovic, "Adaptive OFDM modulation for underwater acoustic communications: design considerations and experimental results," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 357–370, 2013.

[9] A. Song, M. Stojanovic, and M. Chitre, "Editorial underwater acoustic communications: Where we stand and what is next?," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 1–6, 2019.

[10] S. Holdcroft and A. Hobbs, "JANUS Class 17 venilia: Secure pre-canned messaging, report," *DSTL Cyber and Information Systems*, pp. 1–22, 2021.

[11] A. M. Hobbs and S. Holdcroft, "Tiny underwater block cipher (TUBcipher): 27-bit encryption scheme for JANUS class 17, report," *DSTL Cyber and Information Systems*, pp. 1–22, 2021.

[12] M. J. Dworkin, Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC, National Institute of Standards and Technology, Report number SP 800-38D, 2007.

[13] G. Ateniese, A. Capossele, P. Gjanci, C. Petrioli, and D. Spaccini, "SecFUN: Security framework for underwater acoustic sensor networks," *IEEE OCEANS*, Genova, pp. 1–9, 2015.

[14] A. Hamilton, J. Barnett, A. M. Hobbs *et al*., "Towards secure and interoperable underwater acoustic communications: current activities," *NATO IST-174 Research Task Group, Procedia Computer Science*, vol. 205, pp. 167–178, 2022.

[15] A. Caiti, V. Calabro, G. Dini, A. L. Duca, and A. Munafo, "Secure cooperation of autonomous mobile sensors using an underwater acoustic network," *Sensors*, vol. 12, no. 2, pp. 1967–1989, 2012.

[16] G. Dini and A. L. Duca, "A cryptographic suite for underwater cooperative applications," in *Proc. 2011 IEEE Symposium on Computers and Communications*, pp. 870–875, 2011.

[17] J. Daemen, "Cipher and hash function design strategies based on linear and differential cryptanalysis," Doctoral Dissertation, KU Leuven, 1995.

[18] C. Peng, X. Du, K. Li, and M. Li, "An ultra-lightweight encryption scheme in underwater acoustic networks," *Journal of Sensors, Hindawi Publishing Corporation*, pp. 1–10, 2016.

[19] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 212–219, 1996.
[20] National Institute of Standards and Technology (NIST), Report on post-quantum cryptography. (2016). [Online]. Available: http://dx.doi.org/10.6028/NIST.IR.8105
[21] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.
[22] M. Mosca and M. Piani. Quantum threat timeline report 2022. [Online]. Available: https://globalriskinstitute.org
[23] A. Mashatan and D. Heintzman, "The complex path to quantum resistance," *Communications of the ACM*, vol. 64, no. 9, pp. 46–53, 2021.
[24] D. Monroe, "Post-quantum cryptography," *Communications of the ACM*, vol. 66, no. 2, pp. 15–17, 2023.
[25] A. He, D. Lou, E. She, S. Guo, H. Watson, S. Weng, M. Perepechaenko, and R. Kuang, "FIPS compliant quantum secure communication using quantum permutation pad," *arXiv preprint arXiv:2301.00062*, 2022.
[26] D. Lou, A. He, M. Redding, M. Geitz, R. Toth, R. Döring, R. Carson, and R. Kuang, "Benchmark performance of digital QKD platform using quantum permutation pad," *IEEE Access*, vol. 10, pp. 107066–107076, 2022.
[27] R. A. Fisher and F. Yates, *Statistical Tables for Biological, Agricultural, and Medical Research*, USA: Hafner Publishing Company, 1953.
[28] Y. Beaupré, M. Barbeau, and S. Blouin, "Underwater confidential communications in JANUS," in *Proc.15th International Symposium on Foundations and Practice of Security (FPS)*, Ottawa, Canada, Dec. 2022.
[29] K. Hickman, "The SSL protocol. Tech. rep.," Netscape Communications Corp., Feb. 1995.
[30] Y. Luo, L. Pu, Z. Peng, and Z. Shi, "RSS-based secret key generation in underwater acoustic networks: advantages, challenges, and performance improvements," in *Proc. IEEE Communications Magazine*, vol. 54, no. 2, pp. 32–38, February 2016.
[31] Y. Huang, S. Zhou, Z. Shi, and L. Lai, "Channel frequency response-based secret key generation in underwater acoustic systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 5875–5888, Sept. 2016.
[32] K. Pelekanakis, C. M. Gussen, R. Petroccia, and J. Alves, "Towards physical layer cryptography for underwater acoustic networking," in *Proc. of the 5th International Underwater Acoustics Conference and Exhibition*, pp. 271–279, 2019.
[33] M. Barbeau, "Confidential underwater communications using quantum permutation PAD in counter mode," in *Proc. 12th International Conference on Communications, Circuits and Systems (ICCCAS)*, Singapore, May 5-7, 2023.