A NOVEL ERROR CORRECTION SCHEME IN QUANTUM KEY DISTRIBUTION (QKD) PROTOCOL

***Siao Ping Lee*[1], *Chee Kyun Ng*[1,2] *and Makhfudzah Mokhtar*[1]**

[1, 2, 3] Department of Computer and Communication Systems Engineering,
Faculty of Engineering, Universiti Putra Malaysia, Malaysia.
[2]Malaysian Research Institute on Ageing, Universiti Putra Malaysia, Malaysia.

Email: xquire_v@hotmail.com[1], mpnck@upm.edu.my[2], fudzah@upm.edu.my[3]

ABSTRACT

Ideally, in any quantum key distribution (QKD) communication system, each sifted key is expected to be received without error. However in practice, due to infeasibility of generating pure single photon and device impairment problem, some of the sifted key may experience errors. This results to the increment of quantum bit error rate (QBER) that requires error reconciliation for correcting error. The main concept in error reconciliation is very much related to the capability of correcting all errors while minimizing eavesdrop information. The quantum error correcting code such as Hamming code which used in Winnow protocol is found to be more attractive. However the Winnow protocol can only correct one error out of seven bits. Adopting this classical error correcting code, an improved reconciliation scheme namely Siao Ping 1985 (SP 1985) protocol is proposed in this paper to correct more errors in faster pace without additional formulation for the overall simplicity criterion. This research is aimed at building up efficiency and effectiveness of reconciliation of the robust BB84 protocol in coping with noise interference. The proposed SP 1985 reconciliation protocol utilizes a pair of forward and reverse order syndromes for error pattern recognition. It is carried out in a simple structure which can correct up to double erroneous bits and detect four erroneous bits for each seven bits. Therefore, it is sufficient to deliver the desirable outcome after investigating its capability by correcting two errors out of seven bits compared to Winnow protocol. Its effectiveness can be measured based on simulation result which leads to reducing the QBER.

Keywords: Hamming code, error correction, QKD, reconciliation protocol.

1.0 INTRODUCTION

Confidentiality, integrity, authentication and non-repudiation are four significant criteria specified to ensure secrecy in communication between legitimate parties nowadays [1 - 6]. In order to achieve these requirements, several measures based on cryptography have long since been practiced. Previously confidentiality is secured via encryption by transforming ordinary message into scrambled text prior to dissemination, thus concealing information in the message. Meanwhile, one-way cryptographic hash function was exploited to corroborate the correctness of a received message without undue amendment in transit, thus contributing towards the verification of message integrity. Surpassing the hash function, the utilization of message authentication code allows authentication to justify user identity besides safeguarding message's integrity. Lastly, digital signature is utilized to address non-repudiation, deterring refutability of forgeable actions like financial transaction consummated via electronic payment [1].

The Advanced Encryption Standard (AES) [7, 8] is a symmetric-key cryptography which has been adopted worldwide today to protect classified information. An algorithm described by Ronald Rivest, Adi Shamir and Leonard Adleman or commonly known as RSA [9] is the pioneer in brand-new asymmetric-key cryptography, used mainly to consolidate key-agreement protocol. Together with lengthy secret key, the former utilizes substitution-permutation network which creates confusion and diffusion for secure cryptography, whereas the later makes use of infeasibility to factor large numbers in retrieving key via classical computer for the same purpose [10].

The cryptography algorithms are mostly designed to comply with computational hardness assumption and not withstanding imminent threat imposed by computationally efficient device. Permitting superposition of binary states, a quantum computer which executes operation on quantum bits or qubits is believed to be capable of speeding up computations tremendously once the associated technologies are in place. Thus, it renders the compromised asymmetric-key cryptography and endangers computationally secure symmetric-key cryptography [1]. Therefore, a classic technique of symmetric-key cryptography known as one-time pad (OTP) [11] is regarded as the ultimate solution, as it has been proven to be information theoretically impregnable against cryptanalysis, if a perfectly random secret key of infinite length is employed only once and never reused [12, 13]. Due to lack of practical implementation, it was not much attended until now.

As unguarded delivery of secret key may jeopardize the probable scheme, the quantum key distribution (QKD) which escorts the secret key through quantum channel using quantum state encoding such as photon polarization, is suggested to facilitate OTP in order to set up a secure communication for the secret key sharing [13 - 18]. Having its security ascertained by Heisenberg uncertainty principle [19] and no-cloning theorem of quantum mechanics, QKD guarantees delivery of the secret key in such a way that possible eavesdropping can be detected during error rate estimation [20]. The renowned QKD protocol, which has been proven unconditionally secure against any eavesdropping, was built upon inspiration from quantum realization of unforgeable bank notes [21] and promulgated by developers of Charles Bennett and Gilles Brassard in 1984 or typically known as Bennett-Brassard 1984 (BB84) protocol [22].

In fact, the joint venture between OTP and QKD is consistent with Kerckhoffs's principle which enunciates that key’s secrecy should be the one and only pivot leveraging security of a cryptosystem [23]. However, errors attributed to imperfections in the physical implementation are prevalent, with or without eavesdropping. Consequently, reconciliation is vital for secret key distillation, which serves as prerequisite for information-theoretically secure cryptography. Reconciliation is carried out in the authenticated classical channel to correct undesired errors such that the discrepancies between sender’s and receiver's secret key can be fixed for successful encryption and decryption respectively. It can be accomplished by employing either simple classical error correcting code or advanced quantum error correcting code.

Winnow protocol has been introduced in [24] to reduce the disclosure of partial information to eavesdropper by taking advantage of both parity bit and Hamming code for single-bit error correction. However, the need of several iterations is necessary because Winnow protocol tends to correct a block of sifted secret key that is interspersed with three or more odd multiple bits of error inaccurately while abandoning detection of even multiple bits of error. Moreover, the Hamming code which used in Winnow protocol is found can only correct one error out of seven bits. There is a very strong motivation to develop a reconciliation protocol that can minimize public communication between legitimate communicants with improved error correcting capability.

Thus, this paper aims to enhance reliability of QKD by proposing an efficient and effective reconciliation protocol that rectifies errors in single pass with the improved error correcting capability into BB84 protocol. A modified Hamming code is developed to improve BB84 protocol by correcting two errors out of seven bits which leads to QBER reduction. Thus, the syndrome of conventional Hamming code has been redefined to increase error detecting and correcting capability, which improves overall reconciliation process without iteration in order to reduce interactivity indirectly. This design utilizes a pair of forward and reverse order syndromes for the error pattern recognition. Thus, a new reconciliation protocol namely Siao Ping 1985 (SP1985) will be introduced in BB84 protocol to cater this modification. This new reconciliation protocol is developed and evaluated in terms of amount of disclosed bit and quantum bit error rate (QBER).

The rest of the paper is organized as follows. The rest of the paper is organized as follows. Section 2 introduces some preliminaries on QKD and its previous related protocols. The details of the proposed SP 1985 reconciliation protocol which adopting the Hamming code is described in Section 3. The uniqueness of proposed SP 1985 protocol and its performance evaluations compared with other protocols are presented in Sections 4 and 5 respectively. Finally, the paper is concluded in Section 6.

2.0 QUANTUM KEY DISTRIBUTION (QKD)

The QKD, which is the best known application of quantum cryptography, is not appointed for informative data encryption as modern cryptography but it committed to ensure secrecy of delivered key. In turn, its secret key can be used as OTP or assigned for other techniques of symmetric-key cryptography as standard cryptographic key [13 - 18]. The basic structure of a QKD system generally comprises both quantum and classical channels as shown in Fig. 1.

The QKD protocol is begun with a sender or Alicetransmits a stream of random key encoded in conjugative quantum states, photonpolarization in typical, to a legitimate receiver or Bob overquantum channel. Upon reception of photon states, Bob applies his own randommeasurement to find out their respective polarization. Then, both Alice and Bob determine the correlation between the transferred key and measured key whichare conducted throughthe authenticated classical channel without revealing actualinformation of the key. Finally, an identical string of sifted key is deduced at bothends discreetly. Stipulated by the Heisenberg uncertainty principle and no-cloning theorem, eavesdropping is detectable because the measurement performed by eavesdropper or Eve will perturb some photon states unavoidably. If error rate exceeds beyond acceptable error threshold after the legitimate parties crosschecking afraction of their sifted keys, eavesdropping occurred and the established sifted key isdiscarded without losing any valuable information [13 - 18].

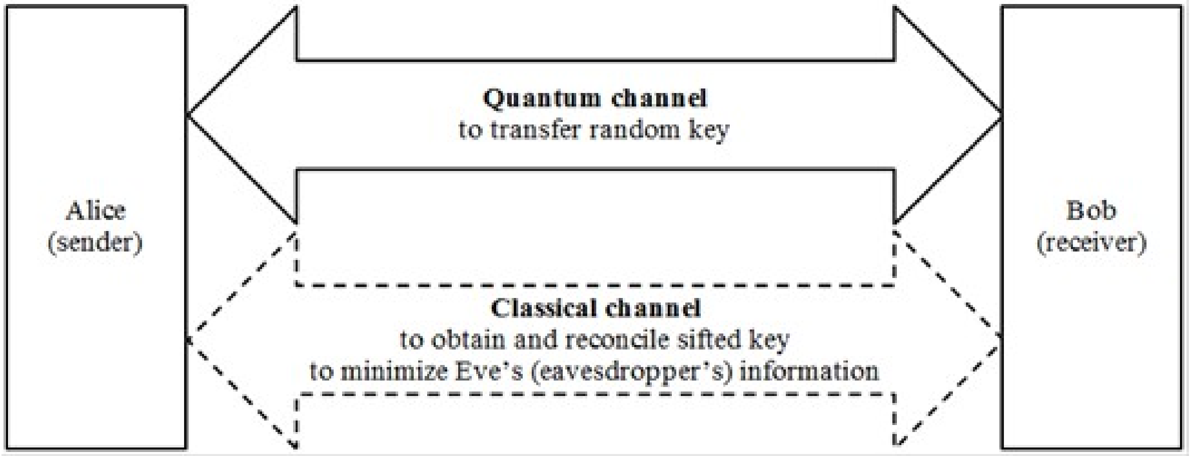

Alice
(sender)
Quantum channel
to transfer random key
Bob
(receiver)
Classical channel
to obtain and reconcile sifted key
to minimize Eve's (eavesdropper's) information

Fig. 1: Basic structure of QKD system.

There are a lot of protocols have been devised in QKD research such as BB84, Ekert 1991 (E91) [25], Bennett 1992 (B92) [26], Bennett-Brassard-Mermin 1992 (BBM92) [27], six-state [28], three-state [29] and Scarani-Acín-Ribordy-Gisin 2004 (SARG04) [30] protocols. The B92, six-state, three-state and SARG04 protocols are variations of BB84 protocol based on prepare-and-measure strategy, whereas E91 and BBM92 protocols are variations of BB84 protocol based on quantum entanglement means. BB84 protocol is adopted in this research due to its unconditional security [31 - 33], sustainability and simplicity.

2.1 BB84 Protocol

In the BB84 protocol as shown in Fig. 2, Alice sends a stream of random key through quantum channel to Bob after recording photon state of each key element. The key is firstly coded in bits then further encoded in conjugative quantum states, constituted by rectilinear and diagonal polarization of photon conventionally. The mapping of bit to respective polarization is indicated at the bottom of Fig. 2. Bob acknowledges his receipt of photons and measures them using a stream of random rectilinear and diagonal bases, which are independent from those of Alice. Whenever the photon state is a subset of basis of measurement, he gets the correlated result. The corresponding measurement results, known as raw key, are recorded. After transference of the random key, Bob informs Alice about the stream of basis being used for measurement through the authenticated classical channel, which is accessible to the passive eavesdropping. Alice notifies Bob which of his measurement is compatible with the delivered photons, and the photon state should have been detected correctly, enabling them to disregard the result that susceptible to disruptive measurement. After discarding anomalies in respective raw key, they deduce identical sifted keys in secret, which can be used for cryptographic purpose. Obviously, their secret key is not predetermined but is developed in conjunction of their random choices, with an aid of guided investigation [22].

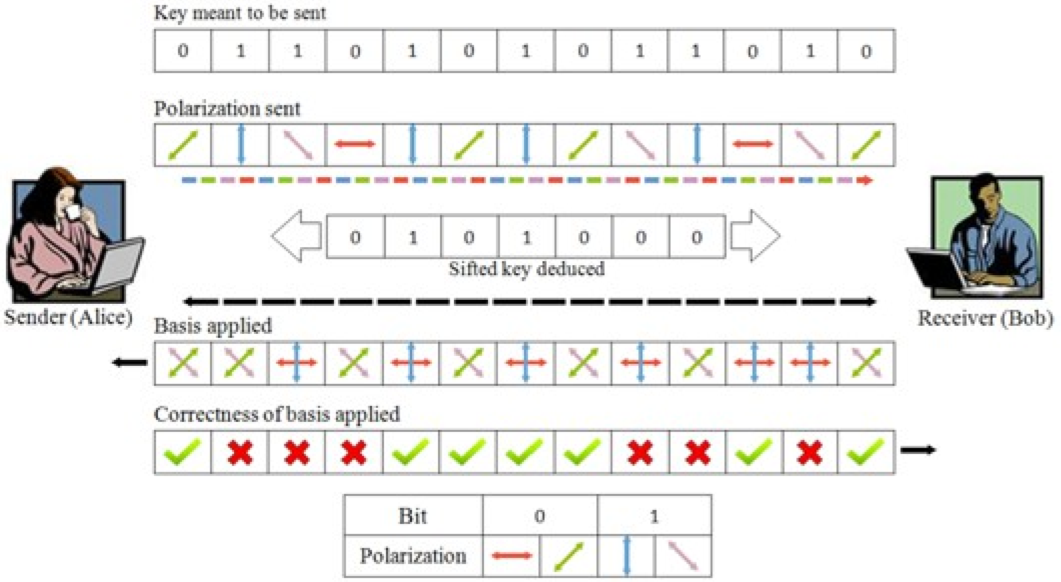


Fig. 2: Schematics of the BB84 protocol for ideal case.

Resultantly, Bob's sifted key suffers from 25% of QBER in respect to Alice's sifted key, where QBER is a yardstick measuring deviation of the sifted keys pair in bits [13, 34]. Hence, after deducting the sifted key in QKD, a portion of the key is sacrificed by both parties. If the inconsistencies are beyond tolerance, they reasonably abandon the sifted key to foil Eve's intervention in retrospect. Otherwise, another portion of sifted key that is not disclosed for public comparison, denoted as working secret key in ideal case will be used [3, 13, 22, 34, 35]. Thus, after the sifting process, reconciliation is vital to ascertain identicalness of the sifted keys pair.

2.2 Reconciliation in QKD

A common practice is to have the persisting errors corrected if QBER is tolerable, using the authenticated classical channel shown in Fig. 1 [13, 36]. Since the cause of errors is intangible between technical impairment and malicious eavesdropping, the latter is assumed to be the origin of disturbance as a conservative measure [34].

Some proofs of QKD's security have been presented to showcase the corresponding noise resistant threshold [34, 37]. In the earliest attempts, BB84 protocol was proven secure against all attacks permissible by laws of quantum mechanics whenever the QBER is less than 7.4% [33] and up to 7.56% [38] in two independent research studies. The security of entanglement purification based E91 protocol was proven as well [39]. Unifying [33] and [39] typical proofs with small changes, BB84 protocol using one-way classical post-processing was shown secure as long as QBER is less than 11% [16, 40]. During that meantime, the bound was elevated to 18.9% if two-way classical post-processing is admissible [41].

Once reconciliation is initiated, the error detection and correction make the concerted effort to mitigate inconsistencies in the sifted keys pair by using interactive or non-interactive protocol. An interactive reconciliation protocol requires repetitive exchange of parity bit between Alice and Bob via a two-way communication channel to detect and correct errors. On the contrary, a non-interactive reconciliation protocol applies concept of one-way source coding with side information to eliminate the interactivity between Alice and Bob when performing error correction [13], as shown in Fig. 3.

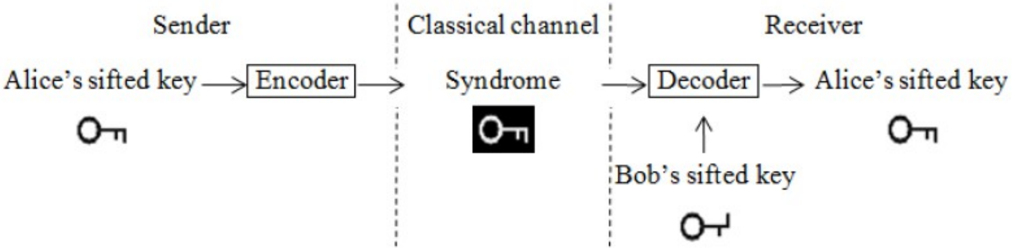


Fig. 3: Source coding with side information in reconciliation.

In a conceptual manner, Alice's sifted key is first encoded into respective syndrome. The syndrome is then transmitted over the authenticated classical channel to Bob, and fed into a decoder together with his own sifted key to restore Alice's sifted key with high probability. In this way, the sifted key with flaw at receiving end is mended allegedly [13]. The non-interactive reconciliation protocol is a preferable technique since it can catalyze efficiency of error correction and minimize public communication concurrently.

2.3 BBBSS Protocol

BBBSS protocol, is a novel interactive reconciliation protocol designed for reconciliation in QKD. In this protocol, the position of bits in Alice's and Bob's string of sifted key is randomly permuted according to an agreed-upon arrangement such that the errors are more uniformly redistributed. The rearranged strings of sifted key are partitioned into blocks, hoping that each block is interjected only with one bit of error after the shuffle. The parity bit for each block is compared between Alice and Bob correspondingly. The blocks of parity matching are considered as errorless tentatively whereas those of diverging parity are inferred to be interfered with any odd number of erroneous bits. The last bit of each block is discarded to counteract partial information gained by Eve.

Whenever the latter scenario occurs, binary search is initiated, in such a way that the erroneous block is divided into two sub-blocks of almost equal size and then parity bit of either sub-block is compared between Alice and Bob correspondingly. The binary search in that sub-block is carried on if conflicting parity recurs. Otherwise, the bisection is shifted to the other sub-block, which is assumingly interjected with at least one bit of error. The binary search ends at any time when an erroneous bit is located and corrected by Bob. The last bit of each sub-block is discarded to counteract partial information gained by Eve. If no error is located, the erroneous bit was coincidentally removed as last bit.

As even multiple bits of error in other erroneous blocks still stay undetected after performing above-mentioned steps, the random permutation and parity check are repeated for several iterations using remaining bits, with arbitrarily chosen subset as block that has a growing size. Twenty consecutive agreements of parity are recommended to assure that remaining errors are approximately negligible. Thus, BBBSS protocol locates and corrects errors using exhaustive binary search through numerous exchange of parity bit between Alice and Bob for several iterations. For an arbitrary block of sifted key in bits, it detects an odd number of erroneous bits and corrects one bit of error in single pass but cannot get rid of an even number of erroneous bits in principle [26].

2.3.1 BBBSS Protocol Applying Cyclic Redundancy Check (CRC)

Traditional BBBSS protocol leaves even multiple bits of error unmanaged within iteration. In order to make those errors traceable, the exchanged parity bit between Alice and Bob for consistency verification is recently proposed to be replaced by check value of CRC such as a generalization form of even parity bit during reconciliation. The generator polynomial can be specifically designed to detect all odd and even numbers of erroneous bits but restricted by operation of BBBSS protocol, in which only one bit of error can be eliminated in a block of sifted key in single pass. As a consequence, several iterations are still required to complete the reconciliation in general. Additionally, there is a difficulty to synchronize the generator polynomial between Alice and Bob beforehand. After all, it was shown via simulation result that the dwindling of QBER is expedited for iterations involved in BBBSS protocol after integrating CRC, in reference to application of parity bit [42]. Speculatively, this approach also minimizes interactivity involved in the protocol since the detection of an even number of erroneous bits is made possible.

2.3.2 BBBSS Protocol Applying Cryptographic Hash Function

Another contemporary proposed modification is to have the parity bit in BBBSS protocol substituted by message digest of cryptographic hash function such as Message-digest Algorithm Number 5 (MD5). The message digest can be used in reconciliation to detect any number of errors for the reason that generating the same message digest from a pair of deviating sifted keys via a cryptographic hash function is infeasible. Its procedure, challenge and performance as well as interactivity involved are similar to those of replacing parity bit with CRC in BBBSS protocol [43].

2.4 Winnow Protocol

At the beginning of Winnow protocol, after shuffling the bits of sifted keys pair in the same way, Alice's and Bob's string of sifted key are also divided into blocks and then subjected to parity check correspondingly. One bit in each block is then discarded because of the parity check. After that, non-interactive reconciliation begins. First of all, syndrome is calculated and sent from Alice to Bob, for each of the blocks exhibiting odd result in preliminary test. It is noted that syndrome is primitively an indicator implying correctness of a received codeword during error detection, but here is where it fits into reconciliation. At receiving end, syndrome measurement is carried out by Bob using received syndrome in tandem with his own sifted key's syndrome to compute difference between their syndromes, and determine associated correctable error pattern of his sifted key such that the most probable error can then be corrected by him independently. Normally, the assigned error correcting code is Hamming code, the first effective linear block code invented to be able to correct one bit of error in a valid codeword.

Confined by Hamming code's limited error correcting capability, this method will have a block of sifted key deduced by Bob that is interspersed with three or more odd multiple bits of error incompletely corrected such as only one of the erroneous bits is corrected, not corrected or worse yet, wrongly corrected, causing an extra erroneous bit. Furthermore, this method cannot detect even multiple bits of error, leaving them uncorrected. Hence, iterations that independent of each other are still a must during reconciliation. Remaining bits of sifted key in each block that equivalent to redundancy bits of a Hamming code's codeword, are also discarded before commencement of new round of reconciliation. Some erroneous bits that fall among the removed bits are thus discarded without undergoing error correction [24, 36].

Thus, Winnow protocol applies parity bit to detect error just the same as aforesaid BBBSS protocol, but Winnow protocol manages to reduce the interactivity between Alice and Bob further by applying Hamming code to correct one bit of error in an erroneous block having size of codeword length. The idea can be a double-edged sword since reconciliation will be performed mistakenly whenever the block is interspersed with three or more odd multiple bits of error. In addition, it skips detection of even multiple bits of error in each iteration. These withhold it in terms of efficiency [24, 36, 44]. For optimum performance, research has shown that block size of eight bits will minimize number of iterations required and amount of bit exposed [36].

2.5 Winnow Protocol Applying Convolutional Code

Blocks containing three or more odd multiple bits of error are deteriorated while those with even multiple bits of error stay undetected within iteration in Winnow protocol. To resolve this predicament, Hamming code is proposed to be replaced by convolutional code such that any odd number of erroneous bits in a block of optimal size can be corrected without introducing additional errors. The main difference of this proposition in comparison with original approach is the way Bob calculates an estimate of Alice's sifted key. In this case, the Viterbi algorithm which provides maximum likelihood performance, is used in decoder. In comparison with original approach, this method suppresses QBER substantially in less iteration, at the expense of rising disclosed bit and operational complexity [45].

3.0 THE PROPOSED SP 1985 RECONCILIATION SYSTEM ARCHITECTURE

The aim of this research is to develop a reconciliation protocol that can minimize public interactivity between legitimate communicants, in terms of effectiveness of reconciliation with the improved error correcting capability into BB84 protocol. The Winnow protocol, which is the Hamming code driven non-interactive reconciliation protocol, is adopted as the blueprint for the development of proposed SP 1985 reconciliation protocol. Thus, the conventional interpretation of Hamming code's syndrome has to be modified to increase error detecting and correcting capability, allowing all-rounded reconciliation without iteration to reduce interactivity indirectly. Complementarily, a new algorithm for reconciliation is needed in Winnow protocol to cater this modification. The typical flow of Winnow protocol will be simplified to decrease the interactivity even more.

The core constituent of Winnow protocol is Hamming code. For any integer m that is greater than or equal to three, a Hamming code (n, k, dmin) exists with the following parameters:

Codeword length in bits, $n = 2^m - 1$ (1)

Number of message bits, $k = 2^m - m - 1$ (2)

Number of redundancy bits, $m = n - k$ (3)

Minimum distance in bits, $d_{\min} = 3$ (4)

Error correcting capability in bit, $t = 1$ (5)

Parity-check matrix *H* and generator matrix *G* for a systematic Hamming code can be formatted as

$$H = \left[I_m \, P \right] \tag{6}$$

$$G = \left[P^T I_k \right] \tag{7}$$

where I_m is a *m*-by-*m* identity matrix, likewise for I_k while *P* is a *m*-by-*k* sub matrix comprises non-repeating column vectors of weight, such as number of non-zero elements, ranging from two up to *m*, and P^T is the transpose of *P*. The Hamming code (7, 4, 3), which is of the simplest form among its family, is used in this research, employing specific *H* and *G* in the formation such as

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{8}$$

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Since the $d_{\min}$ of a linear block code, such as the minimum number of bits that differ between any pair of valid codewords of that code, is equal to the smallest number of columns in *H* that sum to zero, Hamming code (7, 4, 3) has a $d_{\min}$ of three and is capable to correct one bit of error in the code words, abiding by

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \tag{10}$$

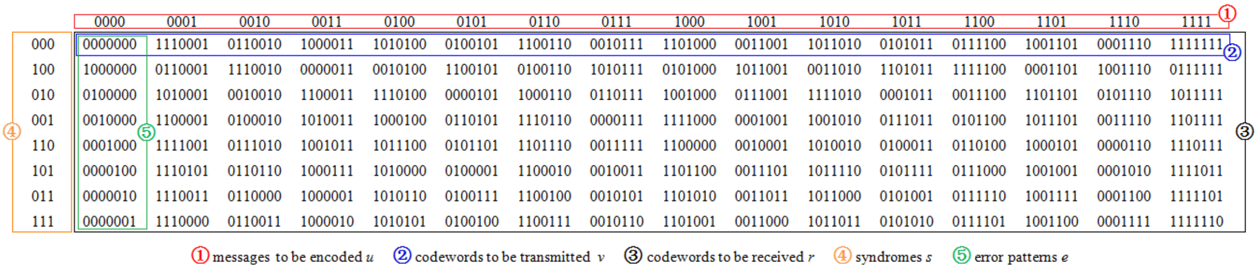
For linear block code, the encoded u message, corresponding transmitted v codeword, possible received r codeword, associated correctable error pattern (formally known as coset leader) and respective s syndrome are related as

$$v = u \cdot G \tag{11}$$

$$r = v \oplus e \tag{12}$$

$$s = r \cdot H^T \tag{13}$$

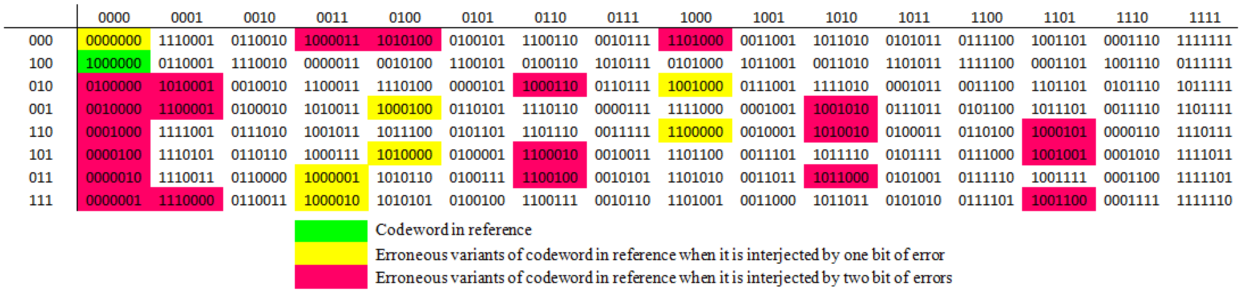
Being one of the perfect codes, all elements of above-named Hamming code (7, 4, 3) can be organised into a standard array in which exactly all the correctable error patterns lead the cosets, as shown in Fig. 4. The circumstanced erroneous codeword can then be corrected by looking up the standard array [46]. Fig. 5 diagrammatizes interpretation of erroneous codeword in context of QKD.



	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	0000000	1110001	0110010	1000011	1010100	0100101	1100110	0010111	1101000	0011001	1011010	0101011	0111100	1001101	0001110	1111111
100	1000000	0110001	1110010	0000011	0010100	1100101	0100110	1010111	0101000	1011001	0011010	1101011	1111100	0001101	1001110	0111111
010	0100000	1010001	0010010	1100011	1110100	0000101	1000110	0110111	1001000	0111001	1111010	0001011	0011100	1101101	0101110	1011111
001	0010000	1100001	0100010	1010011	1000100	0110101	1110110	0000111	1111000	0001001	1001010	0111011	0101100	1011101	0011110	1101111
110	0001000	1111001	0111010	1001011	1011100	0101101	1101110	0011111	1100000	0010001	1010010	0100011	0110100	1000101	0000110	1110111
101	0000100	1110101	0110110	1000111	1010000	0100001	1100010	0010011	1101100	0011101	1011110	0101111	0111000	1001001	0001010	1111011
011	0000010	1110011	0110000	1000001	1010110	0100111	1100100	0010101	1101010	0011011	1011000	0101001	0111110	1001111	0001100	1111101
111	0000001	1110000	0110011	1000010	1010101	0100100	1100111	0010110	1101001	0011000	1011011	0101010	0111101	1001100	0001111	1111110

① messages to be encoded u ② codewords to be transmitted v ③ codewords to be received r ④ syndromes s ⑤ error patterns e

Fig. 4: Standard array of a Hamming code(7, 4, 3) with attached syndromes.



	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	🟨 0000000	1110001	0110010	🟥 1000011	🟥 1010100	0100101	1100110	0010111	🟥 1101000	0011001	1011010	0101011	0111100	1001101	0001110	1111111
100	🟩 1000000	0110001	1110010	0000011	0010100	1100101	0100110	1010111	0101000	1011001	0011010	1101011	1111100	0001101	1001110	0111111
010	🟥 0100000	🟥 1010001	0010010	1100011	1110100	0000101	🟥 1000110	0110111	🟨 1001000	0111001	1111010	0001011	0011100	1101101	0101110	1011111
001	🟥 0010000	🟥 1100001	0100010	1010011	🟨 1000100	0110101	1110110	0000111	1111000	0001001	🟥 1001010	0111011	0101100	1011101	0011110	1101111
110	🟥 0001000	1111001	0111010	1001011	1011100	0101101	1101110	0011111	🟨 1100000	0010001	🟥 1010010	0100011	0110100	🟥 1000101	0000110	1110111
101	🟥 0000100	1110101	0110110	1000111	🟨 1010000	0100001	🟥 1100010	0010011	1101100	0011101	1011110	0101111	0111000	🟥 1001001	0001010	1111011
011	🟥 0000010	1110011	0110000	🟨 1000001	1010110	0100111	🟥 1100100	0010101	1101010	0011011	🟥 1011000	0101001	0111110	1001111	0001100	1111101
111	🟥 0000001	🟥 1110000	0110011	🟨 1000010	1010101	0100100	1100111	0010110	1101001	0011000	1011011	0101010	0111101	🟥 1001100	0001111	1111110

🟩 Codeword in reference

🟨 Erroneous variants of codeword in reference when it is interjected by one bit of error

🟥 Erroneous variants of codeword in reference when it is interjected by two bit of errors

Fig. 5: Erroneous variants of a codeword in context of QKD.

In most scenarios, message bits are encoded into correctable codewords with additional redundancy bits prior to transmission. For a Hamming code (n, k, $d_{\min}$), there is a total of 2^k correctable codewords, each having n possible erroneous variants whenever they are interjected with one bit of error during transmission. At receiving end, error free codewords will give rise to all-zero syndromes which indicates no error detected, whereas circumstanced erroneous codewords will result in non-zero syndromes which theoretically permits up to double-bit error detection, whichapplying (13) during syndrome measurement. Single-bit error correction can then be performed via syndrome decoding by adding each erroneous codeword with corresponding error pattern associated with the non-zero syndrome bitwise using binary XOR operation [46].

In QKD, the bits of sifted keys pair deduced by Alice and Bob are not treated as message bits which are regularly encapsulated into correctable codewords for reconciliation. This is to ensure that information of the sifted key will not be exposed once the associated H or G is synchronized between them publicly. Instead, all the possible received codewords encompassing both error free and erroneous codewords in standard array are used to represent the bits of sifted keys [10]. From Fig. 5, when an chosen arbitrarily codeword is interjected with one or two bits of error, all possible erroneous variants of the chosen codeword are distributed to other cosets which having different syndrome from the one belongs to the chosen codeword. A posteriori, result of all elements in the same cosetis observed to constraint of $d_{\min}$ of three. In other words, each codeword that associated with the same syndrome has minimum of three dissimilar bits. This characteristic is used as a hint in hypothesizing that the maximum occurrence of two bits of error in any codeword can be detected, and hence differentiated from codeword in reference which is regarded as errorless. It gets corrected with associated error pattern, which has been determined via expedient syndrome measurement. Hence, one or two bits of error interjected into a segment of sifted key, which wrongly deduced by Bob, can be corrected using the same way. This hypothesis is distinct from classical theory which formulates the Hamming code is eligible merely for single-bit error correction after performing syndrome measurement.

In Winnow protocol, for an erroneous 7-bit block of sifted key that shows diverging parity during preliminary test, the difference between syndrome of the block of sifted key deduced by Alice and the one deduced by Bob, when computed by Bob during syndrome measurement to determine the associated correctable error pattern before the most probable single-bit error correction can be performed by Bob [24, 36]. The reconciliation, which is complemented by privacy maintenance, is shown in Fig. 6.

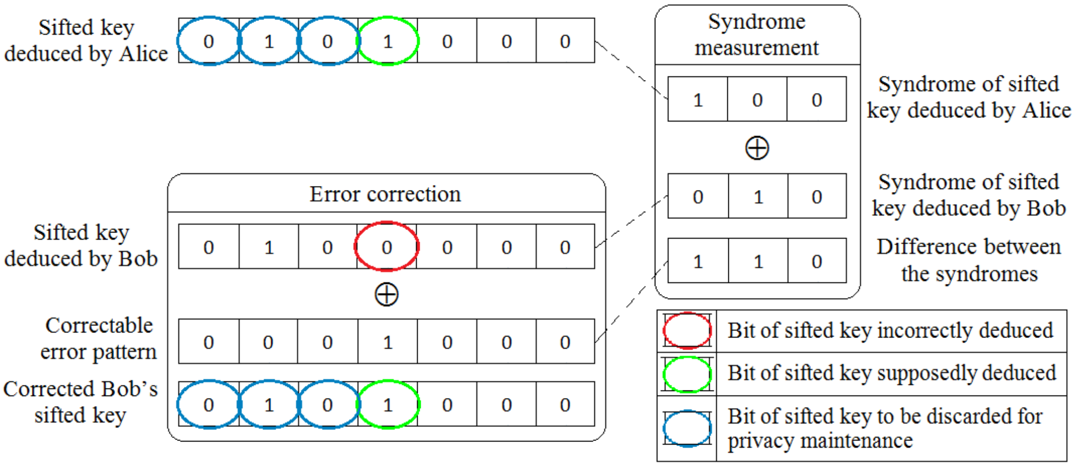
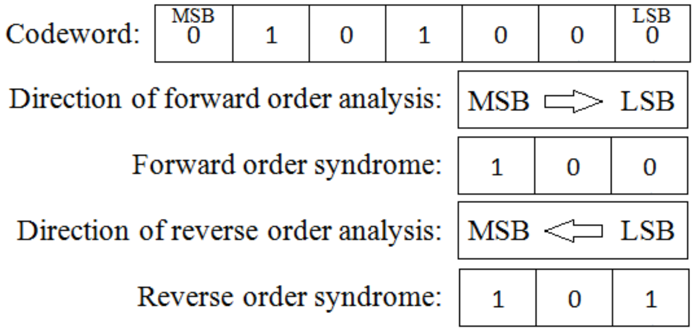

Sifted key deduced by Alice
0 1 0 1 0 0 0
Syndrome measurement
1 0 0
Syndrome of sifted key deduced by Alice
⊕
Error correction
0 1 0
Syndrome of sifted key deduced by Bob
Sifted key deduced by Bob
0 1 0 0 0 0 0
1 1 0
Difference between the syndromes
⊕
Correctable error pattern
0 0 0 1 0 0 0
Corrected Bob's sifted key
0 1 0 1 0 0 0
Bit of sifted key incorrectly deduced
Bit of sifted key supposedly deduced
Bit of sifted key to be discarded for privacy maintenance

Fig. 6: Reconciliation and privacy maintenance using Winnow protocol.

Hence, in the proposed SP 1985 reconciliation protocol, in order to detect any Hamming (7, 4, 3) codeword that is interjected with up to two bits of error, codewords with weight of two in every coset of the standard array are collectively gathered as extra correctable error patterns associated with respective syndrome. Resultantly, there is a mix of single-bit and double-bit error patterns associated with each non-zero syndrome. Without introducing additional parameter which may be favourable for possible eavesdropping, the syndrome measurement is done twice in slightly distinctive manner for an attempt to reconcile possible errors in the codeword such that two set of error patterns in respect to two set of syndromes are made available for matching analysis. Thus, a simple concept of logical reasoning is featured by analyzing the codeword in forward and reverse orders. It is utilizing an idea that the exact error pattern should remain the same regardless of the direction in which analysis is performed, such that whether from the most significant bit (MSB) towards the least significant bit (LSB) or vice versa as shown in Fig. 7.



Codeword: MSB 0 | 1 | 0 | 1 | 0 | 0 | LSB 0

Direction of forward order analysis: MSB ⇨ LSB

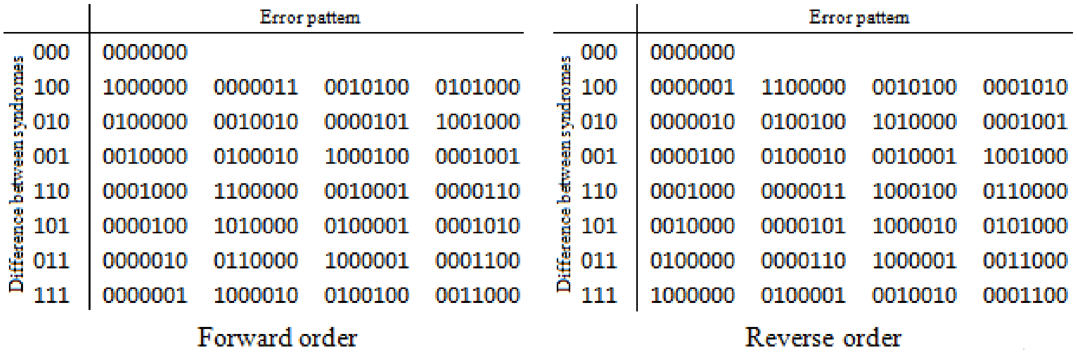
Forward order syndrome: 1 | 0 | 0

Direction of reverse order analysis: MSB ⇦ LSB

Reverse order syndrome: 1 | 0 | 1

Fig. 7: The order of analysis with respective syndrome in SP 1985 reconciliation protocol.

In QKD application, syndrome in forward order is the syndrome calculated when a block of sifted key is analyzed in forward order (MSB ➔ LSB), while syndrome in reverse order is the syndrome calculated when a block of sifted key is analyzed in reverse order (LSB ➔ MSB). Indeed, syndrome in forward order is the syndrome that has been used in Winnow protocol. The difference between a block syndrome of sifted key deduced by Alice and the one deduced by Bob in forward order as well as reverse order, are correspondingly computed by Bob to determine the associated error patterns in both orders as shown in Fig. 8.



Difference between syndromes	Error pattern			
000	0000000			
100	1000000	0000011	0010100	0101000
010	0100000	0010010	0000101	1001000
001	0010000	0100010	1000100	0001001
110	0001000	1100000	0010001	0000110
101	0000100	1010000	0100001	0001010
011	0000010	0110000	1000001	0001100
111	0000001	1000010	0100100	0011000

Forward order

Difference between syndromes	Error pattern			
000	0000000			
100	0000001	1100000	0010100	0001010
010	0000010	0100100	1010000	0001001
001	0000100	0100010	0010001	1001000
110	0001000	0000011	1000100	0110000
101	0010000	0000101	1000010	0101000
011	0100000	0000110	1000001	0011000
111	1000000	0100001	0010010	0001100

Reverse order

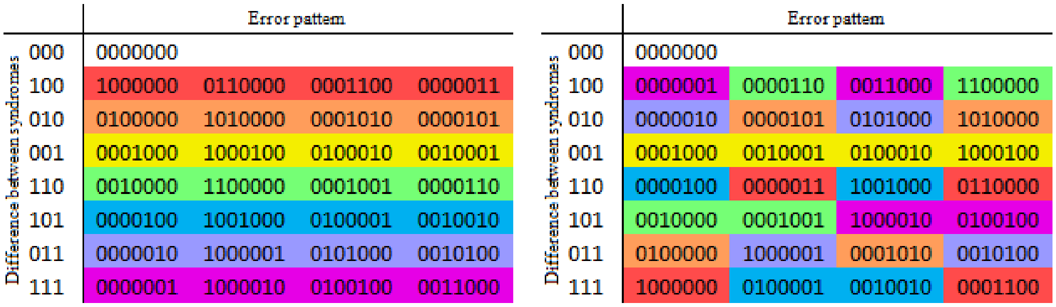
Fig. 8: The error patterns associated with difference between syndromes in respective orders.

It can be seen that error patterns associated with non-zero syndrome in forward order are a collection of codewords with weight of one or two in every coset of the standard array during preparatory stage, while error patterns associated with non-zero syndrome in reverse order are those of forward order but experienced straight left right flipping. Such adjustment is made such that posterior matching analysis and error correction can be performed by Bob in reference to conventional forward order. Whenever syndrome measurement does not result in all-zero syndromes in forward order and that of in reverse order, the maximum occurrence of two bits of error in a block of sifted key is detected. Otherwise the differences are all-zero syndromes, which intimating the block of sifted key is errorless. The matching analysis is then carried out to determine the identical error pattern associated with difference between syndromes in respective order, in which ruling out irrelevant error patterns and pinpointing the exact one for successful error correction.

Selection of *H* is a precautionary step taken to ensure feasibility of the matching analysis. With other formation of *H*, the established relationships connecting difference between syndromes and associated error patterns in forward and reverse orders may end up leaving more than one pair of identical error patterns after carrying out matching analysis, which giving rise to uncertainty in pinpointing the exact error pattern among residual ones and hence causing unsuccessful error correction. An example is given using H formatted as

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (14)$$

which is a formation specified in [24, 36]. Associations of error patterns with difference between syndromes in respective order using the formation of H as defined by (14) are shown in Fig. 9.



Difference between syndromes	Error pattern			
000	0000000			
100	1000000	0110000	0001100	0000011
010	0100000	1010000	0001010	0000101
001	0001000	1000100	0100010	0010001
110	0010000	1100000	0001001	0000110
101	0000100	1001000	0100001	0010010
011	0000010	1000001	0101000	0010100
111	0000001	1000010	0100100	0011000

Forward order

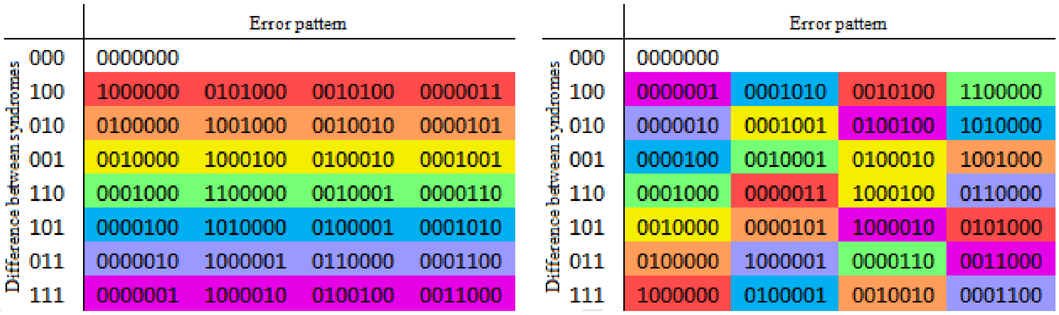
Difference between syndromes	Error pattern			
000	0000000			
100	0000001	0000110	0011000	1100000
010	0000010	0000101	0101000	1010000
001	0001000	0010001	0100010	1000100
110	0000100	0000011	1001000	0110000
101	0010000	0001001	1000010	0100100
011	0100000	1000001	0001010	0010100
111	1000000	0100001	0010010	0001100

Reverse order

Fig. 9: The established relationships that foil matching analysis.

Error patterns associated with the same syndrome in forward order are all tagged using the same color for easier tracing of their distribution in reverse order. It can then be effortlessly spotted that whenever difference between syndromes in respective order is non-zero, or in other words, whenever a 7-bit block of sifted key deduced by Bob is erroneous, at least two pair of error patterns are associated with the same syndrome in forward order and that of reverse order, irrespectively. Therefore, two or more pair of identical error patterns are left behind after carrying out matching analysis, for example, “0001100” is an adhering error pattern that cannot be weeded out after performing the matching analysis if “1000000” is the exact error occurred and vice versa. As a result of the ambiguous error pattern, a complete error correction cannot be preceded. Hence, formation of H has to be chosen cautiously to prevent the onset of such a plight.

The associations of error patterns with difference between syndromes in respective order using the formation of H as mentioned in (8), which is the one specified for this research, are shown in Fig. 10. The error patterns are likewise tagged using color for the same reason as stated earlier. It can then be observed that error patterns associated with the same non-zero syndrome in forward order are associated with assorted non-zero syndromes in reverse order. Any such or similar distribution ensures practicality of matching analysis and guarantees success of desirable error correction.



Difference between syndromes	Error pattern			
000	0000000			
100	1000000	0101000	0010100	0000011
010	0100000	1001000	0010010	0000101
001	0010000	1000100	0100010	0001001
110	0001000	1100000	0010001	0000110
101	0000100	1010000	0100001	0001010
011	0000010	1000001	0110000	0001100
111	0000001	1000010	0100100	0011000

Forward order

Difference between syndromes	Error pattern			
000	0000000			
100	0000001	0001010	0010100	1100000
010	0000010	0001001	0100100	1010000
001	0000100	0010001	0100010	1001000
110	0001000	0000011	1000100	0110000
101	0010000	0000101	1000010	0101000
011	0100000	1000001	0000110	0011000
111	1000000	0100001	0010010	0001100

Reverse order

Fig. 10: The established relationships that facilitate matching analysis.

The syndromes calculated and sent from Alice to Bob definitely give away partial information to Eve. By having an arbitrary7-bit block of Alice's sifted key represented by $\left[a_1\,a_2\,a_3\,a_4\,a_5\,a_6\,a_7\right]$and using dedicated H for this research given by (8), the values of corresponding syndrome in forward order, which are represented by $\left[s_1\,s_2\,s_3\right]$ and assumed to be known by Eve, are given as

$$s = r \cdot H^T$$

$$\left[s_1\,s_2\,s_3\right] = \left[a_1\,a_2\,a_3\,a_4\,a_5\,a_6\,a_7\right]\begin{bmatrix}1 & 0 & 0 & 1 & 1 & 0 & 1\\ 0 & 1 & 0 & 1 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 1 & 1 & 1\end{bmatrix}^T \tag{15}$$

$$\begin{bmatrix}s_1\\ s_2\\ s_3\end{bmatrix} = \begin{bmatrix}a_1 \oplus a_4 \oplus a_5 \oplus a_7\\ a_2 \oplus a_4 \oplus a_6 \oplus a_7\\ a_3 \oplus a_5 \oplus a_6 \oplus a_7\end{bmatrix}$$

It is evident that given any values for a_4, a_5, a_6 and a_7, which are equivalent to message bits of a codeword, there exist unique values for a_1, a_2 and a_3, which are equivalent to redundancy bits of a codeword, such that the syndrome of $\left[a_1\,a_2\,a_3\,a_4\,a_5\,a_6\,a_7\right]$in forward order is the publicly known $\left[s_1\,s_2\,s_3\right]$. For a given syndrome in forward order, there is not a way letting Eve to count out any of the 16 equally likely possibilities for value of a_4, a_5, a_6 and a_7 as a whole, if a_1, a_2 and a_3 are thrown away. Hence, the bits of sifted key that resemble the redundancy bits of a codeword are reasonably discarded to invalidate the partial information acquired by Eve [10].

The usage of syndrome in reverse order narrows the possibilities down to just two. This is because exactly two 7-bit blocks of Alice's sifted key, which are complement of each other, will be resulting in the same syndrome in forward order and the same syndrome in reverse order. As an instance, referring to Fig. 3.1, "1110001" and "0001110" are the merely two 7-bit blocks of sifted key that result in "000" as syndrome in forward order and "101" as syndrome in reverse order. Since there are only two remaining possibilities, one bit is already adequate to represent both of them. As a result, after discarding bits of sifted key that resemble the redundancy bits of a codeword, another three bits in an originally 7-bit block of sifted key are also discarded as complementary privacy maintenance to wipe out Eve's additional information that is contributed by syndrome in reverse order. Only the fourth bit in an arbitrary7-bit block of Alice's and Bob's sifted key will be reserved for subsequent privacy amplification to distill final secret key, if that block is successfully reconciled.

The algorithm of proposed reconciliation protocol that rectifies errors of BB84 protocol in single pass with maximum of double-bit error correcting capability is shown in Fig. 11. First of all, the position of bits in Alice's and Bob's string of sifted key is randomly permuted via folio interlacement such that possible sequent errors are dispersed at random. The shuffled strings of sifted key are partitioned by both parties into blocks that comprise seven bits out of the total bits each. Alice has syndrome of the first block of sifted key calculated in both forward and reverse orders using her portion of sifted key, and then sent to Bob via the authenticated classical channel. Meanwhile, Bob also has syndrome of the first block of sifted key calculated in both orders using his portion of sifted key. Syndrome measurement is carried out by Bob using received syndromes in tandem with his calculated syndromes to compute difference between their syndromes and determine associated error patterns in both orders.

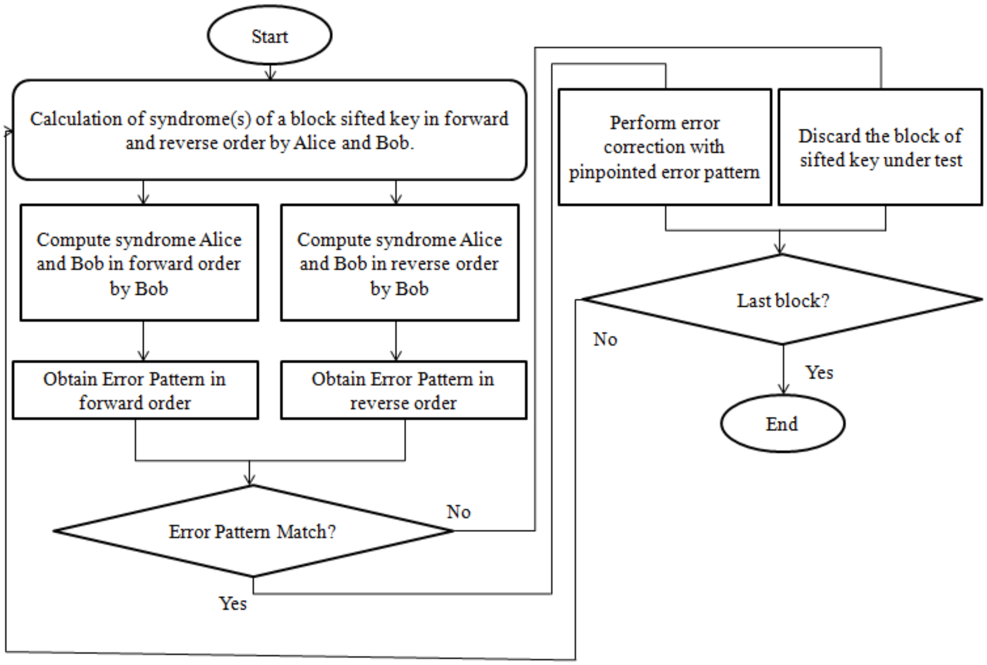

Start
Calculation of syndrome(s) of a block sifted key in forward and reverse order by Alice and Bob.
Compute syndrome Alice and Bob in forward order by Bob
Compute syndrome Alice and Bob in reverse order by Bob
Obtain Error Pattern in forward order
Obtain Error Pattern in reverse order
Error Pattern Match?
No
Yes
Perform error correction with pinpointed error pattern
Discard the block of sifted key under test
Last block?
No
Yes
End

Fig. 11: Flow of the proposed SP 1985 reconciliation protocol.

The matching analysis is carried out by Bob to determine the identical error pattern associated with difference between syndromes in respective order. The conditional decision to be made by Bob will be if there is a match of identical error pattern after performing matching analysis, error correction is performed by adding his block of sifted key under test with pinpointed error pattern bitwise using binary XOR operation. Otherwise, there is not a match of identical error pattern after performing matching analysis.

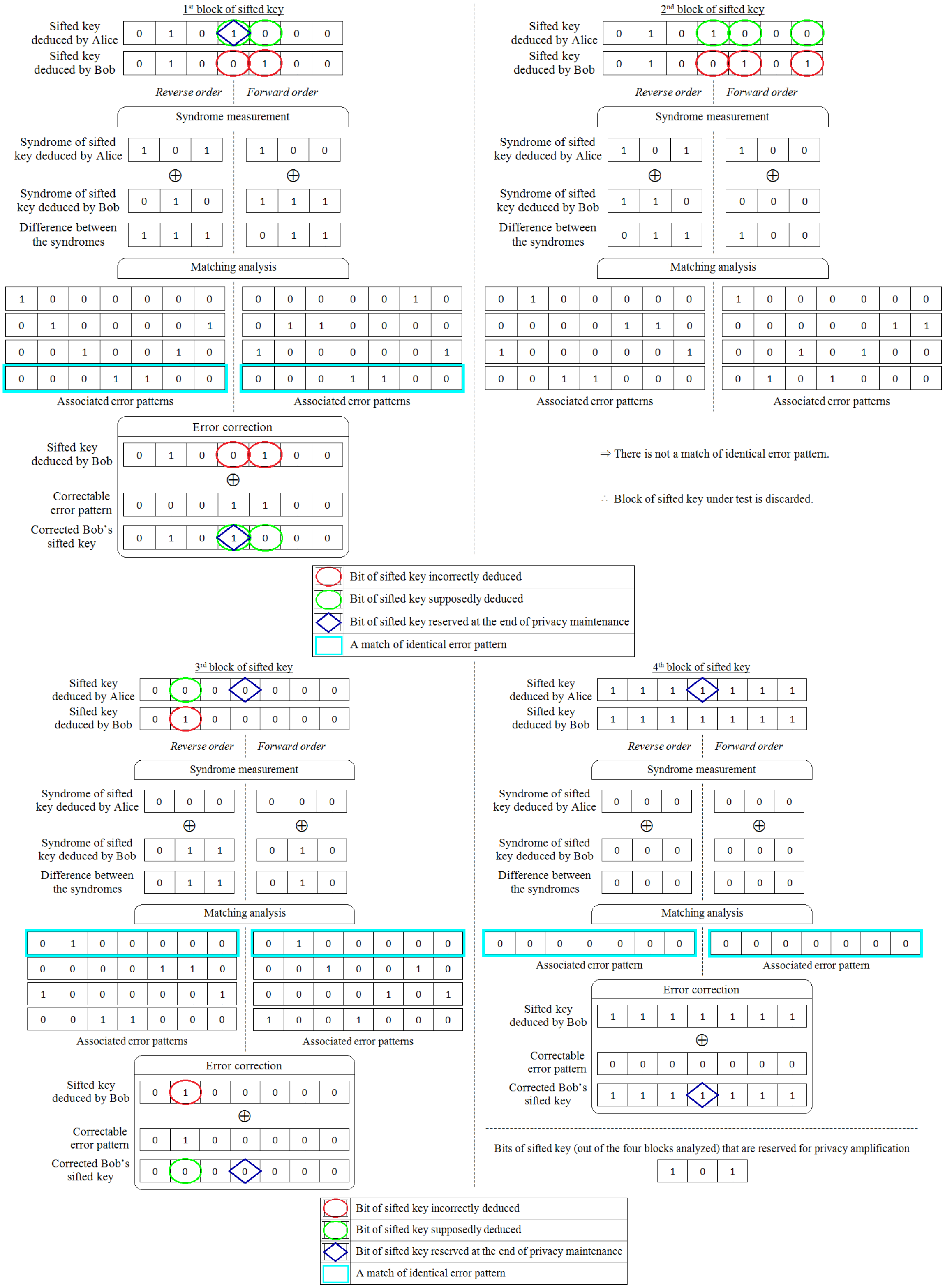

1st block of sifted key
2nd block of sifted key
3rd block of sifted key
4th block of sifted key
Sifted key deduced by Alice
Sifted key deduced by Bob
Reverse order
Forward order
Syndrome measurement
Syndrome of sifted key deduced by Alice
Syndrome of sifted key deduced by Bob
Difference between the syndromes
Matching analysis
Associated error patterns
Associated error pattern
Error correction
Correctable error pattern
Corrected Bob's sifted key
⇒ There is not a match of identical error pattern.
∴ Block of sifted key under test is discarded.
Bits of sifted key (out of the four blocks analyzed) that are reserved for privacy amplification
1 0 1
Bit of sifted key incorrectly deduced
Bit of sifted key supposedly deduced
Bit of sifted key reserved at the end of privacy maintenance
A match of identical error pattern

Fig. 12: Reconciliation and privacy maintenance utilizing the proposed SP 1985 protocol.

Error correction is skipped and his block of sifted key under test is discarded with a notification sent to Alice via the authenticated classical channel such that corresponding block of her sifted key is discarded too. Procedures are kept repeated for ensuing blocks of sifted key before the last block is analyzed. For all the blocks of sifted key that are successfully reconciled, the fourth bit in each block is reserved while the rest are discarded by both parties on account of privacy maintenance.

The reconciliation and privacy maintenance utilizing the proposed protocol is shown in Fig. 12. Remarkably, a 7-bit block of sifted key may be interspersed with three or four bits of error, but such a block will be discarded during reconciliation in accordance with fourth step of the proposed protocol.

4.0 THE UNIQUENESS OF PROPOSED SP 1985 RECONCILIATION PROTOCOL

The proposed SP 1985 reconciliation protocol is capable to discharge the following tasks with precision to detect:

- absence of error in a 7-bit block of sifted key that is errorless, and correct nothing with all-zero error pattern.
- occurrence of error in a 7-bit block of sifted key that is interjected with one or two bits of error, and correct the erroneous bit(s) with associated error pattern.
- occurrence of error in a 7-bit block of sifted key that is interspersed with three or four bits of error without correction.

The first and second capabilities were rationalized in section 3. The third capability is an ancillary feature that comes along with application of the proposed SP 1985 reconciliation protocol in cooperation with the dedicated *H* indicated by (8).

Table 1 shows a probability distribution which shows the possibilities of a segment of sifted key in bits being susceptible to erroneous variation, in relation to number of errors redistributed into a 7-bit block after carrying out random shuffling. Random shuffling is required such that possible sequent errors in the string of sifted key are mostly dispersed at random prior to efficient reconciliation and gives rise to binomially distributed error [36, 47]. It can be completed via folio interlacement by halving and then interweaving a string of sifted key again and again [36]. Hence, the probability of occurrence of error in a specified block is given by binomial distribution expressed as

$$P(X = x) = \binom{N}{x} p_0^{\,x} (1 - p_0)^{N-x} \tag{16}$$

where p_0 is the initial error probability [24, 36].

Table 1: Probability distribution for possible variants of a segment of sifted key.

Error in	**Probability of occurrence, *P* (X =**	**Cumulative probability, *P* (X ≤**
0	4.423 x 10^{-1}	4.423 x 10^{-1}
1	3.827 x 10^{-1}	8.250 x 10^{-1}
2	1.419 x 10^{-1}	9.669 x 10^{-1}
3	2.923 x 10^{-2}	9.961 x 10^{-1}
4	3.613 x 10^{-3}	9.997 x 10^{-1}
5	2.679 x 10^{-4}	9.999 x 10^{-1}
6	1.104 x 10^{-5}	≈ 1.000
7	1.949 x 10^{-7}	1.000

Statistically, it shows that about 96.69% of the situation is under controlled in single pass if up to double-bit error correction is made possible during reconciliation. An equivalent increment of 14.19% in relative to achievement of around 82.50% attained by single-bit error correction soundly implies that it is worthwhile to pursue reconciliation with double-bit error correcting capability.

For a 7-bit block of sifted key deduced by Bob that is interspersed with three or four bits of error, syndrome measurement results in at least one non-zero syndrome as the difference between syndromes computed by Bob in forward order and that of in reverse order, suggesting that the block of sifted key is erroneous. In spite of that, there is no match of identical error pattern associated with difference between syndromes in respective order after carrying out matching analysis, conjecturing that the block of sifted key is interspersed with more than two bits of error.

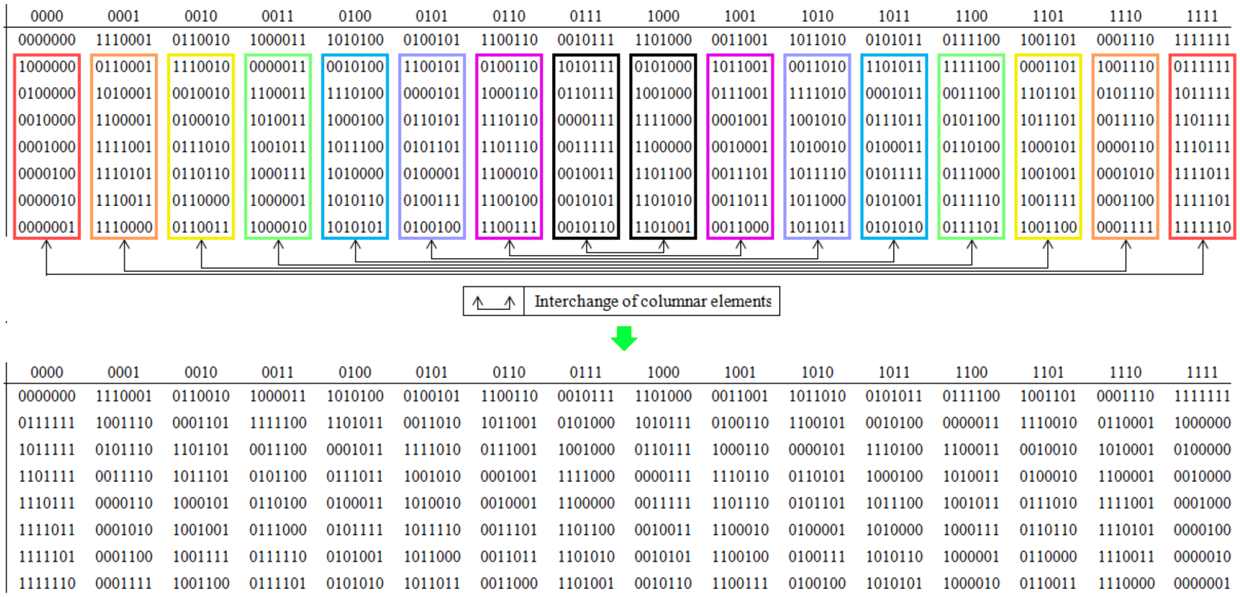
There cannot be a match of identical error pattern after performing matching analysis because only single-bit and double bit error patterns can be associated with each non-zero syndrome, for instance a rule governed by the bound of Hamming code's $d_{\min}$. Hence, if three or four bits out of seven bits of sifted key in a block of sifted key deduced by Bob are erroneous, that block is expediently discarded by Alice and Bob after its reconciliation, as errors in it can be detected but not corrected. Referring to Table 1, the statistics shows that about 99.97% of random erroneous variation is manageable in single pass with the subsidiary capability added to the proposed SP 1985 reconciliation protocol. However, it is remarked that the special feature may not be enabled by other formation of *H*.

At the same time, for bits of sifted key deduced by Bob, the proposed SP 1985 reconciliation protocol of the same specification is limited to discharge the following tasks.

- To correct five or six erroneous bits out of seven bits of sifted key in a block that is interspersed with five or six bits of error, using associated error pattern.
- To detect occurrence of error in a 7-bit block of sifted key that is interspersed with seven bits of error, and correct it with associated error pattern.

It is possible for the proposed SP 1985 reconciliation protocol to detect occurrence of error in a 7-bit block of sifted key that is interspersed with five or six bits of error during syndrome measurement, which yields non-zero syndromes as the difference between syndromes computed by Bob in forward order and that of in reverse order. However, the erroneous block of sifted key will then be corrected with complement of associated error pattern instead of the actual associated error pattern, leading to complement of sifted key rather than errorless sifted key in reference. Such a deadlock is inherited from complementary error correcting capability of Hamming code (7, 4, 3) itself, which is a novel discovery in this research. It is actually possible for the Hamming code (7, 4, 3) to perform sextuple-bit error correction, which complements its popularly acclaimed single-bit error correction, with slight alteration made in the standard array.

An example is shown in Fig. 13 to illustrate the only simple step necessary. All it needs is a little swapping of erroneous codewords with their respective complements (grouped using Hollow Square of the same colour in Fig. 13) correspondingly. With such interchanges, sextuple-bit error patterns are assigned as the associated correctable error pattern, and all possible erroneous variants of each valid codeword are correctly reallocated, allowing the valid codewords that are interspersed with six bits of errors to be corrected by looking up the altered standard array. Apart the complementary error correcting capability, it is impossible for single-bit and sextuple-bit error corrections to be carried out simultaneously.



0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000000	1110001	0110010	1000011	1010100	0100101	1100110	0010111	1101000	0011001	1011010	0101011	0111100	1001101	0001110	1111111
1000000	0110001	1110010	0000011	0010100	1100101	0100110	1010111	0101000	1011001	0011010	1101011	1111100	0001101	1001110	0111111
0100000	1010001	0010010	1100011	1110100	0000101	1000110	0110111	1001000	0111001	1111010	0001011	0011100	1101101	0101110	1011111
0010000	1100001	0100010	1010011	1000100	0110101	1110110	0000111	1111000	0001001	1001010	0111011	0101100	1011101	0011110	1101111
0001000	1111001	0111010	1001011	1011100	0101101	1101110	0011111	1100000	0010001	1010010	0100011	0110100	1000101	0000110	1110111
0000100	1110101	0110110	1000111	1010000	0100001	1100010	0010011	1101100	0011101	1011110	0101111	0111000	1001001	0001010	1111011
0000010	1110011	0110000	1000001	1010110	0100111	1100100	0010101	1101010	0011011	1011000	0101001	0111110	1001111	0001100	1111101
0000001	1110000	0110011	1000010	1010101	0100100	1100111	0010110	1101001	0011000	1011011	0101010	0111101	1001100	0001111	1111110

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000000	1110001	0110010	1000011	1010100	0100101	1100110	0010111	1101000	0011001	1011010	0101011	0111100	1001101	0001110	1111111
0111111	1001110	0001101	1111100	1101011	0011010	1011001	0101000	1010111	0100110	1100101	0010100	0000011	1110010	0110001	1000000
1011111	0101110	1101101	0011100	0001011	1111010	0111001	1001000	0110111	1000110	0000101	1110100	1100011	0010010	1010001	0100000
1101111	0011110	1011101	0101100	0111011	1001010	0001001	1111000	0000111	1110110	0110101	1000100	1010011	0100010	1100001	0010000
1110111	0000110	1000101	0110100	0100011	1010010	0010001	1100000	0011111	1101110	0101101	1011100	1001011	0111010	1111001	0001000
1111011	0001010	1001001	0111000	0101111	1011110	0011101	1101100	0010011	1100010	0100001	1010000	1000111	0110110	1110101	0000100
1111101	0001100	1001111	0111110	0101001	1011000	0011011	1101010	0010101	1100100	0100111	1010110	1000001	0110000	1110011	0000010
1111110	0001111	1001100	0111101	0101010	1011011	0011000	1101001	0010110	1100111	0100100	1010101	1000010	0110011	1110000	0000001

Fig. 13: Alteration of standard array for sextuple-bit error correction.

Similarly, once double-bit error correction is enabled by the proposed SP 1985 reconciliation protocol in fulfilling primary objective of this research, quintuple-bit error correction is enabled as complementary error correcting capability as well but they cannot operate mutually at the same moment. Resultantly, an erroneous block of sifted key that is interspersed with five or six bits of error, will be corrected with complement of associated error pattern instead of the actual associated error pattern when undergoing the proposed SP 1985 reconciliation protocol, leading to complement of sifted key rather than errorless sifted key in reference after error correction. On the other hand, it is impossible for SP 1985 reconciliation protocol to detect occurrence of error in a 7-bit block of sifted key during syndrome measurement if all the bits in the block are erroneous.

This is because in such a scenario, all-zero syndromes, which are false indicators now due to complementary error correcting capability, will be yielded as the difference between syndromes computed by Bob in forward order and that of in reverse order, incorrectly suggesting that the block of sifted key is errorless and leaving the erroneous block not corrected. In order to suppress adverse effect of the complementary error correcting capabilities, occurrence of five or more bits of error in a 7-bit block of sifted key is subdued via statistical treatment, which is included in the following section.

5. Performance Evaluation over the Proposed SP 1985 Reconciliation Protocol

The proposed SP 1985 protocol for reconciliation is simulated using MATLAB®2013. Table 2 shows the design parameters of this research and their corresponding values chosen for the simulation based on the preset *H* as in (8).

Table 2: Design parameters and corresponding values.

Design parameters	Values
Length of sequent error in bits	5
Length of sifted key in bits	1000
Repetition of shuffling in rounds	50

The length of sequent error is defined as the amount of consecutive erroneous bits upon occurrence of error in the string of sifted key deduced by Bob for this research. Although the least common multiple of two and five are ten, which is a value that facilitates calculation of QBER in the phase of evaluation, the length of sequent error of two bits is not chosen for simulation, as it tends to give rise to the biased modelling of sequent error by neglecting the possible occurrence of sequent errors that are beyond double-bit error correcting capability of the proposed solution. Hence, the length of sequent error of five bits is chosen for the simulation.

The length of sifted key is defined as the amount of bits of sifted key to be reconciled in single pass by Bob for this research. The value is set as 1000 bits after adopting idea of expected value of binomially distributed random error *E*(X) to subdue the occurrence of five or more bits of error in a 7-bit block of sifted key. The theory is formulated as

$$E(X) = \frac{length\ of\ sifted\ key}{length\ of\ a\ block\ of\ sifted\ key} \times P(X = x) \qquad (17)$$

Using probability retrieved from Table 1, the length of sifted key of 1000 bits and length of a block of sifted key of seven bits, the expected value of occurrence of erroneous 7-bit blocks of sifted key that make up the total number of erroneous bits in the string of sifted key is quantified as shown in Table 3.

Table 3: Expected value for erroneous blocks of sifted key.

Error in bit(s), *x*	**Probability of occurrence, *P* (X = *x*)**	**Expected value, *E* (X) for 1000 bits of sifted key**	**Expected value, *E* (X) for 3000 bits of sifted key**
1	3.827×10^{-1}	54.67	164.2
2	1.419×10^{-1}	20.27	60.88
3	2.923×10^{-2}	4.176	12.54
4	3.613×10^{-3}	5.161×10^{-1}	1.550
5	2.679×10^{-4}	3.827×10^{-2}	1.149×10^{-1}
6	1.104×10^{-5}	1.577×10^{-3}	4.736×10^{-3}
7	1.949×10^{-7}	2.784×10^{-5}	8.361×10^{-5}

It is evident from the statistical data of Table 3 that erroneous bits in a string of sifted key is mostly predominated by occurrence of one, two, three or four bits of error in a 7-bit block of sifted key as their expected values are relatively significant as compared to five or more bits of error. Hence, five or more bits of error should rarely occur in a 7-bit block of sifted key when length of sifted key is set as 1000 bits. Further increment in the length of sifted key, saying 3000 bits, will have the expected value of occurrence of erroneous 7-bit block of sifted key that is interjected with five bits of error raised to 1.148 x 10-1 by applying (17), which is a value almost as significant as expected value of occurrence of erroneous 7-bit block of sifted key that is interjected with four bits of error, in which the length of sifted key is set as 1000 bits. This implies that one 7-bit block of sifted key might be interjected by five bits of errors when the length of sifted key is extended to 3000 bits, which is unfavourable and should be avoided. Whenever the string of sifted key deduced by Bob is more than 1000 bits, bits of sifted key is reconciled in smaller proportion, each comprising maximum of 1000 bits, in stages within single pass. Finally, through trial-and-error method, it has been figured out that 50 rounds of random shuffling via folio interlacement is required to disperse sequent errors in a string of 1000-bit sifted key at random prior to efficient reconciliation.

With the aforementioned setting, simulation is initiated by generating two strings of sifted key; one is errorless as reference, while the other is interjected with sequent errors. Both strings underwent segmentation, random shuffling, syndrome computation, matching analysis, appropriate reconciliation, privacy maintenance and combination. The simulation is analyzed using different initial QBER (QBER prior to reconciliation), and evaluated against final QBER (QBER right after reconciliation).

Fig. 14 shows the generic simulation result that emphasizes effectiveness of proposed SP 1985 reconciliation protocol. This figure is plotted with final QBER as a function of initial QBER ranging from 0% to 11%, in which the latter is the threshold of tolerable QBER for BB84 protocol using one-way classical post-processing, to highlight the suppression of initial QBER achieved via proposed solution of this research in single pass. Ideally, the proposed SP

1985 reconciliation protocol is capable in quashing initial QBER within the range of interest completely when all the erroneous blocks of sifted key are in reach of error correcting capability of the proposed reconciliation protocol, such that up to two bits of error in an erroneous 7-bit block of sifted key. Conversely, some of the erroneous blocks of sifted key are out of reach of error correcting capability of proposed reconciliation protocol when the initial QBER is raised to 3% or above. The corresponding final QBER is plotted as worst case scenario. Anyhow, there is non-occurrence of five or more bits of error in a 7-bit block of sifted key during simulation, and the non-zero final QBER is constituted by the occurrence of three or four bits of error in the erroneous block of sifted key, which can be identified by the subsidiary capability of SP 1985 protocol and then discarded. Since the occurrence of three or four bits of error in a 7-bit block of sifted key is much less likely compared with that of two or less bits of error, the abandonment as an expedient move does not sabotage overall performance of SP 1985 protocol. As a result, every single erroneous bit in a string of sifted key can be fixed in single pass, one way or another via the proposed SP 1985 protocol, affirming the effectiveness of the proposed solution.

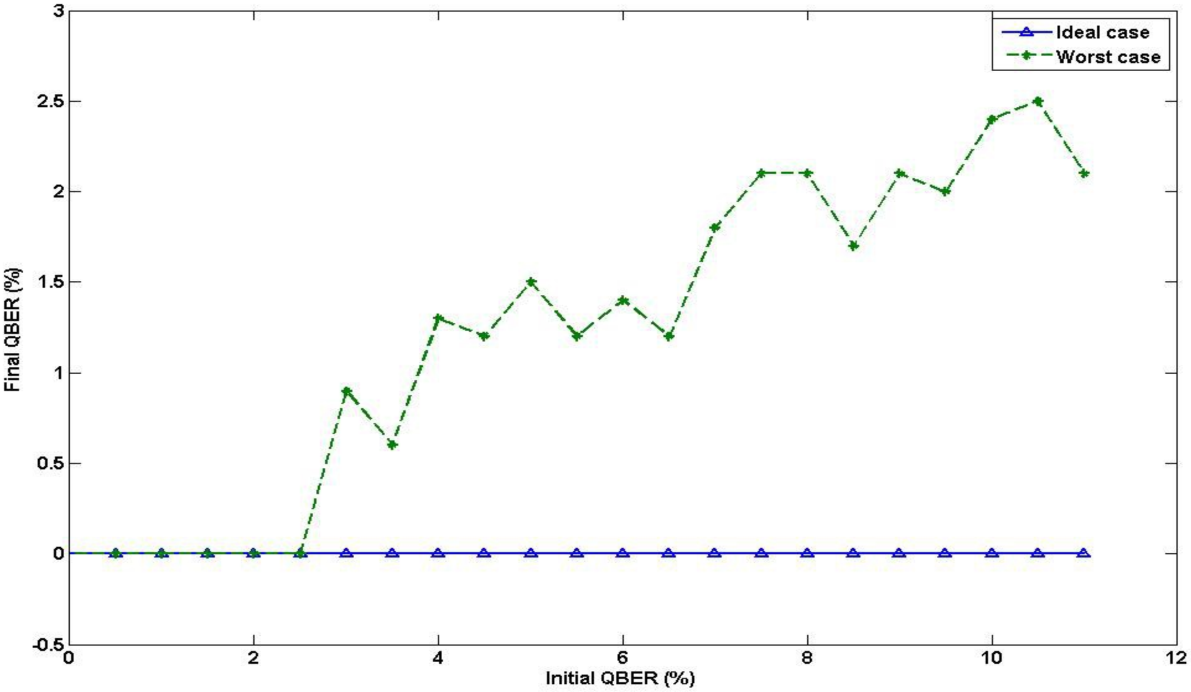


Fig. 14: Generic simulation of final QBER versus initial QBER of SP 1985 reconciliation protocol.

Fig. 15 shows the simulation result in comparison with BBBSS protocol applying parity check and improved BBBSS protocol applying CRC. It is plotted with QBER as a function of number of iteration required for reconciliation. The plots that correspond to alternative approaches are directly applying the data readily available in [42]. The length of sifted key of 2464 bits and initial QBER of 2.3% are used in the simulation. The initial QBER slips gradually throughout iterative reconciliation, but a little quicker with application of CRC as compared with the achievement obtained by original BBBSS protocol applying parity check. Additionally, with the proposition, an improvement in efficiency of around 50% in terms of number of disclosed bitsis recorded in the findings.

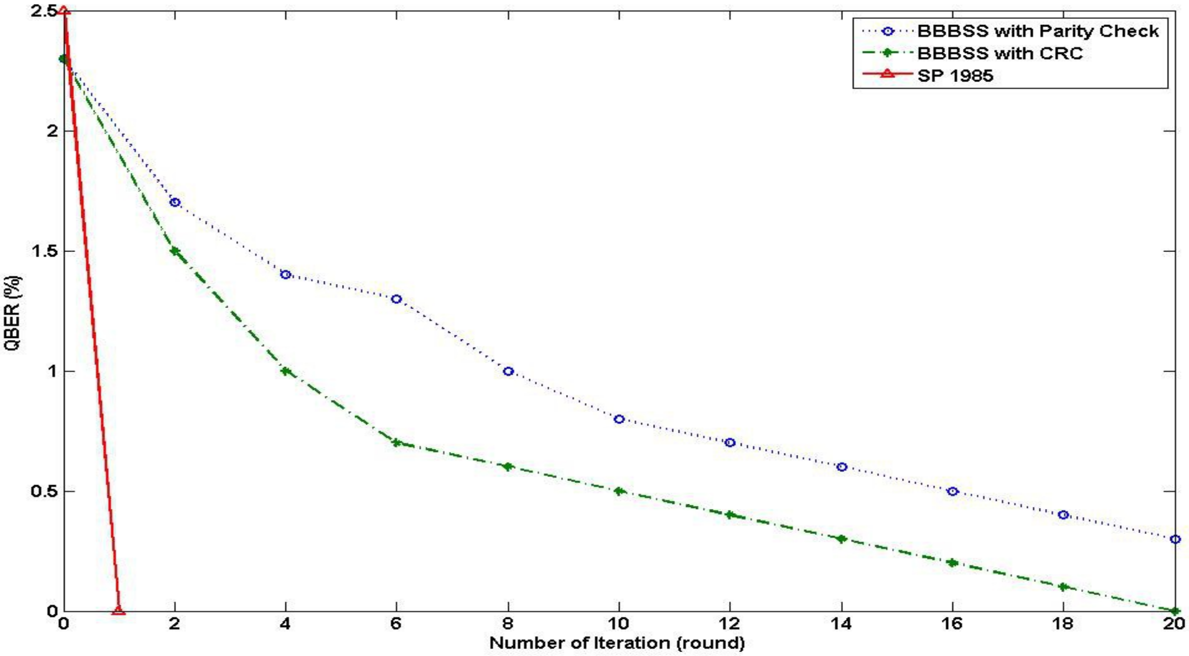


Fig. 15: Graph of QBER versus number of iteration in reference to BBBSS protocol applying CRC.

Fig. 16 shows the simulation result in comparison with BBBSS protocol applying parity check and improved BBBSS protocol applying cryptographic hash function (MD5 in particular). It is plotted with QBER as a function of number of iteration required for reconciliation. The plots correspond to alternative approaches which are directly applying the data readily available in [43]. The length of sifted key of 2459 bits and initial QBER of 1.87% are used in their simulation. The initial QBER drops moderately throughout iterative reconciliation. It drops in faster pace with application of MD5 as compared with achievement obtained by original BBBSS protocol applying parity check. In addition, with the proposition, an improvement in efficiency of around 45% in terms of number of disclosed bits is recorded in the findings. In fact, the performance is similar to that of replacing parity bit with CRC in BBBSS protocol.

Outmatching both propositions have been discussed, which the proposed SP 1985 protocol manages to suppress initial QBER entirely in single pass and all the errors are corrected without iteration. The length of sifted key of 2459 bits, which will be reconciled in stages, and initial QBER of 2% are used in simulation. The success is attributed to capability of SP 1985 protocol in correcting up to two bits of error in an erroneous 7-bit block of sifted key. Although the proposition of application of CRC or MD5 allows detection of all odd and even number of erroneous bits, only one bit of error can be eliminated in a block of sifted key in single pass due to the restriction imposed by operation of BBBSS protocol, rendering iteration indispensable for complete reconciliation. Moreover, it is believed that further increment of initial QBER will neutralize the improvement in efficiency of propositions as discussed in terms of number of disclosed bits as larger number of bits will be disclosed to reconcile higher number of errors, aggravating interactivity between Alice and Bob in turn. Overall the proposed SP 1985 protocol is a better choice than proposition of improved BBBSS protocol applying CRC or MD5 in both terms of effectiveness and efficiency, especially in situation of relatively high initial QBER.

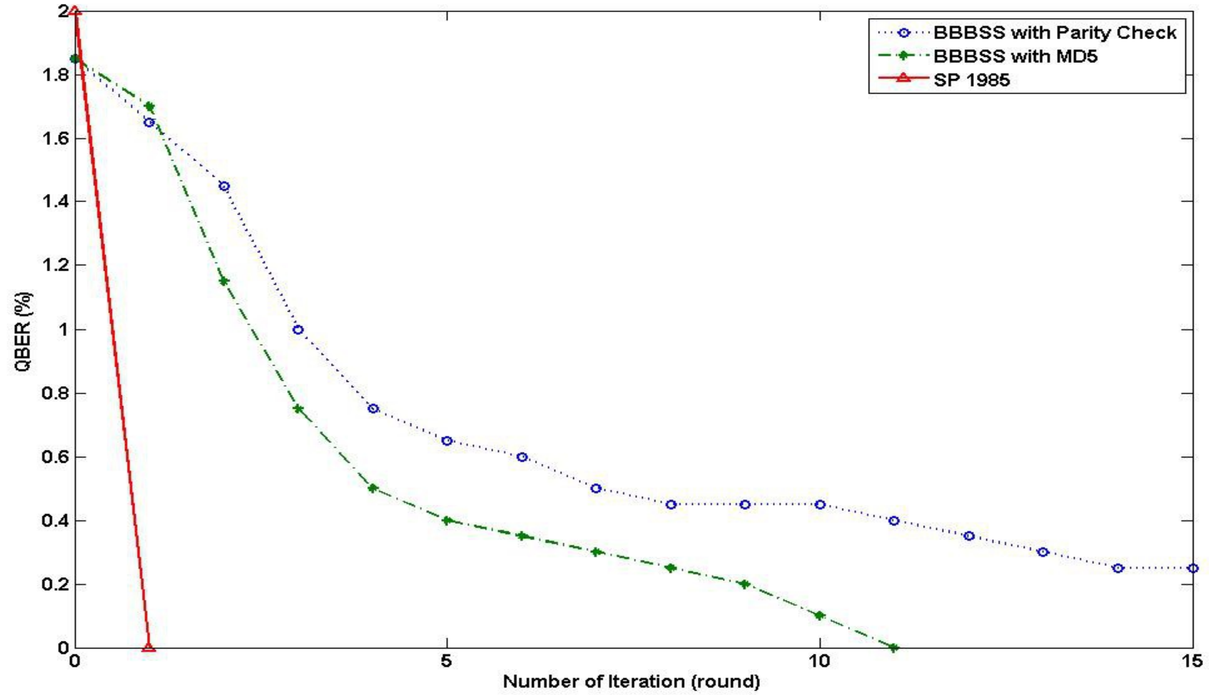


Fig. 16: Graph of QBER versus number of iteration in reference to BBBSS protocol applying MD5.

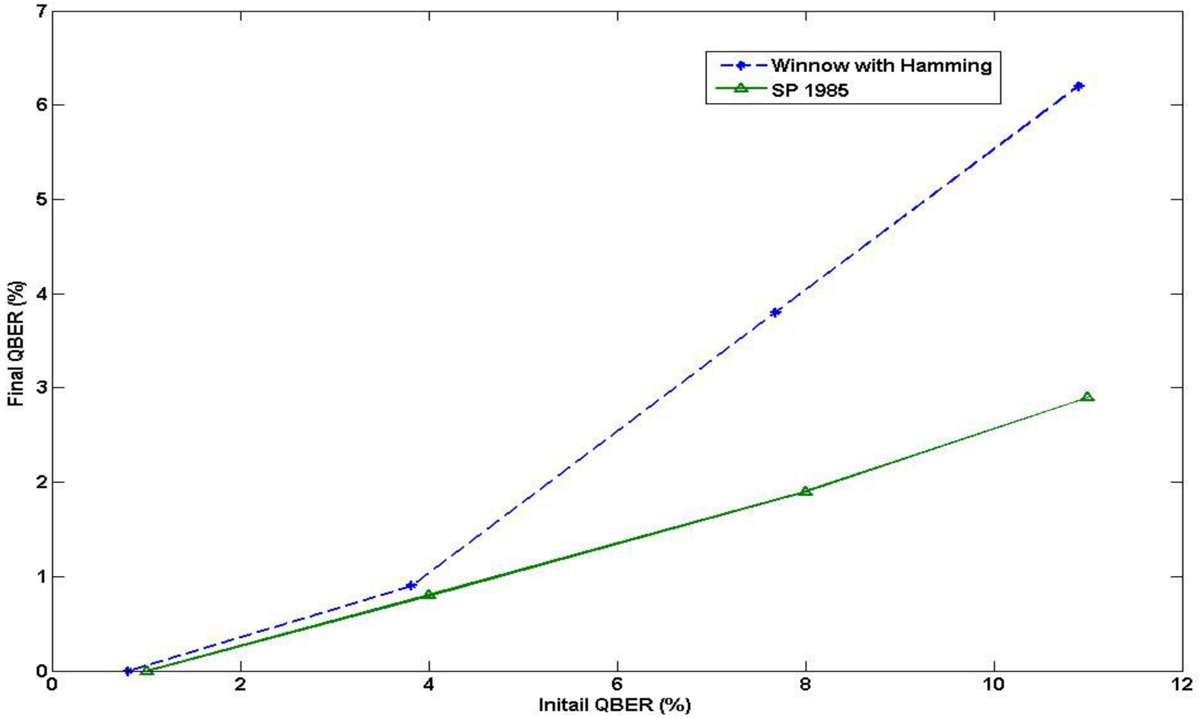


Fig. 17: Graph of final QBER versus initial QBER in reference to Winnow protocol applying parity check and Hamming code.

Fig. 17 shows the simulation result in comparison with Winnow protocol that applying parity check and Hamming code. The plot corresponds to the Winnow protocol that is directly applying the data readily available in [36]. It is plotted with final QBER at the end of reconciliation in single pass as a function of particular initial QBER of 0.80%, 3.81%, 7.68% and 10.90%. The length of sifted key of about 3000 bits and optimized block size are used for initial QBER in this simulation. The figure shows that the final QBER posts a rise in response to increment of initial QBER for both reconciliation protocols, but the percentages recorded in proposed SP 1985 protocol are lower than those of Winnow protocol.

The difference is significant in initial QBER ranging from 4% to 11%. It is due to the capability of proposed SP 1985 protocol in correcting up to two bits of error in an erroneous 7-bit block of sifted key, which is a feature not possessed by Winnow protocol. Furthermore, unlike the proposed SP 1985 protocol, Winnow protocol is incapable of identifying and discarding the erroneous blocks of sifted key, which constitutes toward number of remaining errors at the end of reconciliation in single pass. Hinging on the limited single-bit error correcting capability, several iterations are necessary for a complete reconciliation using Winnow protocol in general.

Fig. 18 shows the simulation result in comparison with improved Winnow protocol applying parity check and convolutional code. It is plotted with final QBER at the end of reconciliation in single pass as a function of initial QBER ranging from 1% to 10%. The plot that corresponds to Winnow protocol with convolutional code is directly applying the data readily available in [13]. The length of sifted key of 100000 bits is used in the averaged values of 100 trials. The simulation result shows that the trend corresponds to SP 1985 protocol outperforms Winnow protocol applying convolutional, although any odd number of erroneous bits in a block of sifted key can be corrected via the improved Winnow protocol. This better performance is due to the same reasoning as stated for comparison with Winnow Protocol applying Hamming code in Fig. 17. Therefore, by taking everything into account, the proposed SP 1985 protocol of this research is a prudent choice than proposition of Winnow protocol in both terms of effectiveness and efficiency, especially in situation of relatively high initial QBER.

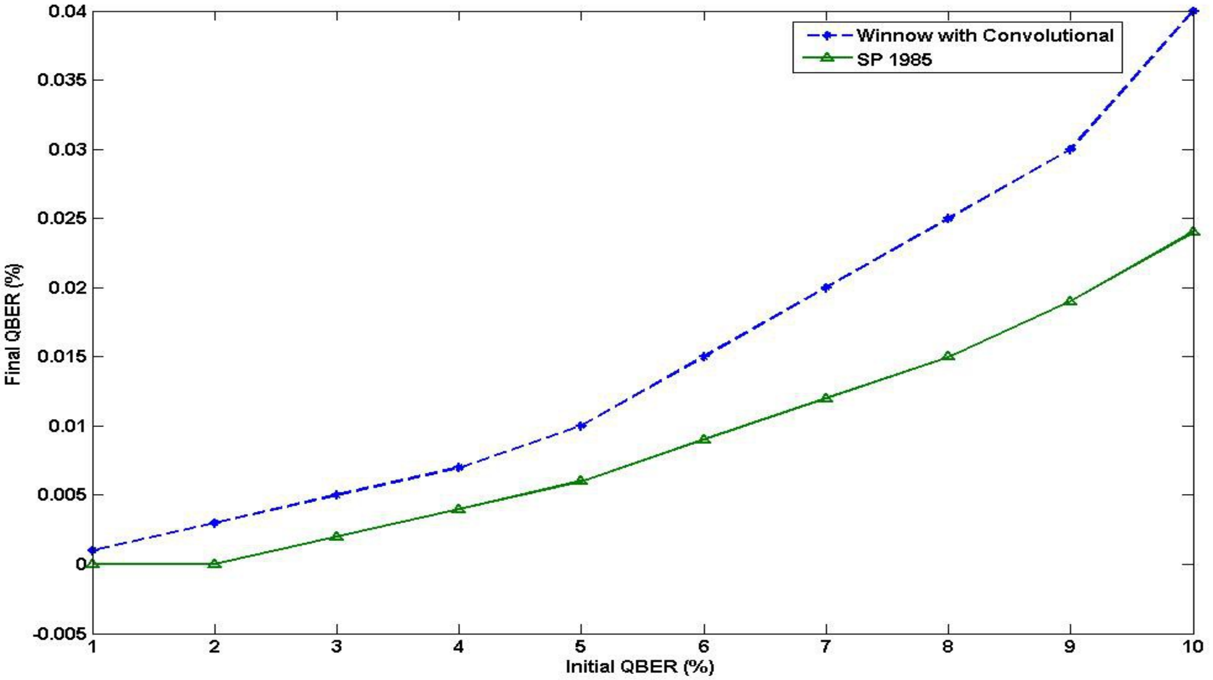


Fig. 18: Graph of final QBER versus initial QBER in reference to Winnow protocol applying parity check and convolutional code.

In pursuing effectiveness and efficiency of reconciliation in terms of number of iteration required, the proposed SP 1985 protocol does not share the weaknesses found in other alternatives. Unlike generator polynomial or cryptographic hash function of improved BBBSS protocols, the parity-check matrix of SP 1985 protocol, which is the basis for reconciliation, can be synchronized between Alice and Bob publicly without encryption. As a non-interactive reconciliation protocol, the proposed SP 1985 protocol eliminates the interactivity between Alice and Bob, which is part of procedure of BBBSS and Cascade protocols, when performing error correction.

Following the predefined procedure, the proposed SP 1985 protocol will not have any block of sifted key deduced by Bob inappropriately treated as in Winnow protocol. Nevertheless, it is sufficient number of disclosed bits required in exchange of prompt reconciliation. In turn, the number of bits to be discarded for privacy maintenance must always be increased correlatively with increment of number of disclosed bits.

5.0 CONCLUSIONS

The quantum error correcting code such as Hamming code which used in Winnow protocol is found to be more attractive for the reconciliation in BB84 protocol. However, the Winnow protocol can only correct one error out of seven bits. In this paper, a new reconciliation protocol, SP 1985 protocol has been developed to enhance the error correcting capability in BB84 protocol. This reconciliation protocol that is capable in correcting up to two bits of error in an erroneous 7-bit block of sifted key, has been presented by applying simple Hamming (7, 4, 3) code. The syndrome measurement is done twice in slightly distinctive manner such that two set of error patterns in respect to two set of syndromes are made available for matching analysis. Thus, it is featured by analysing the codeword in forward and reverse orders where the exact error pattern should remain the same regardless of the direction whether from the MSB toward the LSB or vice versa. With this new interpretation of Hamming code's syndrome and an unprecedented matching analysis, occurrence of three or four bits of error in the erroneous block of sifted key can also be identified by the proposed reconciliation protocol. The results show that typical flow of Winnow protocol has been simplified when using SP 1985 protocol in the reconciliation process to reduce the interactivity which leads to both terms of effectiveness and efficiency improvement.

REFERENCES

[1] Denis,T.S., Johnson, S.: Cryptography for Developers, SyngressPublishing Inc., Rockland, (2007).

[2] Zhang,R., Liu, L.: Security Models and Requirements for Healthcare Application Clouds. IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 268-275, (2010).

[3] Gisin, N., Ribordy, G., Tittel, W., Zbinden, H.: Quantum Cryptography. Reviews of Modern Physics, 74(1), 145-195, (2002).

[4] Tipton, H.F., Krause, M.: Information Security Management Handbook. Boca Raton: CRC Press, (2007).

[5] Chanson, S.T., Cheung, T.W.: Design and Implementation of a PKI-Based End-to-End Secure Infrastructure for Mobile E-Commerce. World Wide Web, Kluwer Academic, 4(4), 235-253, (2001).

[6] Jailani, N., MohdYatim, N.F., Yahya, Y., Patel, A., Othman, M.: Secure and Auditable Agent-Based E-Marketplace Framework for Mobile Users. Computer Standards & Interfaces, Elsevier, 30(4), 237-252, (2008).

[7] Advanced Encryption Standard (AES). Federal Information, Processing Standards, Publication 197, 26 Nov 2001.

[8] Suh, G.E., Clarke, D., Gassend, B., van Dijk, M., Devadas, S.: Efficient Memory Integrity Verification and Encryption for Secure Processors. Proceedings of the 36th International Symposium on Microarchitecture (MICRO-36'03), pp. 1-12, (2003).

[9] Rivest, R.L., Shamir, A., Adleman, L.M.: Cryptographic Communications System and Method", Patent No. US4405829 A, (1983).

[10] Loepp, S., Wootters, W.K.: Protecting Information: From Classical Error Correction to Quantum Cryptography, Cambridge University Press: New York, USA, (2006).

[11] Nagaraj, N.,Vaidya, V., Vaidya, P.G.: Re-visiting the One-Time Pad.International Journal of Network Security, 6(1), 94-102, (2008).

[12] Shannon, C.E.: Communication Theory of Secrecy Systems. Bell System Technical Journal, 656-715, (1948).

[13] Assche, G.V.: Quantum Cryptography and Secret-Key Distillation. Cambridge University Press: New York, USA, (2006).

[14] Fung, C.H.F.,Ma, X.,Chau,H.F., Cai, Q.Y.: Quantum Key Distribution with Delayed Privacy Amplification and Its Application to the Security Proof of a Two-Way Deterministic Protocol. Physical Review A, 85(3), 032308-1-032308-10, (2012).

[15] Lo, H.K.,Ma,X., Chen, K.: Decoy State Quantum Key Distribution. Physical Review Letters, 94(23), 230504-1-230504-4, (2005).

[16] Shor, P.W., Preskill, J.: Simple Proof of Security of the BB84 Quantum Key Distribution Protocol. Physical Review Letters, 85(2), 441-444, (2000).

[17] Scarani, V.,Acín, A.,Ribordy, G., Gisin, N.: Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations.Physical Review Letters, 92(5), 057901-1-057901-4, (2004).

[18] Lo, H.K.: Error Correction and Security in Quantum Cryptography. Proceedings of IEEE International Symposium on Information Theory, pp. 481, (2003).

[19] Busch, P.,Heinonen, T., Lahti, P.: Heisenberg's Uncertainty Principle. Physics Reports, 452(6), 155-176, (2007).

[20] Gottesman, D., Lo, H.K.: Proof of Security of Quantum Key Distribution with Two-Way Classical Communications. IEEE Transactions on Information Theory, 49(2), 457-475. (2003).

[21] Wiesner, S.: Conjugate Coding. ACM SIGACT News, 15(1), 78-88, (1983).

[22] Bennett, C.H., Brassard, G., Quantum Cryptography: Public Key Distribution and Coin Tossing. Theoretical Computer Science, Elsevier, 560(1), 7-11, (2014).

[23] Stipčević, M.: Quantum Random Number Generators and Their Use in Cryptography. Proceedings of the 34th International Convention MIPRO, pp. 1474-1479, (2011).

[24] Buttler, W.T.,Lamoreaux, S.K.,Torgerson, J.R.,Nickel, G.H.,Donahue,C.H., Peterson, C.G.: Fast, Efficient Error Reconciliation for Quantum Cryptography. Physical Review A, 67(5), 052303-1-052303-8, (2003).

[25] Ekert, A.K.: Quantum Cryptography Based on Bell's Theorem. Physical Review Letters, 67(6), 661-663, (1991).

[26] Bennett, C.H.,Bessette, F.,Brassard, G.,Salvail, L., Smolin, J.: Experimental Quantum Cryptography. Journal of Cryptology, 5(1), 3-28, (1992).

[27] Bennett, C.H., Brassard, G., Mermin, N.D.: Quantum Cryptography without Bell's Theorem. Physical Review Letters, 68(5), 557-559, (1992).

[28] Bruß, D.: Optimal Eavesdropping in Quantum Cryptography with Six States. Physical Review Letters, 81(14), 3018-3021, (1998).

[29] Boileau, J.C.,Tamaki, K.,Batuwantudawe, J.,Laflamme, R., Renes, J.M.: Unconditional Security of a Three State Quantum Key Distribution Protocol. Physical Review Letters, 94(4), 040503-1-040503-4, (2005).

[30] Scarani, V.,Acín, A.,Ribordy,G., Gisin, N.: Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations. Physical Review Letters, 92(5), 057901-1-057901-4, (2004).

[31] Inamori, H., Lütkenhaus, N., Mayers, D.: Unconditional Security of Practical Quantum Key Distribution. The European Physical Journal D, 41(3), 599-627, (2007).

[32] Gottesman, D., Lo, H.K., Lütkenhaus, N., Preskill, J.: Security of Quantum Key Distribution with Imperfect Devices. Quantum Information & Computation, 4(5), 325-360, (2004).

[33] Mayers, D.: Unconditional Security in Quantum Cryptography. Journal of the ACM, 48(3), 351-406, (2001).

[34] Lo, H.K., Lütkenhaus, N.: Quantum Cryptography: from Theory to Practice. Physics In Canada, 191-196, (2007).

[35] Mullins, J.: Making Unbreakable Code. IEEE Spectrum, 40-45, (2002).

[36] Zhao, F., Fu, M., Wang, F., Lu, Y., Liao, C., Liu, S.: Error Reconciliation for Practical Quantum Cryptography. Optik - International Journal for Light and Electron Optics, 118(10), 502-506, (2007).

[37] Chau, H.F.: Practical Scheme to Share a Secret Key through An Up to 27.6% Bit Error Rate Quantum Channel. Physical Review A, 66(6), 060302-1-060302-4, (2002).

[38] Biham, E., Boyer, M., Boykin, P.O., Mor, T., Roychowdhury, V.: A Proof of the Security of Quantum Key Distribution. Journal of Cryptology, 19(4), 381-439, (2006).

[39] Lo, H.K., Chau, H.F.: Unconditional Security of Quantum Key Distribution Over Arbitrarily Long Distances. Science, 283(5410), 2050-2056, (1999).

[40] Kraus, B., Gisin, N., Renner, R.: Lower and Upper Bounds on the Secret-key Rate for Quantum Key Distribution Protocols Using One-way Classical Communication. Physical Review Letters, 95(8), 080501-1-080501-4, (2005).

[41] Gottesman, D., Lo, H.K.: Proof of Security of Quantum Key Distribution with Two-Way Classical Communications. IEEE Transactions on Information Theory, 49(2), 457-475, (2003).

[42] Gong, C.Q., Zhou, H.Y., Feng, J.L.: Research on Reconciliation Algorithm in Quantum Key Distribution. The Ninth International Conference on Hybrid Intelligent Systems, vol. 1, pp. 496-498, (2009).

[43] Gong, C.Q., Zhou, H.Y., Feng, J.L.: An Improvement of Protocol Binary in Reconciliation of Quantum Key Distribution. International Conference on Management and Service Science, pp. 1-4, (2009).

[44] Yan, H., Peng, X., Lin, X., Jiang, W., Liu, T., Guo, H.: Efficiency of Winnow Protocol in Secret Key Reconciliation. World Congress on Computer Science and Information Engineering, pp. 238-242, (2009).

[45] Treeviriyanupab, P., Sangwongngam, P., Sripimanwat, K., Sangaroon, O.: Performance of ½-Rate Convolutional Code on Winnow Protocol for Quantum Key Reconciliation. International Symposium on Communications and Information Technologies, pp. 550-553, (2010).

[46] Hamming, R.W.: Error Detecting and Error Correcting Codes. The Bell System Technical Journal, 147-160, (1950).

[47] Rass, S., Kollmitzer, C.: Adaptive Error Correction with Dynamic Initial Block Size in Quantum Cryptographic Key Distribution Protocols. Third International Conference on Quantum, Nano and Micro Technologies, pp. 90-95, (2009).