P- ISSN 1991-8941 E-ISSN 2706-6703
2020,14(2): 95 - 103

Journal of University of Anbar for Pure Science (JUAPS)

Open Access

A Framework for QKD-based Electronic Voting

Ibrahim Saud Khaleel*, Sufyan T. Faraj Al-Janabi, and Ali Jbaeer Dawood
College of Computer Science and IT, University of Anbar, Ramadi, Iraq

ARTICLE INFO

Received: 14 / 10 /2020
Accepted: 31 / 10/ 2020
Available online: 1 / 12 / 2020

DOIhttp://dx.doi.org/10.37652/JUAPS.2020.14.2.18

Keywords:
E-voting,
Helios system,
Quantum key distribution,
One-time pad.

ABSTRACT

This paper deals with the security aspect of electronic voting (e-voting) by introducing quantum key distribution (QKD) to the e-voting process. This can offer an extremely high level of security that can be very beneficial for some significant e-voting tasks. Moreover, a framework for the integration of the QKD with the e-voting system is proposed. The Helios voting system, which is considered as one of the open-source and major voting systems, has been chosen for this integration. Investigation of the main design aspects of building a QKD-based e-voting system has been done. Thus, the expected advantages and limitations of the proposal are discussed and analyzed.

1. INTRODUCTION

Electronic voting (e-voting) is a term that means resorting to electronic means from computer hardware and software to carry out the electoral process procedures in all its stages or parts of them. There are many pros and cons of e-voting related to the Internet. Various systems of e-voting were suggested, but their use is still uncommon in the world. This is mainly due to the lack of confidence on the Internet among voters because it is possible that the system is being attacked from anywhere in the world [1]. Information technology is included in the electoral process, and in some cases, essential to conducting elections. Among the uses of information technology are preparing voter lists, managing and training staff, printing ballot papers, conducting voter education campaigns, recording votes, counting and collecting vote results, and publishing election results. Electronic voting (e-voting) is fairly secure, uses fewer resources, and easy to use. A lot of individuals can have the ability to access a system of e-voting from the public, business, or personal computers. This might be one of the solutions for that not much voter's turnout at polls [2].

Yet, there remains a debate if there is a possibility to conduct elections over the internet or online because of the extreme concerns of guaranteeing security and accuracy of e-elections. This is one of the challenges which are extending beyond security and safety as conventionally specified in computer science. However, former reviews regarding the systems of voting are suggesting that various mundane issues might be handled through utilizing formally verified and open source implementations.

The formal verification regarding the systems of voting down to the used software was especially difficult for some reasons [3]:

- Initially, to define the security characteristics of a voting system is still debated; while a lot of definitions were specified in simulation-based styles, the majority of attempts for formally verifying the cryptographic constructions are focused on game-based style.
- Secondly, the real-world adversary models concerning e-voting should consider the adversarial models going beyond the common view of the provable security, and responsible for the probability that the system of voting might be run in a corrupted environment or backdoored.
- Thirdly, the protocols sometimes have multiple variants, with slight, yet theoretically considerable differences in their security analysis.
- Lastly, the systems of e-voting were distributed, with multiple implementations regarding the clients of voting, providing more complexity for the reasoning regarding their implementation.

Internet voting is a famous case of e-voting. As indicated in Figure 1, internet voting is a major e-voting category that has two main types, uncontrolled or controlled environment. The latter indicates that the voting machines including (computers) were controlled via an election authority, while the former indicates that voters might be using their workplace, public, or personal computers for casting their votes [4].

This paper aims to investigate the benefits and limitations of integrating e-voting with quantum key distribution (QKD). This is achieved by proposing a framework for extending the functionality of the Helios e-voting system to include a one-time pad (OTP)

*Corresponding author at: College of Computer Science and IT, University of Anbar, Ramadi, Iraq. Tel.: þ964 7906200277. E-mail address: *ibrahem.abomusab@uoanbar.edu.iq*

encryption/decryption leveraging the cryptographic key obtained from QKD.

The remaining of this paper is organized as follows: Section 2 presents a literature survey on the related work. A general review of e-voting systems is given in Section 3 before the Helios e-voting system is explored in Section 4. Next, Section 5 is dedicated to the relevant QKD details. Then, the proposed framework for integrating QKD with e-voting is presented in Section 6. Finally, the paper is concluded in Section 7.

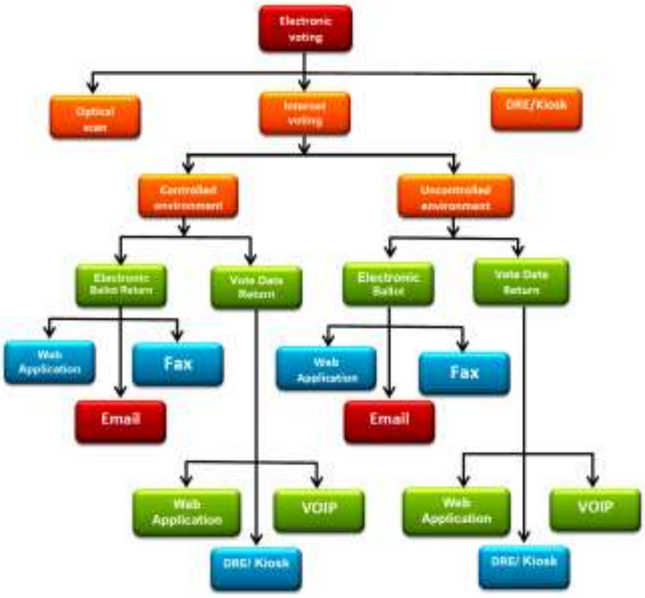


Fig: 1: Electronic voting categories [4].

2. RELATED WORK

This section provides the literature review for some related works in e-voting (especially related to Helios) and QKD. Many proposals for e-voting has been investigated previously. The security of these systems had been analyzed thoroughly. Various symmetric and asymmetric cryptographic algorithms had been explored in this respect [5]-[7]. However, it can be noted that almost all these works are relying on "non-quantum" cryptography.

2.1. Related Helios Voting Literature

Since introducing the first version of Helios in 2008 [8], many enhancements have been considered for it. Indeed, hundreds of thousands of people have used Helios in various organizational election tasks [9]. Karayumak et al proposed an enhancement of Helios interface for individual verifiability. The ease of use of the voting system after improvements has reached an acceptable level compared to the original version of Helios [10]. Bernhard et al studied the security aspect of Helios and provided an arithmetic security model for polling privacy. To avoid enforcing limitations by other methods of defining protocols, they used these cryptographic model games. They also analyzed an abstract version of Helios that follows the same infrastructure rather than direct analysis of the scheme during its implementation [11].

Cortier et al presented a fully distributed threshold encryption system (without a distributor) suitable for the Helios voting system and turned out to be safe under the Decisional Diffie-Hellman assumption [12]. Then, those authors also presented Helios-C which is a system similar to the Helios system in its simplicity. It is supposed to offer more strong verification capability and ballot privacy. It prevented a problem related to Helios which was the possibility of ballot stuffing [13].

Kulyk et al proposed the expansion of the Helios system to ensure participation privacy for voters and universal eligibility verifiability. The scheme also improved Helios towards receipt-freeness. They claimed that their proposed system can be used as an independent system or can be used to improve other schemes such as the Estonian voting scheme[14]. Moreover, Kulyk et al presented an expanded Helios voting system toward proxy voting. The proposed system maintained the requirements of the security of the original Helios system for the votes cast directly in addition to proxy voting [15].

Bernhard et al presented an extension of the Helios system, known as KTV-Helios. They aimed to use existing definitions of the privacy of the ballot and the possibility of verifying against the bulletin board. Besides, they provided an official definition of the receipt-freeness and the privacy of participation, which can be applied to KTV-Helios [16]. Meyer and Smyth showed that the Helios system could fail for ensuring that the representatives were selected via voters since opponents might result in a ballot except a voter's last to be counted. They also explained the way that an opponent might be choosing the ballots' content, and therefore the opponent might unduly impact choosing the representatives [17].

Abid and Al-Janabi [18,19] presented addressed some security and usability issues of Helios. The presented Helios++, which is an enhanced e-voting system based on Helios and public-key certificates. A certification authority (CA) had been added and integrated with Helios. Each voter had given one real account and other fake accounts to be used in case the voter is coerced. Indeed, the Helios interface had been improved and the Arabic language has been added to the system.

2.2. Related QKD Literature

In 1984, the first and most famous QKD protocol was suggested by C. Bennett and G. Brassard [20]. Thus, it is referred to as the BB84 protocol. The first experimental setup for implementing the BB84 protocol was built in 1989 [21]. The BB84 protocol allows secret keys to be generated between two parties over a public channel. According to the major concepts of quantum mechanics, an eavesdropper cannot gain information from intercepted qubits with no disturbance to their state.

Since that time, many enhancements, modifications, and mathematical proofs for QKD protocols appeared in the literature. Indeed, simulation has been effectively used by many researchers to study and analyze various aspects of these variations of QKD protocols [22]- [24]. The issue of authenticity of the parties involved in QKD is a crucial one. Therefore, the authentication aspects of QKD had been carefully studied. In [25], the researchers considered decreasing the costs of unconditionally-secure authentication. Concerning the phases which are constituting each one of the QKD sessions, they suggested two main authentication modes; "partial" authentication as well as "full" authentication modes. It was noticed that the full mode despite its ultimate security property can present an impact on the effectiveness of the QKD system.

Other previous works investigated the integration of QKD (or quantum cryptography protocols in general) with other already-implemented security and networking protocols and standards. For example, some researchers considered the performance analysis for a proposal that integrates QKD into IPSec [26]. Also, a scheme integrating QKD in 802.11i security mechanisms for the distribution of encryption keys was described in [27]. An extension of SSL/TLS that significantly facilitates the integration of QKD within the famous SSL/TLS web protocol was presented in [28]. Moreover, combining QKD to increase the communication’s security of a power system was proposed in [29].

Sabino analyzed Neuchatel's e-voting protocol. Accordingly, she proposed some improvements related to security and verifiability concerns. These improvements are expected to be achieved mainly by introducing some quantum cryptography techniques like QKD, quantum bit commitment, and multi-party computation to the original protocol [30]. However, the only quantum cryptography protocol that is known to have provable security so far is QKD. Thus, her work might only have some theoretical importance and cannot be reliably adopted into the practical side.

3. ELECTRONIC-VOTING SYSTEMS

E-voting is a tool to increase the efficiency of the electoral process and increase confidence in its management. If the e-voting is applied correctly, it can increase the poll’s security, the speed to announce results, and facilitating the process of voting. Yet, in the case when e-voting isn’t adequately designed, this might be undermining the confidence in the whole election process [31]. The System of e-voting must be based on the following requirements [4]:

- The system of e-voting should be available throughout the election.
- The system should be used easily.
- The system should be preventing voters from casting more than a single vote.
- The system should have the ability of verifying the voter.
- The system should have the ability of correctly counting the votes.
- Integrity after the election: - there must be no deletion, replacement, or removal of votes.

E-voting has a lot of benefits over conventional approaches of voting; a few of them are as follows [32]:

- Facilitating the direct polling process for people with special needs and sensitive business.
- Enabling residents abroad to participate in elections.
- Speed and accuracy in extracting results.
- Less cost and faster scheduling results in improving its accessibility and accuracy.

E-voting also has some disadvantages compared to the traditional way of voting. These need to be addressed carefully by researchers to develop the required countermeasures. Some of these disadvantages are [32]:

- The use of modern technology in the electoral process requires the provision of basic factors such as the presence of stable electrical energy and high economic potential.
- When using the e-voting system in the electoral process, the electronic system may fail or an error occurs in the design of the program.
- The possibility of electronic system exposure to piracy operations from abroad. Therefore, security services and mechanisms against these acts are needed.
- The ability to vote on behalf of others such as a family vote, as the head of the household owns the electronic cards on behalf of his family members.

4. THE HELIOS E-VOTING SYSTEM

Helios is one of the e-voting, web-based, and open-source systems majorly designed via Ben Adida [33]. The browser and font code are written in HTML and JavaScript, whereas the back-end server code was written in Python programming language. In addition, the Ballot Preparation System (BPS) is guiding voters through ballot as well as recording their choices. Also, the process used for creating ballot as well as processing the votes is on the basis of Benaloh's Simple Verifiable Voting Protocol.

In 2008, the Helios system was created. Trusting the server isn’t required due to the character regarding system work. Even if the system administrators were malicious, the process of voting is totally verifiable. A voter has the ability of verifying their vote which will be counted among final votes; thus, Helios creates individual verifiability. A voter can verify that all votes were correctly estimated and therefore the system is providing universal verifiability**,** while Helios was supposed for being verifiable, each one of the votes includes a smart ballot tracker that might be examined against Ballot Tracking Center for ensuring that the ballot is received as well as tallied adequately [33].

4.1. Server Architecture

Helios back-end can be defined as one of the web applications which are written in Python, run in the application server of Cherry Py 3.0, with a Lighttpd web server. Also, all the data stored in the Postgre SQL database. Also, all the server-side logic was carried out in Python, with the templates of HTML which are rendered utilizing the Cheetah Templating engine. A lot of back-end API calls return the JSON data structures utilizing Simple JSON library, also the voting booth server-side template was single page web application involving JavaScript logic as well as Template HTML/JavaScript templates. Considering the application software, Python Cryptography Toolkit was utilized for some theory utilities including the random number and prime number generation [8].

4.2. Creating Elections

Elections can be created just by registered Helios users. The process of registration will be handled in a way comparable to most common web sites in the following way [8]:

- Users entering the e-mail address, required password, and name.
- An e-mail with the embedded confirmation link will be sent to a certain e-mail address.
- A user will click on a confirmation link for activating his/her account.

After that, registered users will be creating an election with the election name, time and date when voting will start and when it is expected to end. Following creation, Helios will be generating and storing new El Gamal keypair for election. Just the public key is provided to registered users: Helios will be keeping the key secret; an administrator is a user who formed the election. Administrative user has the ability of adding, updating, and removing voters instantly. A voter can be recognized via an e-mail address and a name particular for a certain election [8].

4.3. Strengths and Weaknesses of Helios

The Helios voting system has many strengths that made it one of the best and most popular e-voting programs, some of which can be mentioned as follows:

- The Helios system is fully open-source and allows end-to-end verification [10].
- Trusting in the server is not required because the Helios voting process is fully verifiable [34].
- Encryption is done using JavaScript, so the user can even disconnect the computer from the Internet after downloading all credentials, making its options, encrypting the vote, and reconnecting the Internet to the vote [35].
- All encrypted votes are shown on the bulletin board. The Helios system achieves the Ballot secrecy [36].
- The bulletin board permits only one vote to link with an identity [12].

Despite the strengths of the Helios -system, it also has many weaknesses, including:

1. Helios does little to save voters from coercion [9].
2. Helios does not do much to counteract the threat of a web browser or client-side operating system compromise [37].
3. Helios can be accessed over the Internet, making it susceptible to attacks such as a denial of service attacks [35].
4. Anyone can know who has voted whether the real name or the nickname and that's because the bulletin board is public [38].
5. In the future, if the encryption algorithms used in Helios are broken, the attacker will be able to decrypt all votes [39].
6. Helios only aims to achieve the privacy of the ballot and clearly, ignores the concepts of confidentiality in favor of efficiency [11].

5. QUANTUM KEY DISTRIBUTION (QKD)

The quantum key distribution purpose is for Bob and Alice, initially sharing no private information, to have a shared random key which stays unknown to Eve. This shared secret key then can be subsequently used, as in Vernam or the one-time-pad (OTP) encryption, to send important secret messages whenever needed. Note that it is also possible to use it with other secret-key algorithms, such as DES or AES.

Bob and Alice utilize a quantum channel for sending polarized photons, and a conventional public channel, over which they make public discussion. Eve has the freedom of making measurements on the quantum channel photons one at a time. Furthermore, Eve becomes aware of all the message contents which are sent over that public channel with no ability to interfere with such public transmissions. Bob and Alice utilized the public channel for discussing and comparing signals that have been transmitted over the quantum channel, testing them for eavesdropping [40]. This is depicted in Figure 2.

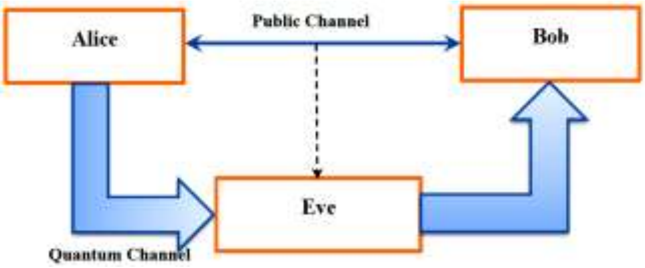


Fig:2: A schematic for a quantum cryptosystem.

Often, QKD is inadequately referred to as quantum cryptography, since it is the most effective example of a quantum cryptographic task that has provable security. The results from fundamental aspects related to quantum mechanics: The procedure to measure the quantum system generally disturbs the system. Hence, a third party attempting to eavesdrop on the key should be measuring it, thus providing detectable anomalies [24]. QKD is utilized for producing and distributing the key, not for transmitting message data. The key might be utilized with any selected algorithm of encryption for encrypting (and decrypting) the message, that might be after that transmitted over a standard communication channel [40].

In the following subsections, the basic procedure for QKD is described. The description is based on the Bennet-Brassard protocol and assumes using the polarization states of single photons for the quantum channel. An eavesdropper (Eve) is assumed conservatively to have unlimited technology (consistent with quantum physics) to deal with single photons. Thus, the QKD protocol can be divided into the following five stages: Initialization, quantum transmission, error elimination, estimating Eve's information, and privacy amplification [40]. Note that to maintain simplicity, the authentication issue of the public channel is not considered in this description. More details on this point can be found elsewhere [25].

5.1. Initialization

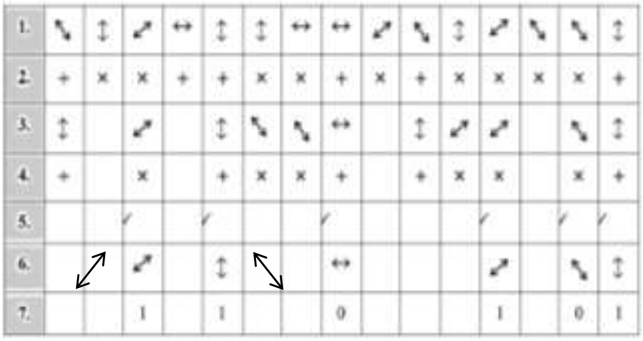
In the beginning, Alice sends a *request* message to Bob utilizing a public channel. In this message, Alice tells Bob that she wants to start a QKD protocol with him. She may also specify the clock period at which she wants to start the quantum transmission. Then, she waits for an *acknowledgment* message on the public channel from Bob. These request and acknowledgment messages can be of different formats depending on the type of application. It is possible that more than one message of request and/or acknowledgment must be exchanged in a few situations.

Concerning such stage, Alice and Bob might adapt to a protocol to convert polarization states into bits. This might be interpreting bit 'O' for 0^o and 135^o polarizations, and a bit '1' for 90^o and 45^o polarizations

5.2. Quantum Transmission

This stage can be described according to the BB84 protocol via the following steps (See Figure 3):

1- A random sequence of four canonical types of polarized photons is sent via Alice to Bob.
2- Independently and randomly Bob chooses each one of the photons for measuring the diagonal or rectilinear polarization of photons.
3- The type of measurement made by Bob will be publicly announced (yet not the result of measurement).
4- Alice will publicly tell Bob that he made the right measurements (diagonal or rectilinear).
5- Bob and Alice are publicly agreeing to discard all the bits positions for which bob made the wrong measurement. Also, they are agreeing to discard the bit positions in which Bob detector failed in detecting photons at all.
6- The polarizations related to remaining photons were interpreted as a bit 'O' for 0^o and 135^o polarizations, and a bit '1' for 90^o and 45^o polarizations.



1.	⤡	↕	⤢	↔	↕	↕	↔	↔	⤢	⤡	↕	⤢	⤡	⤡	↕
2.	+	×	×	+	+	×	×	+	×	+	×	×	×	×	+
3.	↕		⤢		↕	⤡	⤡	↔		↕	⤢	⤢		⤡	↕
4.	+		×		+	×	×	+		+	×	×		×	+
5.			✓		✓			✓				✓		✓	✓
6.		⤢	⤢		↕	⤡		↔				⤢		⤡	↕
7.			1		1			0				1		0	1

Fig: 3: The basic quantum transmission stage [40].

5.3. Error Elimination

Having completed the above steps, Alice and Bob now possess the so-called raw quantum transmission (RQT). Then, the error elimination steps can be as follows [21]:

1. Alice and Bob must have an agreement on a random bit positions' permutation in their strings for the randomization of the error locations.
2. The strings that have been permuted are then partitioned to size *k* blocks, in a way that the single blocks are considered to not likely include more than a single error. The choice of the suitable value of *k* will be based on an empirical formula concluded from the experimentation. It should be noted that a small random sample of the bits may be initially compared for the estimation of the error rate. Certainly, these bits would then have to be sacrificed.
3. For every one of those blocks, Alice and Bob perform a comparison of the parity of the block (Even parity coding). Blocks that have matching parity will be tentatively expected to be correct.
4. Blocks of the discordant parity are subjected to bisective searching, which is performed to disclose the log(*k*) further sub-block parties, to the point of finding the error and correcting it.
5. Alice and Bob should have an agreement for discarding the final bit of every one of the blocks or sub-blocks whose parity has been disclosed. Thus, they avoid leaking the information to Eve throughout the process of error elimination.

5.4. Estimation of Eve's Information

This stage aims to estimate the knowledge that Eve obtained in the previous steps. When using actual one-photon pulses for the transmission of the bits of the key on the quantum channel, Eve will not be capable of using the strategy of beam-splitting for eavesdropping on the transmission of the quantum channel. In this case, the main strategy of eavesdropping which is possible for Eve is the strategy of interception/resending. Thus, the amount of the bits (*w*) which have been leaked to Eve may conventionally be calculated. More details can be found in [21].

5.5. Privacy Amplification

Assuming that $\boldsymbol{x}$ is a string that is produced from the stage of error elimination, and $\boldsymbol{x}$ length is n bits. Therefore, to perform the approach of the privacy amplification, a publicly selected random hash function from a suitable class is implemented on $\boldsymbol{x}$ which results in a string $h(\boldsymbol{x})$. The length of $h(\boldsymbol{x})$ may be specified to be $n–w–s$ bits, where w represents the calculated number of the bits that have been leaked to Eve, and s represents a random parameter of safety which is higher than 0.

For calculating the hash function required for performing privacy amplification, at the start the length of the final string ($n–w–s$) is computed. After that, $n–w–s$ independent arbitrary sub-sets are selected publicly. The selected sub-set parities are computed and kept private for the construction of required $h(\boldsymbol{x})$, which is a suitable Carter-Wegman hash function. This calculated hash function represents the shared secret key of cryptography [40].

6. THE PROPOSED FRAMEWORK

The proposed QKD-based e-voting framework is based on a three-tier architecture, as shown in Figure 4. From bottom-up, these tiers are:

1. *The QKD Tier:* This is responsible for performing all the stages and details of the QKD protocol including, quantum transmission, error elimination, and privacy amplification.
2. *Encryption and Synchronization Tier:* This tier is mainly responsible for two main tasks. The first is performing symmetric encryption using the secret key obtained from QKD. To maintain the highest level of security, the OTP will be assumed as the default encryption algorithm because OTP is the only known cipher that can achieve perfect secrecy. Using other symmetric ciphers like AES or DES will be provided as an option. The second task of this tier is the required functionality to synchronize the secret keys obtained from QKD between the peer entities. This functionality is necessary for the efficient use of these keys.
3. *The e-Voting Tier:* This upper tier is where the user will be interacting to perform various election tasks. In this process, the voter casts his/her vote from the place where he/she is, whether it is his/her home or his/her workplace through the voting program, and upon choosing what suits him/her, he/she sends the (encrypted) vote to the private polling station for the elections. The Helios-voting system is assumed to be used in this tier; however, basically any other e-voting system can also be used.

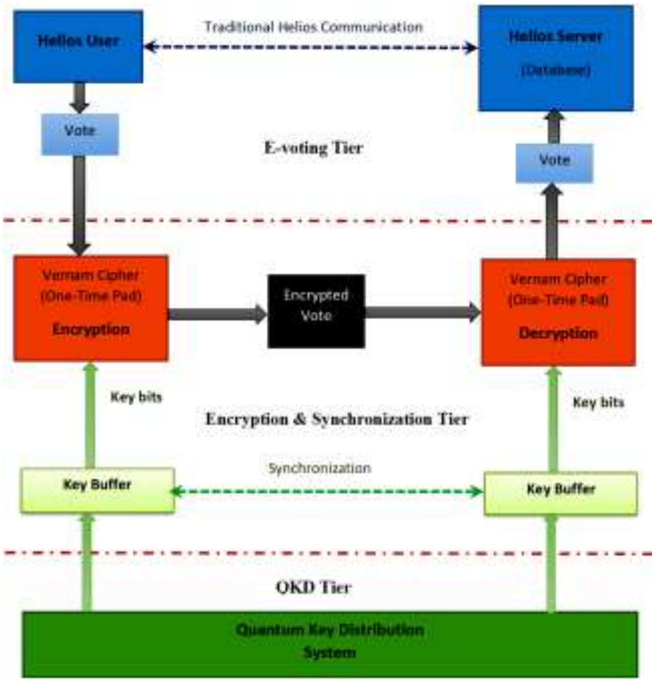


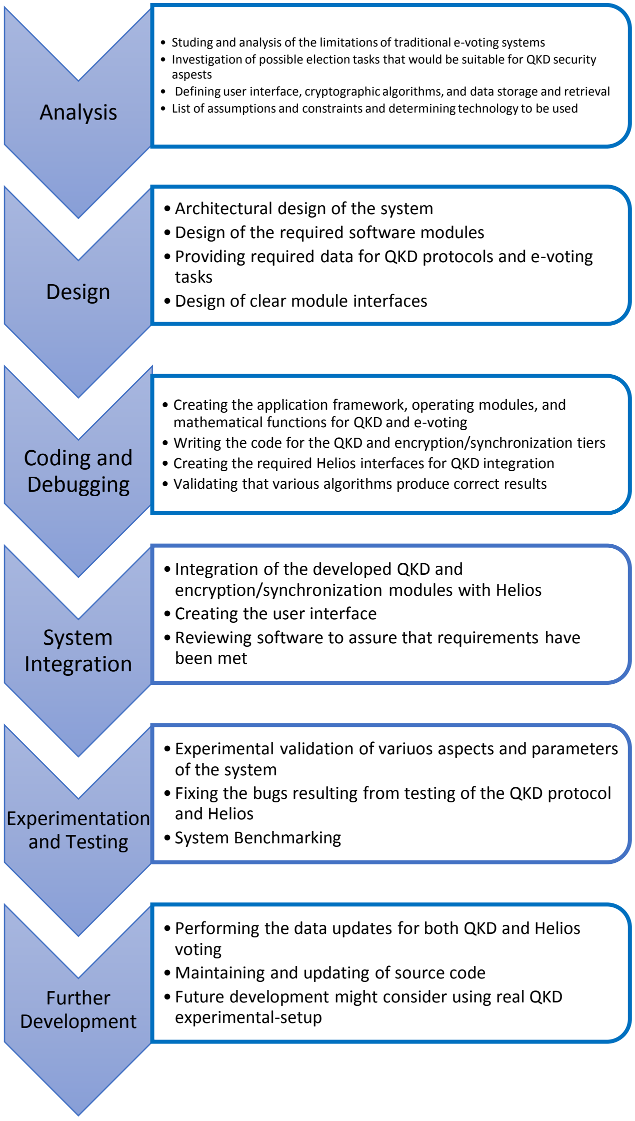
Fig: 4: Simplified architecture of the proposed QKD-based e-voting system.

The functionality of the Helios system will be extended by including additional modules required for the efficient and secure integration of the QKD protocol. Thus, the use of symmetric encryption based on QKD will be provided as an additional option for Helios users. Therefore, various modules related to QKD, key-synchronization, and symmetric encryption are needed to be added to Helios. Software simulation using Python is to be used for the simulation of the QKD protocol and to achieve its integration with Helios. This approach is totally prudent to investigate various aspects and limitations of this integration before moving into the use of the actual QKD apparatus in the future.

The System Development Life Cycle (SDLC) of this proposal consists of the following phases: Analysis, design, coding and debugging, system integration, experimentation and testing, and further development. This SDLC provides the implementation guidance for this software project. The main tasks of each phase are outlined in Figure 5. Despite that this work is mainly a software project focusing on using QKD software simulation, future development may also consider Helios integration with real QKD set-ups.

Referring to Section 5, the simulation of the basic steps of the QKD protocol and symmetric encryption is shown by Algorithm 1. It can be noted that when there is severe eavesdropping on the quantum channel, the whole process of QKD needs to be restarted. Otherwise, the QKD can be repeated again and again till producing all the secret key bits required for symmetric encryption. After establishing the shared secret key of encryption, the encrypted messages may be transmitted on public channels.

y



Analysis
- Studing and analysis of the limitations of traditional e-voting systems
- Investigation of possible election tasks that would be suitable for QKD security aspests
- Defining user interface, cryptographic algorithms, and data storage and retrieval
- List of assumptions and constraints and determining technology to be used

Design
- Architectural design of the system
- Design of the required software modules
- Providing required data for QKD protocols and e-voting tasks
- Design of clear module interfaces

Coding and Debugging
- Creating the application framework, operating modules, and mathematical functions for QKD and e-voting
- Writing the code for the QKD and encryption/synchronization tiers
- Creating the required Helios interfaces for QKD integration
- Validating that various algorithms produce correct results

System Integration
- Integration of the developed QKD and encryption/synchronization modules with Helios
- Creating the user interface
- Reviewing software to assure that requirements have been met

Experimentation and Testing
- Experimental validation of variuos aspects and parameters of the system
- Fixing the bugs resulting from testing of the QKD protocol and Helios
- System Benchmarking

Further Development
- Performing the data updates for both QKD and Helios voting
- Maintaining and updating of source code
- Future development might consider using real QKD experimental-setup

Fig:5: The proposed system development life-cycle.

In the case of using encryption algorithms like the AES, the number of the required key bits are minimal. However, to achieve maximum security, the OTP cipher is utilized for the procedures of encryption and decryption. The message (vote) will be represented in the binary form. After that, it will be XORed with the bits of the secret key in the transmitter for providing ciphertext. This will be transmitted on the public channel to reach the recipient, where the ciphertext will be X-ORed again with the same bits of the secret key. Thereby, it results in the reproduction of the original message.

```
Algorithm 2: The block parity comparison process.
Input: The raw quantum transmission (RQT)
Output: Sifted bits that Eve has some knowledge on
Start
Step 1: Initialize  i ←1
            // i represents the number of required iterations
                     j ←1
            // j is related to the number of blocks
Step 2: For all i do {
            Random permutation of RQT
            Calculate Block size
Step 3: For all j do {
            Compare blocks' parities
Step 4: If agreed parities, go to the next step
            Else, { do the bisective search
                    Discard error bit }
Step 5: If j > Number of blocks, go to the next step
            Else, go to Step 3 }
Step 6: If iterations completed, go to End
            Else, go to step 2 }
End
```

```
Algorithm 3: The privacy amplification procedure.
Input: Shared sifted bits that Eve has some knowledge on
Output: Shorter secret key that Eve has almost no knowledge on
Start
Step 1: Calculate the final string length (n–w–s), where
            n is the length of the string produced from error
            elimination, w represents the calculated number
            of the bits leaked to Eve, and s represents a
            positive safety parameter.
            Initialize i ← 1
Step 2: for all i do {
            Choose randomly subset independently
            Calculate the subset parity required for the
            construction of the Carter-Wegman hash function
            h(x)                          // keep it secret
Step 3: If i > string length, go to End
            Else, go to Step 2 }
End
```

Referring to subsections 5.3 and 5.5, the simulation of the block parity comparison process in the error elimination stage and the main procedure for privacy amplification is described by algorithms 2 and 3, respectively. The parity comparison process is the main step in elimination the RQT errors according to the adopted protocol. Indeed, the privacy amplification is a crucial technique to almost eliminate all of Eve's knowledge about the produced secret key.

7.CONCLUSION

Security can be the main concern regarding e-voting systems. Thus, in this paper, a framework that enables the integration of QKD within the e-voting process has been proposed. This can enable e-voting to make the benefit of the ultimate security offered by QKD. Privacy amplification based on a suitable class of Carter-Wegman hash function is used to effectively eliminating Eve's partial information about Alice and Bob's key. To maintain the highest security level of the system, OTP encryption is suggested for its provable secrecy property. Using other block ciphers encryption is provided as an additional option for the Helios users. To the best of our knowledge, this is the first framework proposal for designing an e-voting system based on QKD (without relying on any other unproven quantum cryptography protocols). We believe that this proposal can be applied in critical small to medium-scale electoral tasks.

It is planned to present more experimental and simulation results related to this work in a subsequent paper.

REFERENCES

[1] Zagórski F., Carback R., Chaum D., Clark J., Essex A., and Vora P., 2013, "Remotegrity: Design and use of an end-to-end verifiable remote voting system," International Conference on Applied Cryptography and Network Security, 441-457.

[2] Kaczmarek T., Wittrock J., Carback R., Florescu A., Rubio J., Runyan N., et al., 2013, "Dispute resolution in accessible voting systems: The design and use of audiotegrity," International Conference on E-Voting and Identity, pp. 127-141.

[3] Cortier V., Drăgan C., Dupressoir F., Schmidt B., Strub P., and Warinschi B., 2017, "Machine-checked proofs of privacy for electronic voting protocols," IEEE Symposium on Security and Privacy (SP), 993-1008.

[4] Ahmed M., and Abo-Rizka M., 2013, "Remote Internet voting: security and performance issues," World Congress on Internet Security (WorldCIS-2013), 56-64.

[5] Al-Janabi S., and Abid N., 2019, "Security of Internet Voting Schemes: A Survey," Revista AUS Journal, Special Issue 26-2, 260-270.

[6] Naidu P., and Kharat R., 2016, "Multi-factor authentication using recursive xor-based visual cryptography in online voting system," International Symposium on Security in Computing and Communication, 52-62.

[7] Imbar R, and Tirta E., 2007, "Analysis, Design and Implementation of Lubricant Sales Information Systems Case Study: Company PT Pro Roll International", Jurnal Informatika, 3, 119-149.

[8] Adida B., 2008, "Helios: Web-based Open-Audit Voting," USENIX Security Symposium, 335-348.

[9] Pereira O., 2016, "Internet voting with Helios," Real-World Electronic Voting: Design, Analysis and Deployment, 8604.

[10] Karayumak F., Kauer M., Olembo M., Volk T., and Volkamer M., 2011, "User study of the improved Helios voting system interfaces," 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST), 37-44.

[11] Bernhard D., Cortier V., Pereira O., Smyth B., and Warinschi B., 2011, "Adapting Helios for provable ballot privacy," in European Symposium on Research in Computer Security, 335-354.

[12] Cortier V., Galindo D., Glondu S., and Izabachene M., 2013, "Distributed elgamal á la pedersen: application to helios," in Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, 131-142.

[13] Cortier V., Galindo D., Glondu S., and Izabachene M., 2014, "Election verifiability for helios under weaker trust assumptions," in European Symposium on Research in Computer Security, 327-344.

[14] Kulyk O., Teague V., and Volkamer M., 2015, "Extending helios towards private eligibility verifiability," International Conference on E-Voting and Identity, 57-73.

[15] Kulyk O., Marky K., Neumann S., and Volkamer M., 2016, "Introducing proxy voting to Helios," 11th International Conference on Availability, Reliability, and Security (ARES), 98-106.

[16] Bernhard D., Kulyk O., and Volkamer M., 2017, "Security proofs for participation privacy, receipt-freeness and ballot privacy for the helios voting scheme," Proceedings of the 12th International Conference on Availability, Reliability and Security, 1-10.

[17] Meyer M., and Smyth B., 2019, "Exploiting re-voting in the Helios election system," Information Processing Letters, 143, 14-19.

[18] Abid N., and Al-Janabi S., 2019, "A Framework for I-Voting based on Helios and Public-Key Certificates," Revista AUS Journal, Special Issue 26-2, 234-243.

[19] Abid N., and Al-Janabi S., 2020, “The secure I-voting system Helios++,” International Journal of Computing and Digital Systems, University of Bahrain (Published online on 25/7/2020).

[20] Bennett C., and Brassard G., 1984, “Quantum cryptography: public key distribution and coin tossing,” International Conference on Computers, Systems and Signal Processing, India, 175-179.

[21] Bennett C., Bessette F., Brassard G., Salvail L., and Smolin J., 1991, “Experimental quantum cryptography,” J. of Cryptology, 5, 3-28.

[22] Sharifi M., and Azizi H., 2007, "A simulative comparison of BB84 protocol with its improved version," Journal of Computer Science & Technology, 7, 87-95.

[23] Qiao H., and Chen X., 2009, "Simulation of BB84 Quantum Key Distribution in depolarizing channel," Proceedings of 14th Youth Conference on Communication, 123-129.

[24] Al-Janabi S., and Hashim R., 2011, “Key reconciliation techniques in quantum key distribution", Proceedings of the First Engineering Conference of the College of Engineering, University of Anbar, 20-27.

[25] Al-Janabi S., and Jasim O., 2011, "Reducing the Authentication Cost in Quantum Cryptography," The 12th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2011), UK, 363-368.

[26] Sfaxi M., Ghernaouti-Helie S., and Ribordy G., 2005, "Using quantum key distribution within IPSec to secure MAN communications", Proceedings of the IFIP-MAN 2005 Conference on Metropolitan Area Networks, Vietnam, 255-260.

[27] Nguyen T., Sfaxi M., and Ghernaouti-Helie S., 2006, "802.11i encryption key distribution using quantum cryptography," Journal of Networks, 1(5), 33-40.

[28] Al-Janabi S., 2008, “A novel extension of SSL/TLS based on quantum key distribution”, Proceedings of the International Conference on Computer and Communication Engineering (ICCCE08), Malaysia, I, 919-922.

[29] Zhou J., Lu L., Lei Y., and Chen X., 2014, "Research on improving security of protection for power system secondary system by quantum key technology," Power Syst. Technol., 38, 1518-1522.

[30] Sabino M., 2016, “Quantum Cryptography applied to Electronic-Voting Protocols,” M.Sc. Thesis/ Mathematics and Applications, IST, Tecnico, Lisboa.

[31] Qadah G., and Taha R., 2007, "Electronic voting systems: Requirements, design, and implementation," Computer Standards & Interfaces, 29, 376-386.

[32] Cortier V., Galindo D., Küsters R., Mueller J., and Truderung T., 2016, "Sok: Verifiability notions for e-voting protocols," IEEE Symposium on Security and Privacy (SP), 779-798.

[33] Frank M., Halderman J., Aschermann C., Adida B., Holz T., Hu S., et al., 2020, "An Ever-evolving Game: Evaluation of Real-world Attacks and Defenses in Ethereum Ecosystem," 29th {USENIX} Security Symposium, 2020.

[34] Bokslag W., and de Vries M., 2016, "Evaluating e-voting: theory and practice," arXiv preprint arXiv:1602.02509, 2016.

[35] Schneider A., Meter C., and Hagemeister P., 2017, "Survey on remote electronic voting," arXiv preprint arXiv:1702.02798, 2017.

[36] Quaglia E., and Smyth B., 2018, "Authentication with weaker trust assumptions for voting systems," International Conference on Cryptology in Africa, 322-343.

[37]Adida B., De Marneffe O., Pereira O., and Quisquater J., 2009, "Electing a university president using open-audit voting: Analysis of real-world use of Helios," EVT/WOTE, 9, 120-128.

[38] Volkamer M., Spycher O., and Dubuis E., 2011, "Measures to establish trust in internet voting," in Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance, 1-10.

[39] Chang-Fong N., and Essex A., 2016, "The cloudier side of cryptographic end-to-end verifiable voting: a security analysis of Helios," Proceedings of the 32nd Annual Conference on Computer Security Applications, 324-335.

[40] Al-Janabi S., 1999, "Quantum Cryptographic Key Distribution in Optical Communication Networks," Ph.D. Thesis, College of Engineering, Al-Nahrain University, Baghdad.

إطار عمل للتصويت الإلكتروني المبني على التوزيع الكمي لمفاتيح التشفير

إبراهيم سعود خليل و سفيان تايه فرج و علي جبير داود
كلية علوم الحاسوب وتكنولوجيا المعلومات ، جامعة الأنبار ، رمادي، العراق

الخلاصة:

يتناول هذا البحث الجانب الأمني للتصويت الإلكتروني من خلال ادخال التوزيع الكمي لمفاتيح التشفير في العملية. وهذا من شأنه أن يوفر درجة عالية من الأمنية التي تتطلبها بعض مهام التصويت الحساسة. وقد تم اقتراح إطار عمل لدمج التوزيع الكمي لمفاتيح التشفير ضمن عملية التصويت الإلكتروني. وقد تم اختيار نظام هيليوس للتصويت الإلكتروني المفتوح المصدر لهذا الغرض. وقد تم تقصي الجوانب الرئيسية لتصميم منظومة التصويت الإلكتروني المبني على التوزيع الكمي لمفاتيح التشفير. وبناء عليه مناقشة وتحليل الفوائد المتوقعة والمحددات لهذا النوع من المنظومات.