International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Jan-2021)
http://dx.doi.org/10.12785/ijcds/100108

The Secure I-Voting System *Helios++*

Noor Hamad Abid[1] and Sufyan T. Faraj Al-Janabi[1]

[1]*College of Computer Science and IT, University of Anbar, Ramadi, Iraq*

Received 11 Mar. 2020, Revised 24 Apr. 2020, Accepted 25 Jul. 2020, Published 1 Jan. 2021

Abstract: Voting is the process through which representatives of the country (or an organization) are chosen. However, some voters may not go to the polls to vote for personal or public reasons. One solution to this problem is Internet voting (I-voting) where it can be voted from anywhere and anytime. I-voting has many advantages and certainly, there are disadvantages. Many I-voting systems have been proposed. Helios, an open-source system, is one of the most popular voting systems.
This paper presents Helios++, which is an enhanced I-voting system based on Helios and public-key certificates. Improvements to the Helios system have been proposed. A certification authority (CA) has been added and integrated with Helios. This authority creates voter certificates containing public and private keys that are used later in the voting process, where they are used for encryption and generating digital signatures. Signing the votes can be done by either the RSA or DSA algorithms. Indeed, each voter has given one real account and other fake accounts to be used in case the voter is coerced. Finally, the Helios interface has been improved and the Arabic language has been added to the system.

Keywords: Internet Voting, Certificate Authority, Helios, Multi-Accounts, Public Key Certificates.

1. INTRODUCTION

Elections are a formal collective decision-making process in which the population chooses a person to hold public office. There are many ways to vote, including paper and electronic voting. The world has begun to turn to electronic voting because this method is easier and faster than traditional voting. Internet voting (I-voting) is one of the methods of electronic voting through which it can be possible to increase the rate of participation because the voters can vote from any place or device and also the results are accurate and calculated in a short time [1].

Many I-voting systems have emerged (For example, see [2], [3], [4], and [5]). However, the Helios voting system [6] is one of the most relevant I-voting tools to date [7]. The Helios system has gained fame because it is open source, permits end-to-end verification, does not require the installation of any software, and everyone has the right to vote only once. Despite these advantages, the Helios system suffers from being conducted only in a low-coercive and small-scale environment, it takes no action to prevent or reduce coercion, and inexperienced voters have difficulty voting because the commands are a bit difficult and complicated [8], [9].

Helios can be considered the cornerstone in open-source I-voting. Thus, it has been treated as one of the main references for developing new I-voting systems. Hence, several improvements have been proposed to the Helios system in various aspects.

V. Cortier et al. analyzed the secrecy of the ballot in the Helios system. A security gap had been discovered that allows the adversary to violate voter privacy. The gap exploits the lack of independence of the ballot and works by replaying the voter ballot, which amplifies the voter's contribution to the election result, and this magnification can violate privacy. A solution to this problem was presented and it was found that this solution can meet the official definition of ballot secrecy using the applied calculus [10].

D. Chung et al. provided the Helios system with multi-servers' capability. These servers communicate via the Paxos protocol, a protocol for distributed environments that bear errors. The measure comes to reduce the risk of denial of service attacks. In the event of an attack on one server and disabling it, it would still be possible to vote from other servers [11]. S. Srinivasan et al. noticed that an insecure server in Helios can stuff ballot and this seems to limit claims of the system of end-to-end verification. It was investigated how ballot stuffing could be addressed with a minimum change in the current voting experience in Helios. The ideas presented are general and intuitive enough to be applied in the context of the other I-voting systems. They also addressed some recent attacks that exploit the ballots' malleability in Helios [12].

M. Meyer et al. showed that Helios does not meet the strong non-reusability because the opponent can cause a ballot other than that of the voter. In particular, the opponent can intercept the authorization token associated with the ballot that the opponent wants to count, wait until the voter broadcasts his last ballot, and then release the

E-mail: hopingsmile9@gmail.com, sufyan.aljanabi@uoanbar.edu.iq

intercepted token. The issued token causes the bulletin board to accept the opponent's ballot and archive the voter ballot. They also showed that the opponent could choose the contents of these ballots. Thus, opponents can unduly influence the choice of representatives [13]. Other proposals for Helios improvements can be found in [14], [15], and [16].

Evaluation studies concluded that Helios is a very useful I-voting system in several aspects. However, they also shown some important practical implications related to the use of Helios. Therefore, existing shortcomings in coercion resistance, usability, and some security issues advise for a gradual approach, in which I-voting plays an increasing role is the most desirable option [7], [17]. Therefore, this work introduces a new version of Helios that avoids some limitations of the previously proposed versions.

In this paper, Helios++, an I-voting system based on the public key infrastructure and the Helios system is presented. The main objective of the proposed system is to improve the Helios system in terms of both security and usability, especially for Arabic-speaking people. Some other theoretical related aspects to Helios ++ can be found elsewhere [18], [19]. The remaining of the paper is organized as follows: More information about the Helios system is presented in Section 2. Then, the design and implementation aspects of the Helios++ system are explained in sections 3 and 4, respectively. In Section 5, the experimental results are presented and discussed. Finally, the paper is concluded in Section 6.

2. THE HELIOS VOTING SYSTEM

Helios system is an open-source web-based I-voting system that was released in 2008. Trusting a server isn't needed because of the character of the system's work. Even if administrators of the system are malicious, the voting process stays fully verifiable. The voter can verify that her/his vote has been counted among the final votes, so the Helios system provides individual verifiability. The voter can also verify that all votes have been calculated correctly and thus the system provides universal verifiability [20].

Helios system uses a third party to validate the votes, enough for eligibility, assuming that the third party is honest. The validated ballots are listed along with the voters' identities, and a maximum of one ballot is listed next to each identity, enough to not be reusable. Other ballots are archived. Auditing is used to determine whether unauthenticated ballots are listed, or if non-voter ballots are authenticated incorrectly by a dishonest third party [21]. Figure 1 shows the Helios voting protocol.

The voting steps in the Helios system are [22], [23]:

1. Registration to the system.
2. Click on "Vote in this Election" link and go to the "Voting Booth" page.
3. Read instructions on how to vote and then answer election questions.
4. Press the "Proceed" button and go to the next page.
5. Review voting, and also there are three options.
   a) Click on the "edit responses" button and return to the "Voting Booth" page.
   b) Press the "Submit" button and go to the "Submit Box" page.
      - If you press the "Cancel" button you will be taken to the Helios homepage.
      - If you press the "Cast this ballot" button, you will be taken to the confirmation page and then to the Helios homepage.
   c) Click the "Verify Encryption" button and go to the "Helios verifier" page.
      - If you click on the "Ballot Verifier" link, you will be taken to a verification page, the voting will be verified, and then back to the Helios homepage.
      - If the "back to voting" button is clicked, you will be returned to the "Voting Booth".

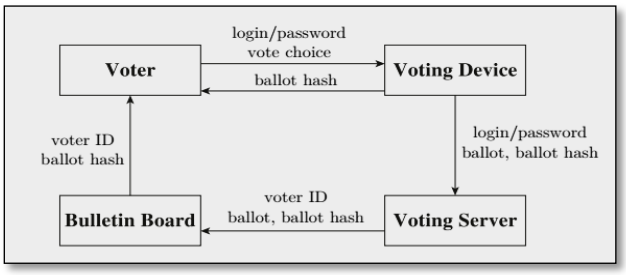

Figure 1. Helios voting protocol [22].

3. DESIGN ASPECTS OF THE PROPOSED HELIOS++ SYSTEM

The proposed Helios++ system includes improvements to the Helios voting system in four main areas: Security, scalability, anti-coercion, and usability. The suggestion of these improvements has come after studying the Helios system and showing its weaknesses that make it less used by voters. These improvements can make Helios++ safer and more widely used.

The main voting steps of the Helios++ system are:

1. Registration to the system by either Google, Facebook, Yahoo, or LinkedIn account.
2. Press the "Arabic" button and convert the language into Arabic.
3. Click the "Certification Authority" button and go to the Certification Authority (CA) page.
   a) Press the "Create Certificate" button and then the voter certificate is automatically generated.

b) Click on the "Download Certificate" button and the voter certificate will be downloaded to the voter device.

c) Click on the "Download Public Key" button and then the Helios Public Key is downloaded to the voter device

d) Click on the button "Verify Helios" and then go to a page to verify the validity of the Helios certificate.

4. Sign Vote (RSA or DSA) using keys in the certificate.
5. Encrypt Vote using keys in the certificate.

There are two basic sides of the proposed system: The server-side and the client-side. On the client-side, the voting process is conducted and the vote is encrypted. Voters can vote even if they are not connected to the Internet, thereby reducing the chances of attack. After the vote, voters can reconnect to the Internet and send the vote to the server. On the server-side, decryption, counting, and announcing results are performed. The proposed system is designed to include both server and client sides.

A. Security and Scalability

Helios++ incorporates a public-key CA to create the required cryptographic keys. Public keys are created and connected to the voter. The keys are used for encrypting and signing votes. Adding the CA to produce the keys for encryption and digital signature dramatically increases system security by enabling advanced security services to protect the system and data. Indeed, the scalability of the Helios++ system can be increased in a distributed environment compared to the other typical deployment of the Helios system. Figure 2 illustrates integrating the CA in the proposed system.

B. Coercion Resistance

The Helios++ system enables a voter to have more than one account to vote. Each voter has a “real” account (based on his/her choice) used to vote and other “fake” accounts that can be used by the voter when she/he is subjected to coercion. During the voting process, if the voter registers in her/his “real” account, she/he can vote in the elections and her/his vote is placed in the main database and is therefore counted. If a voter is coerced, she/he will be able to use one of her/his “fake” accounts to vote as the attacker wants. Voting, in this case, is placed in the secondary database, so her/his vote will not be counted within the final result. This is illustrated in Figure 3.

Note that we use the terms “real” and “fake” accounts to mean that the voter during the registration (or later) tells the system to count the vote coming from login based on the “real” account and to ignore any votes coming from login based on other “fake” accounts. Thus, “real” and “fake” are merely related to the voting process and cannot be extended to other meanings since all the accounts are actually belonging to that voter.

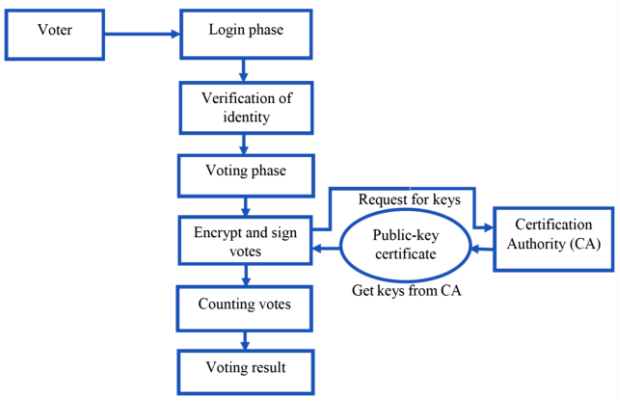

Figure 2. A block diagram illustrating the integration of the certificate authority (CA) within Helios++.

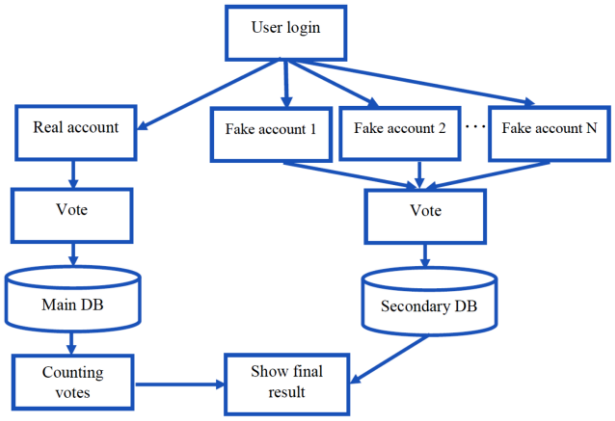

Figure 3. A block diagram illustration of adding multi-accounts to Helios++.

C. Helios++ Interface

The original English interface of Helios has been improved as it is difficult to understand and deal with the interface by people who do not have extensive knowledge of technology. Many commands are difficult to understand or it is not clear why they exist or what they should do. In the proposed Helios++ system, some commands have been modified and unnecessary ones have been omitted.

Furthermore, a new Arabic interface has been added to Helios++ to make the system more usable by people who do not know English in Arab countries. That is where many voters cannot read English words or feel uncomfortable when dealing with commands in that language. Also, significant work has been done to improve this new Arabic interface by reducing the unnecessary commands and adding clarifications for each step.

D. Mutual Authentication

In the current version of Helios++, it is assumed that the voting environment is controllable somehow. For example, it could be organizational elections (e.g., university-level elections). Thus, the registration phase can be based on the voter’s organizational email so that

other personal accounts of the voter can be provided based on that. Indeed, authentication is done for the voter assuming that the registration phase has been done in a trustworthy environment. However, for larger scale elections, an untrustworthy environment should be assumed. In this latter case, we are investigating a hybrid mutual authentication protocol that is based on both digital signature and biometry for voter verification similar to that outlined in [24]. Inclusion of the biometric authentication would mean that the enrollment phase needs to be done in advance before elections. Then, the biometric recognition phase will be activated during the voter's login. However, this is considered out of the scope of the current work and will be investigated in the future.

4. IMPLEMENTATION OF HELIOS++

In the proposed Helios++ system, there is a CA that creates certificates for voters and issues public and private keys to them. This helps to prove their ownership of these keys and they are authorized to vote in the elections. The CA also creates a certificate for the Helios++ server. The keys are used in the encryption process as well as the digital signature. The voting process in Helios++ is outlined in Algorithm 1.

Considering the implementation of the multi-accounts feature, at first the voter logs into the system using the available registration methods (The hybrid mutual authentication using digital signatures and biometry mentioned above is not included in the current version of the work). The voter is then asked if he/she wants to make this account his/her real account. If the voter chooses "yes", his/her vote will be counted among the final votes. If a voter chooses "no", his/her vote will not be counted. Voters can change their real accounts at any time (just before the election).

Note that relying on IP or physical addresses of voters for deciding which accounts belong to one person lets the system vulnerable to spoofing and impersonation attacks. Alternatively, relying on organizational email accounts limits the usability of the system and still suffers from stealing identity attacks. Thus, a sophisticated solution to this issue can be based on the protocol outlined in Section 3-C. Algorithm 2 outlines the use of multi-accounts in the voting process of Helios++.

For working on the user interface side, Django uses the MTV pattern to design software. It is a collection of three elements: Model, Template, and View. The model helps in dealing with the database. It's the data access layer that handles data. The template is a presentation layer that completely manages the UI portion. The view is used to implement business logic and interact with the model to carry data and render the template. The user asks for a resource for Django. Django acts as a controller and verifies the resource available in the URL. If a URL is set, the view that interacts with the model and template is called, where a template is rendered. Django responds to the user and sends the template as a response.

Algorithm 1: *The Helios++ voting process*

INPUT: Voter's login

OUTPUT: Voting final result

1. Voter login to the system
2. Generate the certificate:
   PU_{Voter} = Voter's public key
   PR_{Voter} = Voter's private key
3. Vote for a candidate:
   V = Vote
4. Encryption and signature:
   Encrypt = $(PU_{Voter,}$ V)
   Sign = $(PR_{Voter}$, V)
5. Cast the ballot:
   FinalResult = FinalResult + V
6. Return (FinalResult)

Algorithm 2: *Voting using multi-accounts*

INPUT: Voter's login by a "real" or "fake" account

OUTPUT: Updating the related database and showing the final result

1. Voter login to the system
2. If using a "real" account;
   a) Vote for a candidate
      V = Vote
   b) Update the main database (DB_{main}) with the new vote:
      $DB_{main} = DB_{main} + V$
   c) FinalResult = DB_{main}
3. If using a "fake" account;
   a) Vote for a candidate
      V = Vote
   b) Update the secondary database (DB_{Sec}) with the new vote:
      $DB_{Sec} = DB_{Sec} + V$
   c) FinalResult = DB_{Sec}
4. Return (FinalResult)

To improve the system user interface, the work has been done on the necessary parts of the templates. Templates appearing to users that contain commands and words have been modified, simplified, and enhanced. Some of the unnecessary commands and words have been removed and explanatory objects have been added. As for adding Arabic to the system, the commands in Python have been used to translate the interface. Also, a button that appears on all pages to change the language from English to Arabic and vice versa has been added.

5. Result and Discussion

In this section, some experimental results of the Python implementation of the proposed Helios++ system are presented. Figure 4 represents the interface of the CA that appears to the voter. The voter can create her/his certificate, download her/his certificate, download the Helios++ certificate, and also check the Helios++ certificate. When the "Generate Certificate" button is pressed, the system automatically generates a voter certificate that contains the voter's keys. These keys are used during the voting process. The "Get Certificate" and "Get Public Key" buttons allow the voter to download his/her certificate, download the Helios++ public key, and save them to her/his device in a safe place. Finally, there is a button labeled "Verify Helios", which when pressed the voter is redirected to a special page containing the certificate of Helios++, where he/she can verify its validity.

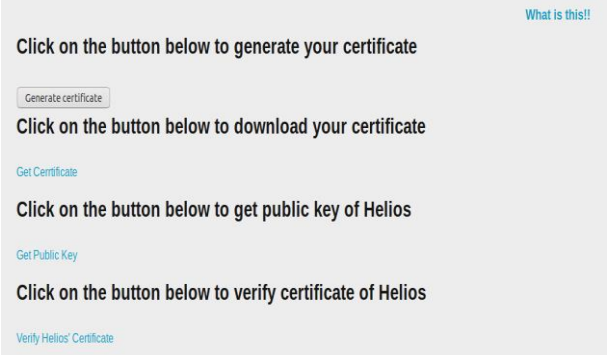

Figure 4. Helios++ CA page.

Upon completion of the certificate creation process, the voter goes to the voting page. After she/he votes and selects the candidate she/he wants, the voter will then choose if she/he wants to encrypt her/his vote and/or sign it (using either RSA or DSA). If selected, the system will automatically use the voter keys in the certificate for encrypting and signing. Figure 5 represents the voting page.

Figure 5. Helios++ voting page.

For testing, a Lenovo computer was used with an Intel Celeron processor and 4GB RAM. The time taken by the system to sign the vote has been calculated. Figure 6 shows the results where RSA with different key sizes and DSA are used for the signature. The sizes of the RSA keys tested are 512, 1024, 2048, and 4096 bits, respectively. The DSA key size is 2048 bits. Note that all results show that the time required to sign a vote is less than one second, which for our case can be considered to be a very short time. Time increases as the volume of the key used increases. When the size of the key used in the DSA algorithm is equal to the key in the RSA, the speed of performance is almost equal or with a very small difference.

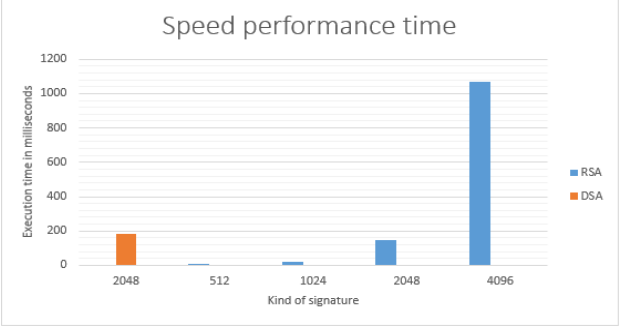

Figure 6 Results of the speed of digital signature performance.

To encrypt the votes, the AES algorithm has been used. The AES is a symmetric encryption algorithm, so an asymmetric RSA algorithm is used with it. The AES key is first created and then the RSA algorithm is used to encrypt the AES key. AES algorithm is then used to encrypt votes. Different key sizes were used for the AES algorithm and calculate the time required for encryption as shown in Figure 7.

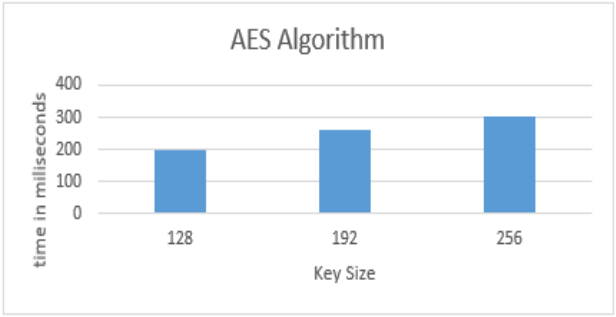

Figure 7 Results of the speed of the AES Algorithm.

The original Helios interface has been improved and made easier to use by voters. For example, Figure 8 shows the "Review page" after it has been improved, where some of the commands have been clarified and some unimportant texts have been removed.

Furthermore, a questionnaire including 60 people of different ages (22 to 70 years old) has been conducted to know their opinions about the easiness of using the proposed I-voting system (Helios++) compared to the original Helios. 50% of the participants are males, 50% are females, 58% are employees and 42% are non-employees, 75% have completed their education, 17% are students, and 8% are uneducated. Participants have voted using Helios as well as the proposed Helios++ systems. Then, they have been asked 10 questions about their

satisfaction with the voting process. The questions and answers are summarized in Table 1.

Figure 8: Helios++ review page.

TABLE 1.QUESTIONNAIRE SUMMARY.

	Questions	Answers	
		Yes	*No*
1	Do you trust Internet voting systems?	63%	37%
2	Was it difficult to use Helios?	68%	32%
3	Do you understand all the orders in Helios?	58%	42%
4	Will you use the Helios system to vote in the future?	38%	62%
5	Was the language in Helios understood?	53%	47%
6	Was the Helios++ easy to use?	85%	15%
7	Did you use the Arabic language to understand the orders in Helios++?	65%	35%
8	Will you use the Helios++ to vote in the future?	81%	19%
9	Did you complete the voting using Helios?	68%	32%
10	Did you complete the voting using the Helios++?	91%	9%

It can be noted that adding the CA to the voting system increases security because it is a trusted entity that creates public and private keys. Public and private keys are used for encryption and digital signature, so the process of creating them by a trusted entity is necessary. This addition also increases the voters' confidence in the system and thus increases the turnout and its use by them. Furthermore, the use of multiple accounts greatly reduces the risk of coercion and also gives the voter the freedom to choose the account through which he wants to log in. Since there are four ways to vote (One real account and three fake accounts are possible for each user in the current implementation of Helios++. These accounts can be based on Google, Facebook, Yahoo, and/or LinkedIn accounts), this makes it difficult for the coercer to distinguish which of these accounts is the real account. The use of this feature added a lot to the proposed voting system.

Finally, the questionnaire shows that the proposed Helios++ system is superior to Helios in terms of ease of use. Those who had difficulty using both systems did not know technology. The majority used the Arabic language during the voting because the group tested was Arab. It was also noted that some people did not complete the voting in both systems due to a large number of instructions and pages to which they travel.

6. CONCLUSION

A new I-voting system called Helios++ has been proposed. This system is based on Helios and public-key certificates. It has improvements to the security, scalability, coercion, and usability aspects of the Helios voting system. On the security and scalability side, it has been suggested to use a CA for creating a certificate for each voter. The possibility of signing each vote was added using either the RSA or DSA algorithm. To reduce the risk of coercion, it is suggested that each voter has up to four accounts. One of these accounts is “real”. This means that when a voter votes through it, this vote will be counted. Other accounts will be used if the voter is coerced, in which case the vote will not be counted. The simplifications and modifications made to Helios interfaces and adding Arabic language interfaces have increased the Helios++ system usability, especially for Arab users. The conducted questionnaire has indicated that voters are more satisfied with the proposed system compared to the Helios system. One possible future work direction is to integrate Helios++ with a biometric authentication system for initial voter enrolment/.

REFERENCES

[1] R. Aditya, B. Lee, C. Boyd, and E. Dawson, "An efficient mixnet-based voting scheme providing receipt-freeness," *International Conference on Trust, Privacy and Security in Digital Business*, 2004, pp. 152-161: Springer.

[2] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang, “An Efficient E2E Verifiable E-voting System without Setup Assumptions,” *IEEE Security & Privacy*, May/June 2017, pp. 14-23.

[3] Wei-Jr Lai, Yung-Chen Hsieh, Chih-Wen Hsueh, and Ja-Ling Wu, “DATE: A Decentralized, Anonymous, and Transparent E-voting System,” *Proceedings of 1st IEEE International Conference on Hot Information-Centric Networking (HotICN 2018)*, 2018, pp. 24-29.

[4] A. Qureshi, D. Megías and H. Rifà-Pous, "SeVEP: Secure and Verifiable Electronic Polling System," *IEEE Access*, vol. 7, pp. 19266-19290, 2019.

[5] B. Shahzad and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," *IEEE Access*, vol. 7, pp. 24477-24488, 2019.

[6] B. Adida, "Helios: Web-based Open-Audit Voting," in *USENIX 17th Security Symposium*, 2008, vol. 17, pp. 335-348.

[7] L. Panizo Alonso, M. Gasco, D. Y. Marcos del Blanco, J. A. Hermida Alonso, J. Barrat, and H. Alaiz Moreton, "E-voting system evaluation based on the Council of Europe recommendations: Helios Voting," *IEEE Transactions on Emerging Topics in Computing* (Early Access), 19 November 2018.

[8] F. Karayumak, M. M. Olembo, M. Kauer, and M. J. E. W. Volkamer, "Usability Analysis of Helios-An Open Source Verifiable Remote Electronic Voting System," *Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, San Francisco, CA, August 8 - 9, 2011.

[9] Olivier Pereira, "Internet Voting with Helios," in *Real-World Electronic Voting Design, Analysis, and Deployment*, Feng Hao, Peter Y. A. Ryan Eds., Taylor and Francis, Auerbach Publications, New York, 2016.
[10] V. Cortier and B. J. J. o. C. S. Smyth, "Attacking and fixing Helios: An analysis of ballot secrecy," *J. Comput. Secur.*, vol. 21, no. 1, pp. 89–148, 2013.
[11] D. Chung, M. Bishop, and S. Peisert, "Distributed Helios-Mitigating Denial of Service Attacks in Online Voting," 2016, UC Davis: College of Engineering. Retrieved from https://escholarship.org/uc/item/7xs630v9.
[12] S. Srinivasan, C. Culnane, J. Heather, S. Schneider, and Z. Xia, "Countering ballot stuffing and incorporating eligibility verifiability in Helios," in *International Conference on Network and System Security*, 2015, pp. 335-348: Springer.
[13] M. Meyer and B. J. I. P. L. Smyth, "Exploiting re-voting in the Helios election system," *Information Processing Letters*, Volume 143, March 2019, pp. 14-19.
[14] V. Cortier, D. Galindo, S. Glondu, and M. Izabachène, "Election verifiability for helios under weaker trust assumptions," *Lect. Notes Comput. Sci.*, vol. 8713 LNCS, no. PART 2, pp. 327–344, 2014.
[15] O. Kulyk, V. Teague, and M. Volkamer, "Extending Helios Towards Private Eligibility Verifiability," in *E-Voting and Identity: 5th International Conference (VoteID 2015)*, Switzerland, September 2-4, 2015, Proceedings, R. Haenni, R. E. Koenig, and D. Wikström, Eds. Cham: Springer International Publishing, 2015, pp. 57–73.
[16] D. Bernhard, O. Pereira, and B. Warinschi, "How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios," in *Advances in Cryptology - ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security*, Beijing, China, December 2-6, 2012. Proceedings, X. Wang and K. Sako, Eds., Springer Berlin Heidelberg, 2012, pp. 626–643.
[17] D. Y. Marcos del Blanco and M. Gascó, "A Protocolized, Comparative Study of Helios Voting and Scytl/iVote," *Sixth International Conference on eDemocracy & eGovernment (ICEDEG)*, Quito, Ecuador, 2019, pp. 31-38.
[18] S. T. F. Al-Janabi and N. H. Abid, "Security of internet voting schemes: A survey," *REVISTA AUS Journal*, Special Issue No. 26-2, 2019, pp. 260-270
[19] N. H. Abid and S. T. F. Al-Janabi, "A framework for I-voting based on Helios and public-key certificates," *REVISTA AUS Journal*, Special Issue No. 26-2, 2019, pp. 234-243.
[20] W. Bokslag and M. de Vries, "Evaluating e-voting: theory and practice," arXiv preprint, arXiv: 02509, 2016.
[21] M. Meyer and B. Smyth, "An attack against the Helios election system that exploits re-voting," arXiv:1612.04099v3 [cs. CR] 3 Nov 2018.
[22] Alicia Filipiak. Design and formal analysis of security protocols, an application to electronic voting and mobile payment, Ph.D. Thesis, Cryptography and Security [cs.CR]. Université de Lorraine, 2018.
[23] W. Bokslag and M. de Vries, "Evaluating e-voting: theory and practice," arXiv:1602.02509 [cs. CY], 2016.
[24] V. Augoye and A. Tomlinson, "Mutual Authentication in Electronic Voting Schemes," *16th Annual Conference on Privacy, Security and Trust (PST)*, Belfast, 2018, pp. 1-2.

Noor Hamad Abid, born in Ramadi, 1993. She received a B.Sc. degree from the College of Computer Science and Information Technology, University of Anbar in 2015. She recently obtained an M.Sc. degree in computer science from the College of Computer Science and Information Technology, University of Anbar, Iraq. Her research interests include cryptography and information security.

Prof. Dr. Sufyan T. Faraj Al-Janabi was born in Haditha, Iraq (1971). He obtained his B.Sc. (1992), M.Sc. (1995), and Ph.D. (1999) in Electronic and Communications Engineering from the College of Engineering, Al-Nahrain University in Baghdad. He started as a faculty member in the Computer Engineering Dept., the University of Baghdad in 1999. Then he became the head of that department in 2001. During May 2004- June 2006, he was the Dean of College of Information Technology, Al-Nahrain University. He became the Dean of College of Computer Science and Information Technology, University of Anbar, Ramadi from July 2006 to May 2010. His research interest includes internet protocols, information security, and quantum cryptography. Prof. Al-Janabi is the winner of the 1st Award for the Best Research Paper in Information Security from the Association of Arab Universities (AARU), Jordan, 2003. He is a member of ACM, ASEE, IACR, and IEEE. More information can be found on the personal website link at: *https://sites.google.com/view/sufyan-aljanabi*