A survey on the security of remote e-voting systems

Alexandra Goltsis
Linköping University
Linköping, Sweden
alego025@student.liu.se

Matilda Engström Ericsson
Linköping University
Linköping, Sweden
mater832@student.liu.se

***Abstract*—The aim of this survey is to investigate the modern protocols and systems available for remote e-voting today. The survey introduces some relevant background to absentee e-voting such as encryption, security requirements and system characteristics of a voting system. The voting systems discussed in this survey are the Estonian i-voting system, Helios, CHVote and sVote for Switzerland, Australia's iVote and the Moscow i-voting system. These systems mostly use technologies such as double envelope scheme, mix networks and different types of encryption. The Moscow i-voting system uses another technology called blockchain, which is a research area of great interest for internet voting systems today.**

From the analyzed literature, we conclude that there are many promising technologies with the purpose of creating secure voting systems online. However, problems were found in every analyzed system, and for that reason we believe more research is needed before implementing these technologies in larger elections. The survey moreover identifies prominent contradictions from the analyzed literature; the cost of internet voting systems, the turnout generated by internet voting systems and the necessary compromise between usability and security.

***Index Terms*—absentee voting, e-voting, internet voting, online voting, voting systems, end-to-end**

I. Introduction

The concepts of democracy and voting are an important part of the society that most people live in today. For many of us, voting in an election means using a physical paper to vote in a pre-decided location. However, imagine if voting could take place in each person's own smart device, no matter where they are? In a democracy, all votes count, and everyone should have the opportunity to vote. It is, however, important to consider the security risks of the systems in these cases. The result of the election should be indisputable. It is important that a vote cannot be traced back to an individual, that each person can be guaranteed that their vote is correctly counted and that there is trust in the system, both from those who understand the technology and from those who do not. In addition, the technology and the systems need to be completely secure both today and in the future, to be able to guarantee that all the requirements for a voting system are fulfilled. So, the question is if any of these modern technologies for remote e-voting are secure enough or if the security risks still are too big to take the risk. On this question, various countries think differently, since a few (e.g. Estonia) already use this technology while some have stopped using it and others have not tried it at all. By analyzing the existing technologies and systems, we aim to be able to make a recommendation on whether voting online is secure to use or not, and if so - how to do it.

A. Limitations

The term electronic voting (e-voting) includes using both machines at polling stations (instead of paper) and remote e-voting, which is voting at any place from any device. Remote e-voting can also be called internet voting (i-voting) or online voting. In this survey we will focus on remote e-voting and the technologies that possibly could be used for that purpose. We will also be focusing on technologies that have been used and/or discussed during the last ten years.

II. Important security concepts

One big challenge of creating remote e-voting systems is to be able to guarantee security, which is also very important to be able to do. To understand these aspects of the systems, we are going to explain some important security concepts for this purpose.

A. Encryption

Many of the developed e-voting systems today use cryptography to make the system more secure. This section aims to explain some cryptographic properties used in systems later discussed in the survey.

1) Homomorphic Encryption: For remote e-voting systems homomorphic encryption is a way of ensuring anonymity, thanks to the fact that one can perform calculations on encrypted votes. This means that the final results can be computed without revealing the content of the votes [35]. The encrypted sum of encrypted values can be found only if the cryptographic system is homomorphic. The down side to homomorphic encryption is that it is inefficient when dealing with a lot of data [25], which is usually the case in an election.

2) Blind signatures: Another way of protecting the anonymity of the votes is to use blind signatures. These are similar to digital contracts, but with the difference that the signer can sign the message without knowing the content of it. For e-voting, servers commonly use blind signatures to be able to sign a vote from an eligible voter without knowing what that particular voter voted for. The point is that the server will not know the relation between the signed results, computed by the server, and the signatures used for verification later. This algorithm uses any public and private key digital signature algorithm, for instance RSA or DSA. To use blind signatures we assume:

- V = the vote,
- d = the private key of the signer,
- e, N = the public key of the signer, and
- s = the signature of vote.

The voter then generate a random number r which satisfies $gcd(r, N) = 1$, and blinds the vote by

$$V' = r^e \cdot V \cdot mod(N).$$

So, V' is the blinded vote that the voter is sending to the signer. The signer is using V' to compute a blinded signature s'. This is done by

$$s' = (V')^d = r \cdot V^d \cdot mod(N).$$

Using this the voter can compute the signature s by

$$s = s' \cdot r^{-1} \cdot mod(N) = V^d.$$

Now the voter has the true signatures of the signer and the process is complete [7, 25, 20].

3) El-Gamal: The El-Gamal encryption system is an asymmetric key encryption algorithm based on the Diffie-Hellman key exchange,not to be confused with the El-Gamal signature scheme. It generates a public key and a private key, that are used to encrypt and decrypt messages. Standard El-Gamal has a multiplicative homomorphic property that is utilized in re-encryption mix nets. In addition, El-Gamal can be modified to support a property called homomorphic addition. This property is useful for tallying votes, as it allows for summation of votes without decrypting the individual ones. The final sum, a cipher text, can then be decrypted by authorities into plain text, which corresponds to the election result [35].

B. Requirements for secure e-voting

To be able to ensure a secure e-voting system, there are some security requirements that need to be satisfied. These are the following [9, 25, 20]:

- Anonymity: In an e-voting system it means that votes cannot be tracked to an individual, meaning that no one should know what you voted for. Although, the fact that you voted can be known.
- Eligibility: If the system has eligibility, it means that only eligible people will be able to vote in the election.
- Uniqueness: In an e-voting system, uniqueness means that only one vote per person will be accepted and counted in the election results.
- Accuracy: Each vote is correctly counted, and has not been modified by anyone.
- Receipt-freeness: The e-voting system will not give the voter a receipt on the vote. This is important to minimize the risk of people buying votes.
- Uncoercibility: An eligible voter should not be forced to make any specific choice in an election. If there is a possibility of that to happen, they need to be able to vote several times. If that is the case, then the last vote has to be the one that counts.
- Availability: The system should be available to the users when they want to vote.
- Individual verifiability: The individual voter can verify that their vote has been cast and recorded as intended.
- Universal verifiability: Anyone can independently verify that, for example, only eligible voters have cast a vote and that the vote has been tallied as recorded.
- Transparency: Different stakeholders, such as political parties, election observers and voters, can independently verify that the election process is conducted according to procedure.

C. Characteristics of a system

Every system has different characteristics. These are important to have in mind when analyzing the security of a system. In this section we will describe those that we are going to discuss later in the survey.

1) End-to-end (E2E): There are generally speaking three stages of voting: voter registration, vote casting and vote tallying. It is important that the system has a good solution to all of these and that it is secure all the way. During the first stage, when the registration of the voters takes place, it is important that each person is authenticated correctly. This can be done by for instance using public key cryptography, and verifying that the authenticated voter is an eligible voter. When the vote, is cast it is important that the uniqueness criteria of the vote is satisfied. Lastly, the votes should be tallied correctly, and the result should be verifiable. Various approaches are used to solve issues within each step [23].

2) End-to-end verifiability (E2Ev): In voting, an E2Ev system is a system where the voter can verify the correctness of the votes. To guarantee E2Ev, there are mainly three concepts that need to be satisfied. The first one, cast-as-intended (CAI), means that the voter should be able to verify that their encrypted vote accounts for what they voted for. The second one, recorded-as-cast (RAC), means that the voter should be able to verify that their vote has been recorded as desired. Lastly, tallied-as-recorded (TAR), means that anyone should be able to verify that the votes have been counted correctly (without knowing what a particular person voted for) [4].

3) Decentralized or centralized system: A decentralized system requires various entities to make decisions about the accuracy of the data, not just one as in a centralized system. Using a decentralized system is, however, not enough to ensure transparency [9]. Using a centralized database comes with the risk that the data might be modified by a third-party and that the result is not shown in real time [3].

III. Related work

Some earlier work related to our work has been done by Vivek et al. [33] who did a literature review on the blockchain technology in e-voting systems. In their survey they conclude that e-voting systems using a technology called blockchain can guarantee security, reliability, decentralized storage and anonymity. They also discuss several architecture and design features that can be used in combination with blockchain, for example; Smart Contracts, Short-Linkable Ring Signature, Elliptical Curve Cryptography and Blind Signatures. They also mention that further improvements can be done to increase the scalability of the systems that are using blockchain. Another similar work has been done by Maesa and Mori [23] who also did a review on the use of blockchain in various systems, including e-voting systems. They discuss some of the requirements for a secure e-voting system, such as the importance of using an E2Ev system and that anyone should be able to verify the election result, even if they did not vote. The authors conclude their survey by speculating about the blockchain technology. They believe it will evolve and be used in even more fields in the future.

Stenbro [31] did a survey on e-voting in general. In this survey the author discusses the basics of voting systems, the requirements for secure e-voting systems and cryptographic theory. He also discusses some early e-voting systems, such as the double envelope scheme used in Estonia, the Helios System and the Norwegian voting system. The author concludes that a remote e-voting system never will be completely secure, but that the advantages may outweigh the disadvantages.

IV. Technologies

There are many different technologies used in remote e-voting systems, which can be combined with each other. Some of them will be explained in this section.

A. Blockchain

The blockchain technology was originally created in 2008 for the cryptocurrency BitCoin, but blockchain can, and has, been used for a lot more than only cryptocurrency. Blockchain is a technology consisting of various technologies such as distributed ledgers, public key encryption, merkle tree hashing and consensus protocols [32]. During the last years, the usage of blockchain in e-voting systems has been a very popular topic. It has been discussed and written about a lot, which differs from a few years ago when there was not nearly as much information about it, even though the technology existed already.

Blockchain is, as the name suggests, a chain of blocks where each block contains a hash of the previous block. This property makes blockchain an append-only structure. Changing or removing a message in an existing block is nearly impossible, due the fact that the hash values in the all following blocks have to change too. This means that the chain of the blocks are immutable, which implies that it can ensure vote accuracy. Additionally, it is a decentralized peer-to-peer network, which makes the data both individually and universally verifiable. To accept a new entry in the network, the nodes in the network must agree about the correctness of the vote [19]. According to Yang et al. [35] a positive aspect of blockchain in voting systems is that it functions without a centralized party maintaining the database. Another positive aspect is the high availability of such a system [2]. However, there are also problems, such as some performance issues which arise when creating large networks with many blocks [9]. Despite these positive characteristics of blockchain there are also contradicting papers which conclude that blockchain is not secure enough for remote e-voting, as a decentralized system also has drawbacks [26].

To make blockchain even more secure, various cryptography technologies can be combined with blockchain, for example; smart contracts, secret contracts and blind signatures. These will be discussed here.

1) Smart Contracts: A smart contract is a self-executing contract, which mostly is used together with blockchain. The purpose of the smart contract is to be able to use digital contracts without a central authority. Smart contracts are signed, and then sent to the blockchain network where they get either accepted or rejected by miners, who verify the contracts. The smart contract has five stages: negotiation, development, deployment, maintenance as well as learning and self-destruction [34].

The first public blockchain technology was Ethereum, which today is the most widely used platform [34]. Authors Al-Madani et al. [3] created a voting application based on Ethereum Blockchain technology using smart contracts.

2) Secret Contracts: Aaron Fernandes et al. [9] created a blockchain-based framework and protocol intended for e-voting, in which they also used smart contracts. One of the issues with blockchain is that it cannot ensure the anonymity of the voters. This is where secret contracts come in. The secret contracts solution ensures that only eligible voters can vote, but also that the vote can not be tracked back to the voter.

3) Blind Signatures: Blind signatures can be used throughout all three voting stages. In the registration phase it can be used to hide the identity of the voter. The voter sends their personal information to the server, which checks if the user is an eligible voter. If the voter is eligible, the voter then receives a certificate and a unique identifier to use in the next phase, the casting phase. However, the unique identifier is first blinded, using blind signature. Then, during the tallying, the accepted votes are counted [25].

Irina Dyachkova and Anton Rakitskiy [7] describe in their paper how they created an anonymous voting system using blind signatures in combination with blockchain. They also used an anonymous data transfer channel to make the decentralized network used in blockchain anonymous. The vote is added to the chain of blocks using blind signatures, and this cannot be changed or removed. The system has functionality that allows the voter to check their own vote, as well as the total number of votes. According to the authors, the system ensures all the requirements for a secure system regarding anonymity. They believe the system to be a good alternative to traditional voting systems.

B. Double Envelope Scheme

The goal of using the double envelope scheme protocol is to guarantee the secrecy of the vote. This characteristic makes the protocol often used in systems for absentee voting [17]. The protocol was named after its process, as it puts the message in two digital envelopes, which really is digital encryption using cryptography keys twice on the message. The ballot is put into the first envelope by encrypting the message using the election private key. Then, that envelope is put into the second one by using the voter's digital signature. The ballot has now been put into a double envelope and can be sent to the election server [18]. The purpose of the inner envelope is to protect the secrecy of the ballot while the outer establishes the voter's identity. When the eligibility of the voter has been verified, the outer envelope can be removed, which leaves only an anonymous vote [17]. The Double envelope Scheme can be combined with several other technologies to create a voting system.

C. Mix Network (mix net)

Mix nets are based on public key cryptography and reorganizes the encrypted input data, so that anonymity can be achieved. This is done by mixing the input data (the votes) and outputting it in an order that makes it impossible to know the input [22]. Mix nets are using so called senders and mix servers. In e-voting, the senders are the voters and their messages represent the voters ballot. The mix server receives input from the senders and mixes the input so that the mix servers output is randomized. This is a technique that has been used in several political elections, for instance in Estonia, Norway, Switzerland and Australia, probably since using mix nets in e-voting protocols can help ensure the anonymity of the vote, as well as verifiability. In fact, the link between the voter and their vote should be secret, as long as not all mix servers are corrupted [14].

There are two types of mix nets; decryption mix net (DMN) and re-encryption mix net (RMN). In decryption mix nets, the messages are decrypted in each layer and then mixed, unlike the second where each server re-encrypt inputs and then mixes them. Re-encryption mix nets are often used together with the El-Gamal encryption method [22].

V. Internet voting systems

In this section we are going to compare existing systems, and describe which technique and protocols they use. The aim is to give an overview of the evolution of absentee e-voting. It will be written in chronological order, according to when the system first was created.

A. Estonian system

Estonia is known for being the first country to use i-voting for nation-wide elections. The Estonian i-voting system was first introduced in 2005. It was first used in parliamentary elections in 2007, and is still used today[1]. The system is primarily based on the double envelope scheme which uses public key cryptography and digital signatures. The basic idea is that the voter encrypts the vote using an election public key and then signs it with the voters personal digital signature [30]. The system has changed and improved through the years, but the process of voting in an election in the Estonian system's infancy was as follows [17]:

- Voter registration: First the voter authenticates itself by using their ID card, mobile ID or a digital identity document. This gives them the options they have to vote for in the election.
- Vote casting: When the voter has made their choice, the ballot is signed with the election public key. Then, the ballot is signed again using the voters personal digital signature, their ID card, mobile ID or digital identity document. Signing the vote with a digital signature allows the voter to re-cast their vote without jeopardizing the uniqueness requirement. The encryption method used is RSA-OAEP.
- Vote tallying: All votes are decrypted by the election private key after the voting period has ended. The election result is presented in two lists; one list containing the votes and one list containing the voters. Now the anonymous ballots can be tallied.

This version of the system is not E2E [30] and the security of this system has been questioned several times. For example, according to authors Springall et al. [30] the Estonian i-voting system relies heavily on complex procedures to ensure security and transparency, which were not upheld during the 2013 election when they observed the election. Some of the issues that the authors found were:

1) officials using PCs that contain other software, such as PokerStars.ee
2) downloading software using http before the election
3) releasing videos of officials typing in the root password and footage of the WiFi credentials, as well as
4) using personal USB sticks when moving the official election results from the counting server.

In the same paper, they performed attacks on a mock system on both the client-side and the server-side. They found that by introducing malware on the server-side they could change the outcome of the election during the tallying process, and on the client-side they could steal votes without the client knowing. These are dangerous attacks for a system that does not have the E2E verifiability property. For that reason, the researchers' opinion was that Estonia should stop the use of i-voting, but these problems were not taken into account by the Estonian government.

There have since been proposals on how to achieve the E2E property. One suggestion was to change the current cryptographic system to homomorphic encryption (e.g. El-Gamal) and a re-encryption mix net [18]. Something similar to this suggestion was implemented in 2017. Today, the votes in Estonia are mixed before decryption in the tallying process and a homomorphic encryption is used. According

[1] https://e-estonia.com/solutions/e-governance/i-voting/

to the information on their website the system today is E2E, as well as upholding the ballot anonymity, uniqueness and uncoercibility requirements [8]. Also, server-code is open-source and published on GitHub[2] [21].

Despite that, there are still improvements to be made on the system. In 2020 Ajish and AnilKumar [28] wrote in their paper that the system has some positive aspects, such as anonymous votes and the recorded as intended property, but that there are still some issues with other aspects. The writers claim that the weakest point in the system is the ID-card used in the authentication, and propose the use of QR code and biometric authentication instead, which according to them would increase security even more.

Another proposal on improvement of the Estonian system is to combine the double envelope scheme with blockchain. Cosmas Krisna Adiputra et al. [2] created one example of this kind of system. That system has higher availability than the double envelope scheme itself, given by the centralized characteristic of blockchain. They also claim that their new proposed system has higher verifiability than the original system, since everyone has access to the data. Although, the downside is that their system violates the anonymity requirement. For this, they propose the use of for example blind signatures, mix networks or homomorphic encryption in combination with their system.

B. Helios

Helios was first released in 2008 and is said to ensure anonymity and E2Ev. The system is open-source and more than 2,000,000 votes have been cast using Helios[3]. However, even the creators of Helios advise against using the i-voting system in larger elections. They write on their website: ”Online elections are appropriate when one does not expect a large attempt at defrauding or coercing voters.” And then continue: ”For some elections, notably US Federal and State elections, the stakes are too high, and we recommend against capturing votes over the Internet”[4]. Due to changes and improvements in the system, there are several versions of Helios.

1) Helios 1.0: The first release of Helios was Helios 1.0, which was released in 2008 [1]. The technologies used in the different stages are the following:

- Voter registration: To create an election or cast a vote the user has to register on Helios. That is done by entering a name, email address and password. When a new election is created a public and private El-Gamal key is generated by Helios, which are used in later stages. Additionally, when an election is created, an email is sent to the voter which contains the email address of the user and an election specific password [5].
- Vote casting: In this phase, the voter can cast the vote on the Helios web application, which uses the El-Gamal key keypair. After the user has cast their vote, the vote is displayed using a SHA-1 hash. The user can now choose to either audit the vote or seal it. If sealed, the vote is encrypted [5].
- Vote tallying: Mix networks are used with the goal of ensuring the anonymity of the votes when they are being counted. This is done by first shuffling all the, still encrypted, votes. Afterwards, the votes are decrypted and counted [1]. The mix network used is called Sako-Kilian Mixnet, which takes El-Gamal ciphertext as input [5].

According to one of the creators of Helios, the votes are both individual and universal verifiable [5].

2) Helios 2.0: Helios 2.0 is an upgrade of Helios 1.0, with improvements on a couple of things that needed to be fixed. The biggest change is how the votes are being tallied, since using a mix net introduced some issues. For instance, it is not possible to use votes with different weights when using mix networks to tally the votes. So, instead Helios 2.0 uses a homomorphic tallying, which also is more efficient. Another update was that they are using distributed decryption [1]. A study [29] has shown that Helios 2.0 are not verifiable, though the creators claim it to be on their web page. Additionally, the interface of the system was improved from Helios 1.0 to 2.0 [1].

3) Other versions: Helios 3.1 is the current version of Helios [29]. The code is open-source and can be found on GitHub[5], where JavaScript, HTML and python are being used. There is also another version called Helios-C, which implements digital signing of the ballot [29].

C. Swiss systems

There are several systems tested in Switzerland. Two of the most famous ones are CHVote and sVote. These systems were first used in 2004 respectively 2005, but neither are used until today. When used, postal voting was much more popular than i-voting where these systems were used. According to Micha and Uwe [24] 80-85% of Swiss residents used the postal service, while 10-15% cast their votes online. They also found that voters who used the online technology likely would have voted using another technology, if the online technology had not been available. However, it is not certain that this would be the case in other countries that do not have a convenient voting system, such as the postal voting system.

1) CHVote: The CHVote system has been used in Geneva. Both CHVote 1.0 and 2.0 have since been discontinued and will not be put into production.

CHVote 1.0: The first version of CHVote[6] is a system that is not verifiable at all [13]. In Geneva, the voters receive a vote mail. It can only be used one time and the voter has to decide whether to vote online, by mail or at the ballot box [12]. If the voter decides to use CHVote, the voter must first enter an identification number. If the user is authorized, the client can connect to a server and enter the vote. The user then must confirm their choice and identity. Lastly, the user receives a confirmation that the vote was accepted.

[2] https://github.com/vvk-ehk/ivxv

[3] https://vote.heliosvoting.org/

[4] https://vote.heliosvoting.org/faq

[5] https://github.com/benadida/helios-server

[6] https://github.com/republique-et-canton-de-geneve/chvote-1-0

In CHVote the voters are required to use a computer that has a browser with a Java-Plugin as the system uses a Java-voting-applet on the client-PC. The PC connects to an e-voting server via an internet connection. The client-PC and internet connection are according to Franke [10] the main issues that need to be addressed in the system. He also states that the system does not solve the trusted platform problem. According to Haenni et al. [13] the main issues with the system was the lack of transparency, verifiability and the insecure platform problem.

CHVote 2.0: The second generation of CHVote[7] was developed to provide E2E encryption with individual and universal verifiability. According to Micha and Uwe [24] i-voting did not increase turnout in Geneva and in 2018 it was discontinued due to financial reasons by the State of Geneva [13].

CHVote 3.0: The CHVote project is however continued by the Bern University of Applied Sciences. The source code is yet to be released, but Haenni et al. [13] continues to produce documentation of the system. The system uses a re-encryption mix net together with El-Gamal encryption. According to them, the most critical component is the printing authority, because there is a risk that a ballot is being submitted using the real identity of the voter. Another issue that needs to be addressed is the possibility that an adversary may attack the voting device. It can be solved either by introducing pure code voting, which severely impacts the usability of the system negatively, or by distributing trusted hardware to voters, which impacts the costs.

2) Scytl's sVote: sVote is an e-voting system created by Scytl and requires users to physically identify themselves at a local administration [24], but can also be used for administrative tasks in addition to i-voting. As of 2018 it was used in Fribourg, Neuchâtel, and Thurgau.

During 2019 the system underwent a public intrusion test where 173 findings were reported. Three of the source code findings were considered critical [27]. After this test, politicians banned i-voting in 2019 due to the security risks, even though it could be concluded that no past election or votes had been manipulated. It is put on hold until experts have concluded that the issues are solved, which they are working on right now. The goal is to have a new secure system. According to Haines et al. [15] the issues with the individual and universal verifiability were due to mistakes in the cryptographic components in the system. These issues also apply to the Australian iVote system in the following section.

D. Australia and Scytl's iVote

In 2015, 5% of the votes in an Australian state election were cast using iVote. Halderman and Teague [16] reviewed the iVote system during the election and found critical security flaws that could compromise the ballot anonymity and steal votes.

In the system, the vote is encrypted on the client side and sent to a voting server. It is also sent to a separate verification service. The caster of the vote receives a receipt for the vote and can either telephone the verification service or visit an online service to verify that the vote was included in the final count. Halderman and Teague [16] found that this mechanism was flawed.

The client uses AngularJS with JavaScript, HTML and CSS. Most of the content is received from cvs.ivote.nsw.gov.au, but some from a third party analytics tool ivote.piwikpro.com.

The major issue with iVote was the use of a third-party server since it used weak SSL configurations, for instance insecure Diffie-Hellman parameters. These parameters allowed the authors Halderman and Teague [16] to steal votes by injecting code into the application in a man-in-the-middle attack. One of the vulnerabilities was a zero-day vulnerability, only known to a few people in the world.

The threads used in the JavaScript framework implemented cryptographic operations and would pass messages between themselves. These messages could, for instance, contain the content of the vote. Halderman and Teague [16] found that these messages could be intercepted and altered, or sent to a server operated by the attacker, together with the voters authentication credentials.

According to them, there are multiple ways to achieve a man-in-the-middle attack on the iVote system. They describe the following:

- ”using client-side malware,
- by compromising insecure Wi-Fi access points,
- by poisoning ISP DNS caches to redirect the traffic to an attacker-controlled IP address,
- by attacking vulnerable routers or links along the path to the server, or
- by redirecting packets by hijacking BGP prefixes.” [16]

These attacks do not need to target a specific voter, any insecure host or infrastructure can be targeted to achieve the attacks. Furthermore, administers on home or workplace networks may take advantage of their privileges to perform attacks.

E. Moscow voting system

The Moscow e-voting system was used for the first time in the Moscow election in 2019, as the first system based on blockchain that has been used in a legal binding election. Everyone that registered in advance could use this voting system, which uses oauth as authentication service for the registration[8].

The Moscow i-voting system used multilevel El-Gamal encryption over finite fields. The smart contracts are made up of Solidity code and are used in a permissioned Ethereum blockchain. Each encrypted ballot is stored as one transaction in the protocol. Voters can relate the encrypted ballot to the corresponding vote in clear text via the blockchain, which should provide the cast-as-intended property.

Moscow allowed researchers to review the code before the election. According to Gaudry and Golovnev [11] the result

[7] https://chvote2.gitlab.io

[8] https://github.com/moscow-technologies/blockchain-voting

was not as verifiable as they thought it would be using a blockchain-based ledger, as the voter in this system has a limited amount of time to check the vote. It is furthermore not possible to rewrite the history of the ledger after a voter has checked it. The private keys are stored in the blockchain to ensure that voters can check the vote.

Gaudry and Golovnev [11] found that the system used too small encryption keys, as three primes of 256 bits were used. These keys are in fact so small that the private key can be computed from the public key in only a few minutes. As the decryption of the ballots was part of the smart contract, the system was modified after this discovery. As 256 bits is the largest (unsigned) integer type, natively supported by the Solidity programming language of the Ethereum smart contracts, it is probable that this was the reason behind choosing the small keys. However, the key size was changed to 1024 bits for the election when the decryption was moved to outside of the smart contract. Although, a key size of 1024 bits is still too small to be secure and the primes are not chosen in public, which means that the designers could choose primes that would allow them to compute the keys using discrete logarithms.

For the patched, second version of the system Gaudry and Golovnev [11] discovered a possible attack against the ballot anonymity, due to the semantics of the system.

F. General contradictions

There is no consensus on the costs of an i-voting system [6]. These systems are often argued to be a cheap type of voting system, but there are few actual studies on the subject. In practice, many countries have discontinued their system for financial reasons. CHVote is an example of this [13].

The turnout generated by absentee e-voting systems is also highly debated. The systems may increase accessibility of voting to more groups of people [6] and are said to increase turnout of the elections [20], but that might not be the case. For instance, a study showed that the turnout had not increased in two Swiss cantons when remote e-voting was tested [24].

These systems may also be difficult to implement depending on the design of the country's electoral system and require a lot of administrative work.

What has been concluded is that the design and implementation of the voting system highly affects the impact of both turnout and costs. If the system is not convenient, voters will go for other options. However, Haenni et al. [13] concluded that not even a convenient system is entirely secure, due to possible malware on the voting device. It is hard to please everyone and find a good balance between a user-friendly and a secure system. Voters consider i-voting to be convenient, but are also aware of possible usability and security issues [6].

Authors Adida et al. [1] found that evidence and counter-evidence for the correctness of the election made it easier to handle potential complaints for open-audit elections, contrary to common belief.

TABLE I
BENEFITS OF I-VOTING

Requirement	Benefit(s)
Availability	Disabled people can vote without assistance Easy access for people in hospital, long term care facilities and those who live in remote areas, abroad or are working
Anonymity of votes	Votes are encrypted
Uniqueness	Machine checks if person has voted multiple times
Vote accuracy	Lower risk that votes arrive late Fewer counting errors, automatic counting
Cost	Automatic counting, implying less staff Low costs for voters

TABLE II
DRAWBACKS OF I-VOTING

Requirement	Drawback(s)
Vote coercion	Difficult to verify if people are voting freely in an uncontrolled environment
Vote accuracy	Votes can be manipulated in a cyber attack or by malware on voting device Identification codes may be stolen or sold In conflict with anonymity of votes
Availability	Denial of Service attacks Unreliable internet connections
Usability	Software patching to prevent malware attacks Prue code voting
Universal verifiability	Difficult for laymen to understand, have to rely on experts, Difficult to recount votes
Individual verifiability	Difficult to implement working mechanisms for RAI and CAI
Cost	Development, maintenance and security updates.

VI. DISCUSSIONS

Even though a lot of the papers are positive about using remote e-voting systems, there are also those that are critical to it. One problem is that if there is a failure, it will most likely be more serious using remote e-voting compared to traditional voting. This is because a failure in a remote e-voting system is often more large-scale and might be harder to detect. A comparison can be made to other systems, such as BitCoin or banking system, which today is widely used online. The problem though is that there are failures and problems with those too, but these are not that impacting as they would be in an election result. Furthermore, high stake elections may also be more targeted by adversaries that other online systems [26]. The benefits and drawbacks of using remote e-voting systems are summarized in Table I and II.

There are great expectations on i-voting systems, however there are many contradictions that can be found in the research field, such as whether those systems actually increase turnout and reduce cost or not. Despite the fact that people spend a lot of time online today, voting is still more common offline. In Switzerland only 10-15% of the votes were cast online, and

TABLE III

SUMMARY OF WHICH REQUIREMENTS EACH SYSTEM ACHIEVE

System name	Technologies	E2Ev	Open-source	Possible improvements
Estonian e-voting system	Double Envelope Scheme, Mix Nets and homomorphic encryption	Yes	Yes, server-side[1]	The authentication and the that fact that is it centralized
Helios 2.0	El-Gamal, homomorphic encryption and hash	Yes	Yes[2]	Authentication
CHVote 3.0 (from Switzerland)	El-Gamal and Mix nets	Yes	Yes[3]	Secure Printing
sVote (from Switzerland)	Mix nets	No	Yes[4]	The cryptographic components
iVote (from Australia)	Mix nets	No	Only for qualified reviewers	Weak SSL configurations
Moscow i-voting system	Ethereum Blockchain, smart contracts and El-Gamal	-	Yes[5]	Too small key sizes

[1] https://github.com/vvk-ehk/ivxv
[2] https://github.com/benadida/helios-server
[3] https://gitlab.com/openchvote
[4] https://gitlab.com/swisspost-evoting/e-voting-system-2019
[5] https://github.com/moscow-technologies/ag-blockchain

in the 2015 Australian state election this number was as low as 5%. In Estonia, however, 44% of votes are cast using the i-voting system[9]. Estonia is, however, also the only country in the EU to have fully implemented i-voting [6]. Estonia has also been using remote e-voting systems for a longer time.

Virtually all i-voting systems in this survey have had issues regarding implementation and security. If the i-voting systems would be at least as secure as the physical voting systems, and have the trust of the public, it would be an easy choice to implement them instead of the physical ones as they increase accessibility. They may however be subject to zero-day vulnerabilities, as found by Halderman and Teague [16]. These vulnerabilities are not possible to mitigate, as they are not known to the public. Attacks like these may compromise the accuracy of the election and affect the outcome. It is also hard to find all vulnerabilities in a system, which means that there will almost always be some left. This is of course a problem. Even the creators of a large remote e-voting system, Helios, recommend against using it in bigger elections.

There have been some issues implementing secure cryptographic mechanisms in the systems that have been reviewed in this survey. The Moscow i-voting system used too small keys and may also have semantic issues. The public intrusion test of sVote revealed three critical source code findings. Although, there are also some promising technologies discussed today. One example is blockchain which has properties as: verifiable, transparent, decentralized and high availability. These are all wanted properties in a remote e-voting system, but there are also problems. For example, regarding the anonymity of the votes, which is stored in the blocks. These problems can most likely be solved, using for example blind signatures which can help solve the anonymity part. Also mix networks is a technology that is promising, which is good for anonymity, verifiability, and robustness. To use these technologies in a good combination (one technology cannot be used alone) would create a system which can be secure according to many definitions, but we can never guarantee complete security. It is also important for the system to be E2E verifiable to be considered secure, as for example the e-voting system in Estonia had to change their system so that it was going in that direction. A full summary of the discussed systems are shown in Table III.

An additional aspect of online voting is that one problem can cause a lot more damage than it can on offline voting, since it requires so much more to change or remove a considerable amount of votes physically. Therefore, it is even more important that the system is secure when using online tools. There is also the problem regarding the voters own devices that they are voting from. How can one ensure that those not have been corrupted before the election? These things are problems that are not solved today.

VII. Conclusions

When we started writing this survey, our biggest question was: Are remote e-voting secure enough for it to be used today? The answer is complicated. There are papers claiming their systems to be very secure and others state that these systems in fact are not that secure. Our conclusion from that will be that it does not seem impossible to implement a secure e-voting system, but it also seems hard. Firstly, this subject requires more research and more time spent on to figure out how to be able to guarantee the security that is needed and what type of security that is. When this is done, a system for e-voting might be tested, but we believe that it is not in the nearest future.

Acknowledgment

We would like to thank our supervisor Ioannis Avgouleas for all the help and support throughout the project.

[9] https://e-estonia.com/solutions/e-governance/i-voting/

REFERENCES

[1] B. Adida et al. "Electing a university president using open-audit voting: Analysis of real-world use of Helios." In: *In Proceedings of the 2009 Conference on Electronic Voting Technology (EVT)/Workshop on Trustworthy Elections (WOTE)*. 2009. URL: https://www.usenix.org/conference/evtwote-09/electing-university-president-using-open-audit-voting-analysis-real-world-use.

[2] C. K. Adiputra, R. Hjort, and H. Sato. "A Proposal of Blockchain-Based Electronic Voting System". In: *Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. 2018. DOI: 10.1109/WorldS4.2018.8611593.

[3] A. M. Al-madani et al. "Decentralized E-voting system based on Smart Contract by using Blockchain Technology". In: *International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*. 2020. DOI: 10.1109/ICSIDEMPC49020.2020.9299581.

[4] L. Alonso et al. "E-Voting System Evaluation Based on The Council of Europe Recommendations: Helios Voting." In: *IEEE Transactions on Emerging Topics in Computing*. 2018. DOI: 10.1109/TETC.2018.2881891.

[5] Adida B. "Helios: web-based open-audit voting." In: *Proceedings of the 17th conference on Security symposium.* 2008.

[6] European Commission. *Study on the Benefits and Drawbacks of Remote Voting*. 2018. DOI: 10.2838/677948.

[7] I. Dyachkova and A. Rakitskiy. "Anonymous Remote Voting System". In: 2019. DOI: 0850-0852.10.1109/SIBIRCON48586.2019.8958064.

[8] State Electoral Office of Estonia. *General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia*. 2017. URL: https://www.valimised.ee/sites/default/files/uploads/eng/IVXV-UK-1.0-eng.pdf.

[9] A. Fernandes et al. "Decentralized Online Voting using Blockchain and Secret Contracts." In: *International Conference on Information Networking (ICOIN)*. 2021. DOI: 10.1109/ICOIN50884.2021.9333966..

[10] D. Franke. "Security Analysis of the Geneva e-voting system". In: *Horbach, M. (Hrsg.), INFORMATIK 2013 – Informatik angepasst an Mensch, Organisation und Umwelt. Bonn: Gesellschaft für Informatik*. 2013, pp. 789–803.

[11] P. Gaudry and A. Golovnev. "Breaking the Encryption Scheme of the Moscow Internet Voting System". In: *International Conference on Financial Cryptography and Data Security*. 2020.

[12] J. Gerlach and U. Gasser. "Three Case Studies from Switzerland: E-Voting". In: 2009. URL: https://cyber.harvard.edu/sites/cyber.law.harvard.edu/files/Gerlach-Gasser_SwissCases_Evoting.pdf.

[13] Rolf Haenni et al. *CHVote Protocol Specification*. Cryptology ePrint Archive, Report 2017/325. https://eprint.iacr.org/2017/325. 2017.

[14] T. Haines and J. Müller. "SoK: Techniques for Verifiable Mix Nets". In: *IEEE 33rd Computer Security Foundations Symposium (CSF)*. 2020. DOI: 10.1109/CSF49147.2020.00012.

[15] T. Haines et al. "How not to prove your election outcome". In: *IEEE Symposium on Security and Privacy (SP)*. 2020. DOI: 10.1109/SP40000.2020.00048.

[16] J.A. Halderman and V. Teague. "The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election". In: 2015. DOI: 10.1007/978-3-319-22270-7_3.

[17] S. Heiberg and J. Willemson. "Verifiable internet voting in Estonia." In: *6th International Conference on Electronic Voting: Verifying the Vote (EVOTE)*. 2014. DOI: 10.1109/EVOTE.2014.7001135.

[18] S. Heiberg et al. "Improving the Verifiability of the Estonian Internet Voting Scheme". In: *6th International Conference on Electronic Voting: Verifying the Vote (EVOTE)*. 2016. DOI: 10.1007/978-3-319-52240-1_6.

[19] F. . Hjálmarsson et al. "Blockchain-Based E-Voting System." In: *IEEE 11th International Conference on Cloud Computing (CLOUD)*. 2018. DOI: 10.1109/CLOUD.2018.00151.

[20] S. Ibrahim et al. "Secure E-voting with blind signature". In: *4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings.* 2003. DOI: 10.1109/NCTT.2003.1188334.

[21] Puiggalí J. et al. "Verifiability Experiences in Government Online Voting Systems". In: *Krimmer R., Volkamer M., Braun Binder N., Kersting N., Pereira O., Schürmann C. (eds) Electronic Voting*. 2017. DOI: 10.1007/978-3-319-68687-5_15.

[22] M. Jakobsson, A. Juels, and R. Rivest. "Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking". In: *Proceedings of the 11th USENIX Security Symposium*. 2002. ISBN: 1931971005.

[23] D. D. F. Maesa and P. Mori. "Blockchain 3.0 applications survey". In: *Journal of Parallel and Distributed Computing*. 2020. DOI: 10.1016/j.jpdc.2019.12.019.

[24] G. Micha and S. Uwe. "Internet voting and turnout: Evidence from Switzerland". In: 2017. DOI: 10.1016/j.electstud.2017.03.001.

[25] T. Nguyen and T. Dang. "Enhanced security in Internet voting protocol using blind signature and dynamic ballots". In: *Electronic Commerce Research*. 2013. DOI: 10.1007/s10660-013-9120-5.

[26] S. Park et al. "Going from bad to worse: from Internet voting to blockchain voting." In: *Journal of Cybersecurity*. 2021. DOI: 7.10.1093/cybsec/tyaa025.

[27] J. Puiggalí. "Implementing a public security scrutiny of an online voting system: the Swiss experience". In: 2019.

[28] Ajish S and K. S. AnilKumar. “Secure I-voting system using QR code and biometric authentication”. In: *Information Security Journal: A Global Perspective*. 2020. DOI: 10.1080/19393555.2020.1867261.

[29] B. Smyth, S. Frink, and M. R. Clarkson. “Election Verifiability: Cryptographic Definitions and an Analysis of Helios, Helios-C, and JCJ”. In: *Cryptology ePrint Archive Technical Report*. 2021. URL: https://eprint.iacr.org/2015/233.

[30] D. Springall et al. “Security Analysis of the Estonian Internet Voting System”. In: *In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS ’14)*. 2014. DOI: 10.1145/2660267.2660315.

[31] M. Stenebro. “A Survey of Modern Electronic Voting Technologies”. In: *Norwegian University of Science and Technology Department of Telematics*. 2010. URL: http://hdl.handle.net/11250/262287.

[32] P. Tasca and C. Tessone. “A Taxonomy of Blockchain Technologies: Principles of Identification and Classification”. In: *Journal of Parallel and Distributed Computing*. 2019. DOI: 10.5195/ledger.2019.140.

[33] S. K. Vivek et al. “E-Voting Systems using Blockchain: An Exploratory Literature Survey”. In: *Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2020. DOI: 10.1109/ICIRCA48905.2020.9183185.

[34] S. Wang et al. “Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2019. DOI: 10.1109/TSMC.2019.2895123.

[35] X. Yang et al. “Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities”. In: *Future Generation Computer Systems*. 2020. DOI: 10.1016/j.future.2020.06.051.