Secure Benchmarking using Electronic Voting

Vivek Agrawal and Einar Arthur Snekkenes
Department of Information Security and Communication Technology,
Norwegian University of Science and Technology, Gjøvik, Norway

Keywords: Ballot, Benchmarking, Electronic Voting, Response, Secure Benchmark.

Abstract: It is a common practice in the industry to organize benchmark processes to establish information security performance evaluation standards. A benchmarking system collects information security-related data from the organization to establish a standard. The information shared by the organization often contains sensitive data (details of the vulnerability, Cyber attacks). The present benchmarking systems do not provide a secure way of exchanging sensitive information between the submitter and the benchmark authority. Furthermore, there is a lack of any mechanism for the submitters to verify that the final benchmark result contains the response submitted by them. Hence, people are reluctant to take active participation in sharing their sensitive information in the benchmarking process. We propose a novel approach to solve the security limitations of present benchmarking systems by applying the concepts of electronic voting to benchmark. Our solution provides secrecy to submitters' identity and to the benchmark responses. Our approach also ensures that all the submitted responses have been correctly counted and considered in the final benchmark result.

1 INTRODUCTION

Researchers and experts suggest that the development and use of sound and repeatable Information Security Management (ISM) practices bring organizations closer to meeting their business objectives. Organizations can measure the quality of ISM practices, either by comparing their processes to other organizations or by measuring compliance according to established security standards (Whitman and Mattord, 2014). Information security is considered to be one of the business requirements that should be appropriately addressed by the enterprises. Enterprises hold a large volume of valuable information which is required to follow compliance with regulations and law about information security.

Benchmarking is a well-known process of improving performance by continuously identifying, understanding, and adapting security practices and processes found inside and outside an organization (Hidalgo and Albors, 2008). Benchmarking requires sharing organization-specific sensitive information to compare the performance in a specific domain. Typically, it requires Benchmark Submitters (members who possess valuable information) to submit the answers to a set of questions to establish a benchmark standard. However, the most significant barrier to benchmarking is the fact that many organizations are not willing to share their organization-specific sensitive data. The submitter may need to share the critical information, i.e., information related to security incidents that they often face. Information related to any successful attack is often perceived as a failure and is kept secret by the organization. The details of these events can create a bad image for the organization in the marketplace (Whitman and Mattord, 2014). Any security incident within the company can jeopardize the business operation and reputation (Kanno, 2009). Therefore, it may be considered risky to participate in the benchmarking process as illegitimate access to the sensitive information may hamper the business operation of the organization.

Currently, benchmarking is practiced almost all over the world (O'Rourke et al., 2012). There is a variety of methods by which different forms of data are developed, collected, and transmitted during the benchmarking event. There may be conflicts of interests in and incentives for the benchmark authorities to manipulate the benchmark process (IOSCO, 2013). The current benchmarking models fail to provide a secure way to share sensitive information (Kanno, 2009). Benchmark does not provide an efficient way for the data submitters to verify that the final benchmark result contains the response submitted by them (ABB, 2017), (ISF, 2017). Hence, it lacks the sense

Agrawal, V. and Snekkenes, E.
Secure Benchmarking using Electronic Voting.
DOI: 10.5220/0006827800250040
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 25-40
ISBN: 978-989-758-319-3

of transparency with collecting the data, analyzing the data, and publishing the final result.

We are establishing 'UnRizkNow' (Agrawal et al., 2017), (Agrawal and Snekkenes, 2017), (Agrawal and Szekeres, 2017) as an open electronic community of practice (eCoP) (Mathwick et al., 2008), (Wiertz and de Ruyter, 2007) to allow information security practitioners (ISP) to share InfoSec knowledge without violating the information security requirements. We are working towards providing a secure benchmarking service on UnRizkNow eCoP. We aim to protect the identity of the members who participate in the benchmark task. We also target to protect the sensitive data shared by the members/organization in the benchmarking process. Therefore, we propose applying the concepts of electronic voting to the benchmarking process on eCoP. We formulate the current benchmarking model based on a literature review. We also establish the requirements of a secure benchmarking system. Furthermore, we map the benchmarking system to electronic voting system by mapping their protocol, structure, and concepts. We also demonstrate how a secure benchmark can be conducted on the UnRizkNow platform using the electronic voting approach. We have identified the following research questions in this study.

RQ1 What are the requirements of a secure benchmarking system?

RQ2 How can a secure benchmarking system be mapped to a electronic voting system?

RQ3 How can a benchmarking system be built using the electronic voting concepts

RQ4 To what extent does the EV approach make the benchmarking system secure?

The paper is organized as follows: In Section 2, we overview of benchmarking and describe the benchmarking model that is widely used. In Section 3 the research method used in this study is described. In Section 4, an overview of the electronic voting system is presented. The essential phases of an EV system, the structure of vote, and security requirement of EV systems and schemes are also presented. A mapping of benchmarking concepts to EV concepts is presented in Section 5. In Section 6, an application of EV concepts to a benchmarking system is described to demonstrate how secure benchmarking can be conducted using the EV approach. A discussion based on the security analysis follows in Section 7. The limitation of the current study and the scope of further improvement is highlighted in Section 8. We conclude in Section 9.

2 OVERVIEW OF BENCHMARKING

The aim of this section is to provide a general overview of benchmarking. We identify major activities and actors involved in a typical benchmarking system. The benchmark model that is widely used everywhere is also presented.

2.1 Benchmarking Protocol

Development of benchmarks is an iterative and ongoing process that is likely to involve sharing information with other organizations working towards an agreeable method (Kelessidis, 2000). Benchmarking is pioneered by Xerox Corporation in the 1979s to perform better in the international competition in the photocopier market (Kelessidis, 2000). The idea of benchmarking was restricted to very few companies, e.g., AT&T, Motorola, Xerox in the beginning. However, governmental and non-profit organizations have begun implementing benchmarking as late as the early 1990's. Information security Forum (ISF) provides benchmarks in the form of their premium service (ISF, 2017). We derive the benchmarking protocol from (ESMA-EBA, 2013) report. A benchmarking system typically comprises the following activities and actors:

1. **Benchmark Administration:** It includes all the stages and processes involved in the benchmarking process. The establishment, design, production, and dissemination of a benchmark from the gathering of the input data and the calculation of the benchmark based on the input data to the dissemination of the Benchmark to users including any review, adjustment, and modification of this process. The legal person or entity responsible for executing this phase is called *Benchmark Administrator* (*BA*). BA also takes care of of publishing Benchmark values, which includes making available such values on the internet or by any other means, whether free of charge or not. According to (ESMA-EBA, 2013) report, the activity of publishing benchmark values can be carried out by a separate entity, Benchmark Publisher.
2. **Benchmark Submission:** The activity of contributing to Benchmark data submissions to a BA. The Benchmark submission is done by *Benchmark Submitter* (*BS*). The data submitted by BS are used exclusively for the calculation of the Benchmark.
3. **Benchmark Calculation:** The activity of performing the calculation of the Benchmark based on the methodology provided by a Benchmark

Administrator and the data collected by the entity performing the calculation or the BA or submitted by BS. A legal person or entity responsible for performing this phase is called *Benchmark Calculation Agent* (*BCA*).

4. **Benchmark Service:** The activity of evaluating the performance in a certain domain by fetching benchmark data from BA. A profession client (by paragraphs 1, 2 and 3 of Section I of Annex II to Directive 2004/39/EC) who is interested in taking benchmark data from BA is called *Benchmark User* (*BU*).

The task of BA and BCA may be performed by distinct legal entities or may be grouped such that one entity performs more than one. Figure 1 shows the information flow among the actors involved in the benchmarking process.

2.2 Structure of a Benchmark

The structure of benchmark depends on the overall objective of the benchmark. A benchmark typically consists of some questions created to assess the performance of the various organization in a particular domain. The question has options which indicate the possible answer to the question. The structure containing the answers to the questions is called a response. There are the following types of question formats:

- Yes/No questions: Submitter's answer is either Yes or No. The benchmark result of this question is a histogram chart consists of the frequency distribution of yes and no generated from the valid responses.
- Multiple-option Question: A question consists of various options, but the submitter can submit only one option. The benchmark result of this question will be a histogram chart consists of the frequency distribution of all the options calculated from the valid response.
- Open question (Numerical): Submitter can formulate the answer and write it down. However, the answer must be a numeral that follows the condition provided in the question. For instance, the age of the submitter question can only take numbers in the range of 1-100. The benchmark result of this question will be an average value calculated on the total valid responses submitted by the benchmark submitter.

2.3 Benchmarking Model

In this study, the principles of benchmarking model are set up according to the guideline given by European Securities and Markets Authority (ESMA) and European Banking Authority (ESMA-EBA, 2013). The same principles are widely followed by many organizations, e.g., ISF (Forum, 2017), ABB (ABB, 2017) and ISM-Benchmark (Kanno, 2009). The benchmarking process is usually conducted in two phases, i.e., Benchmark standard establishment and Benchmark as service. An overview of a complete benchmarking process is shown in Figure 1. The details of the two phases are given as follows:

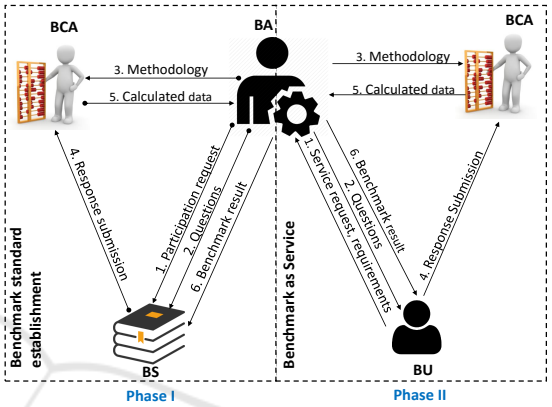


Figure 1: The information flow among the benchmarking actors in a benchmarking system. Phase I is carried out among *BA*, *BCA*, and *BS*. Phase II is carried out among *BA*, *BCA*, and *BU*.

2.3.1 Benchmark Standard Establishment

The first phase of the benchmark process is called Benchmark standard establishment. The aim of performing this phase is to collect data from the relevant organization to understand how well they perform in a given domain. This phase is usually executed synchronously, i.e., all the participants involved in the benchmarking task work simultaneously. Typically, BA hires or establishes a contract with an entity that can act as a BCA in the process. BA also makes a list of all the potential entities who can serve as a submitter. The details of step 1-6 in Phase I are given as follows:

1. *BA* sends a formal request to the members to participate in the benchmarking process and ask for response submission. The status of the member is marked as *BS* when the member agrees to participate
2. *BA* sends questions to assess a particular domain to *BS*.
3. *BA* sends the details of the question format and calculation method to *BCA*. The methodology is used by *BCA* to calculate the benchmark result.
4. *BS* sends the response to *BCA*.

5. *BCA* applies the methodology on the aggregated benchmark data and calculate the result of each question. *BCA* sends the benchmark result to *BA*.
6. *BA* sends the benchmark result to *BS* or posts it on a common web portal.

2.3.2 Benchmark as Service

The second phase of the benchmarking is called Benchmarking as a service. It is often provided by a private organization as a paid service, and by a public organization (government) as a free service. A user (BU) who is interested to know the status of its performance usually go for this type of service. This phase is executed asynchronously, i.e., it is not necessary that all the users contact BA at the same time. However, there is some service-level-agreement involved between BA and BU. The details of the steps 1-6 in phase II are given as follows:

1. *BU* establishes a contract with *BA* to get the latest benchmarked data in the given domain. *BU* sends the details of the requirements, i.e., the domain of the benchmark, the format of the outcome, delivery time to BA.
2. *BA* chooses the relevant questions from the list used in phase I and creates a new set of questions specific to the requirement received from *BU*.
3. *BA* sends the details of the question format and calculation method to *BCA*. The methodology is used by *BCA* to calculate the benchmark result.
4. *BU* answers the questions of benchmark and sends the response to *BCA*.
5. *BCA* applies the methodology on the aggregated benchmark data from *BU* and calculate the result of each question. *BCA* sends the benchmark result to *BA*.
6. *BA* sends the benchmark result to *BU*. This benchmark result contains the response submitted by *BU* to the given questions and the values that have been collected by *BA* in phase I. In this way, *BU* can compare its response with the benchmark standard and assess its performance.

2.3.3 Requirements of a Secure Benchmarking System

In this section, we answer RQ1 by establishing the security requirements of the benchmarking system. As far as we know, no comprehensive list of benchmarking security requirements have been published. Having carefully considered security issues in the context of benchmarking, we state what we believe are the key benchmarking security requirements.

1. **Completeness:** All valid responses should be counted correctly in the final calculation.
2. **Uniqueness:** Benchmark submitter can submit the response only once. The submitters should be allowed to submit their responses only once to control any practice to manipulate the overall result of the benchmark by submitting many responses.
3. **Universal Verifiability:** Anyone can verify that the published result is correctly computed from the responses that were correctly submitted. This is an important requirement as it signifies that the benchmarked data is calculated using the original submitted responses and it is not manipulated.
4. **Individual Verifiability:** Each eligible submitter can verify that his valid response was counted.
5. **Eligibility:** Only entitled benchmark submitter can submit a response.
6. **Secrecy:** Neither benchmark authorities nor anyone else can find out which submitter submitted which response.
7. **Soundness:** Any invalid response should not be counted in the final calculation.

3 RESEARCH METHOD

We applied the concepts of Design Science Research (DSR) (Hevner et al., 2004) to develop the scientific approach in this study. This research aims to solve an existing practical problem in the domain of Information system by creating an artifact based on the existing theories of electronic voting and cryptography. The problem is solved by applying creativity, innovation, and problem-solving capabilities. The created artifact would then be applied to UnRizkNow eCoP to enhance information sharing without compromising the sensitivity of the information. We adopted the five-step research process (Johannesson and Perjons, 2014) to research this study. Figure 2 shows the essential steps in the DSR model.

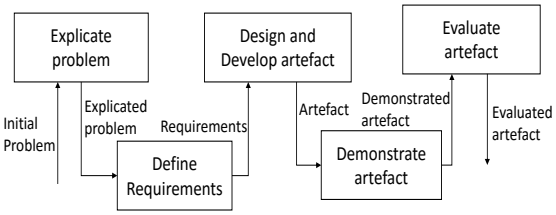


Figure 2: An overview of research method in Design Science Research Methodology (Johannesson and Perjons, 2014).

The first step of the DSR process, i.e., *explicate problem* is to investigate and analyze the practical problem. We defined the problems in the present benchmarking system, i.e., the lack of security. Fur-

ther, *define requirements* outlines a solution to the explicated problem in the form of an artifact and elicits requirements, which can be seen as a transformation of the problem into demands on the proposed artifact. We suggest a novel approach of conducting benchmark using the concepts of the Electronic voting scheme. In the phase, *Design and develop artifact* an artifact is created to address the explicated problem and fulfill the defined requirements. Our artifact consists of mapping the structure, protocol, and the concepts of benchmarking to electronic voting. *Demonstrate artifact* uses the developed artifact and applies this to a real-life case or any illustrative case. This phase aims to show that the artifact can solve an instance of the defined problem. We incorporate the proposed artifact to the UnRizkNow platform so that a secure benchmarking can be conducted on the platform. The final step is *evaluate artifact*, which determines how well the designed artifact solves the primary problem. We perform the security analysis on the developed artifact to show to what extent it fulfills the security requirements of a secure benchmarking system.

We also applied the DSR knowledge contribution framework (Gregor and Hevner, 2013) to highlight the nature of the contribution of our study. Figure 3 presents a 2X2 matrix of DSR research contributions. The x-axis, i.e., Application Domain Maturity (ADM) shows the maturity of the problem from high to low. The y-axis, i.e., Solution Maturity (SM) represents the current maturity of the artifacts from high to low that exist as potential starting points for solutions to the questions. The 2x2 matrix also identifies four kinds of design science contribution. A low ADM and low SM defines a new solution for new problems, and it is referred as *Invention*. A high ADM and Low SM defines new solutions for known problems, also known as *Improvement*. A low ADM and High SM indicates known solutions for new problems, also known as *Exaptation*. Finally, A high ADM and high SM indicates known solution for known problems, referred as *routine design*. Unlike other entities of the matrix, the routine design does not have a major knowledge contribution.

The idea of using the concepts of electronic voting conducting benchmarking tasks makes the benchmarking process more secure and trustworthy. The concept of electronic voting has been evolving in last two decades to facilitate election process in the democratic setting. However, it was never applied and tested in the setting of benchmarking. Therefore, our approach of solving the security and trust challenges in the present benchmarking process by extending the design knowledge that exists in the electronic voting place our contribution is in the ***exaptation*** quadrant of the DSR knowledge contribution framework.

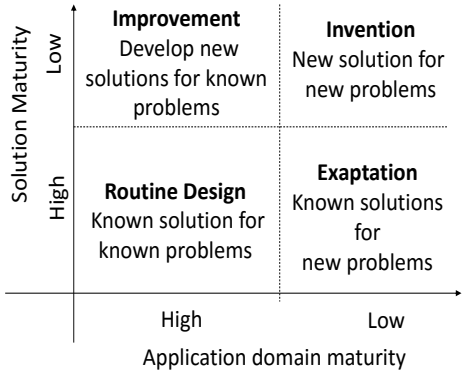


Figure 3: Design science contributions, adapted from (Gregor and Hevner, 2013).

4 AN OVERVIEW OF ELECTRONIC VOTING (EV)

The aim of the section is to present a detailed description of the electronic voting protocol, the structure of electronic voting along with the security requirements.

4.1 Electronic Voting

Electronic voting (EV) is appearing as an efficient and cost-effective way for conducting a voting process. The term e-voting is used to denote a voting process which allows voters to cast a secure and secret ballot over a network (Gritzalis, 2002). The first EV scheme was proposed by David Chaum (Chaum, 1981) in 1981. There have been many other schemes proposed by researchers since 1981, e.g., EV schemes with publicly verifiable secret sharing (Schoenmakers, 1999), (Neff, 2001); EV based on homomorphic encryption (Hirt and Sako, 2000); EV based on secret sharing techniques with a secure multiparty computation (Chen et al., 2014). (Gerlach and Gasser, 2009) describes EV experiences by mentioning how EV systems worked in Geneva and Zurich in Switzerland. Similarly, the EV systems of Estonia are studied in (Madise and Martens, 2006), (Vinkel, 2012). An EV protocol has many essential phases to carry out a successful election. We have compiled a list of phases that are very common across different EV protocols. The phases are as follows:

- Election administration: The process of setting up the election, publication of the identities of eligible voters, the list of candidates and the result of the election.

- Registration: The process of distributing secret credentials to voters and registering the corresponding public credentials.
- Tallying: The process of validating votes and determine the number of votes each party has received.
- Voting: The process of casting a vote in an election
- Ballot Processing: The processing of ballots and storing valid ballots in the bulletin board.

EV protocols involve several parties executing some specific set of roles (Cortier et al., 2014a). However, different schemes use different terms to denote the parties involved in the EV process. Table 1 describes the actors who are responsible for performing the EV tasks in five EV schemes.

4.2 Structure of Electronic Voting

The structure of voting depends on the nature of the election and the expected outcome. An election has a candidacy which consists of some candidates running in the election. A structure containing the vote is called a *ballot*. We identify the following typical election types:

- Yes/No voting: Voter's answer is yes or no. A typical example of this election where a voter is asked to reply to the question, "Do you agree with"regarding 'Yes' or 'No' answers.
- 1-out-of-L voting: Voter has L possibilities but can choose only one. This election format is used to select a leader (e.g., president) from a list of L candidates.
- K-out-of-L voting: Voter selects K different elements from the set of L possibilities. This type of election is used to choose council members in which the voter selects K from L candidates. The candidates who are selected the most number of times will be appointed as the council members. The order of the selection of the candidates is not important. $(1,L) \Rightarrow \{K \in \mathbb{N} : 1 \leq K \leq L\}$
- K-out-of-L ordered voting: Voter puts into order K different elements from the set of L possibilities. This type of election can be used to choose council members, but the candidate who is marked by the voter as first will get the most points.
- Write-in Voting: Voter can formulate the answer and write it down. This type of election is done when the answers are not fixed at the beginning and voters are asked to give their opinion on the given matter.

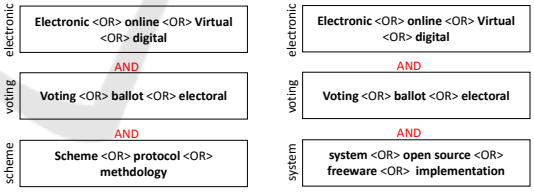
4.3 Requirements of the Secure Electronic Voting

Several researchers have proposed schemes for secure electronic voting processes with varying assumption. Therefore, different schemes fulfill different security requirements. We have compiled a list of requirements from different literature sources to highlight all the useful requirements that have been identified in the existing literature. We made a distinction between *schemes* and *systems* while compiling the list. Therefore, we have different criteria for the study selection in scheme and system.

Electronic Voting Scheme: Scheme is referred to the study where the conceptual model of electronic voting is presented regarding algorithm or theory. We used the search terms in Figure 4a to select the primary studies on the security requirements of electronic voting schemes. Additionally, we applied the following criteria on the search result to narrow down the relevant study.

- The literature is published on and after the year 2000.
- The literature has over 50 citations in the academic literature.
- Published in the English language.

The list is by far complete, but we restricted this study to include six schemes (Zu02 (Rjašková, 2002), Le02 (Lee and Kim, 2003), Le00 (Lee and Kim, 2000), Hi10 (Hirt, 2010), Ch05 (Chaum et al., 2005), Li04 (Liaw, 2004).



(a) (b)

Figure 4: Search terms used to find a) Electronic voting scheme b) Electronic voting system.

Electronic Voting System: We defined the system as those studies which are available as open source code, and it has been implemented in the real case studies. We used the search terms in Figure 4b to select the primary studies on the security requirements of the electronic voting system. Additionally, we applied the criteria that the source code is available to download on a reliable server (e.g. GitHub). The source code is also supported by English documentation or user manual. The list is by far complete, but we restricted this study to include four electronic voting systems (eVote (Pierro, 2017), Belenios (Cortier

Table 1: Actors involved in EV process in different schemes.

EV task	**LE02** (Lee and Kim, 2003)	**Belenios (Cortier et al., 2014b)**	**IVXV (of Estonia, 2017)**	**CHVote (Haenni et al., 2017)**	**eVote (Pierro, 2017)**
Election administration	Election Administrator	Election Administrator	Organiser	Election Administrator	Managers
Registration	Certificate Authority	Registrar	Collector	Printing Authority, election authorities	Managers
Tallying	Tallier	Trustee	Tallier	Election Authorities	Managers
Ballot Processing	Tallier & Administrator	Bulletin Board Manager	Processor	Bulletin Board	Managers
Voting	Voter	Voter	Voter	Voter	Voter

et al., 2014b), Chvote (Haenni et al., 2017), IVXV (of Estonia, 2017)). The details of the requirements of EV protocol are as follows:

1. **Completeness/ Correctness:** All valid ballots should be counted correctly in the final tally (Lee and Kim, 2003), (Hirt, 2010).
2. **Uniqueness/ Unreusability:** Voters can submit only one single ballot (Hirt, 2010).
3. **Universal Verifiability:** Anyone can verify that the published tally is correctly computed from the ballots that were correctly cast (Hirt, 2010), (Rjašková, 2002).
4. **Individual Verifiability:** Each eligible voter can verify that his ballot was counted. This property enables the voter to exclude with high probability the possibility that the vote has been manipulated by a compromised voting client (Haenni et al., 2017).
5. **Eligibility:** Only entitled voters are able to cast a ballot (Hirt, 2010).
6. **Anonymous/Secrecy/Privacy:** Neither voting authorities nor anyone else can find out which voter submitted which ballot (Liaw, 2004), (Hirt, 2010).
7. **Soundness:** Any invalid ballot should not be counted in the final tally (Hirt, 2010).
8. **Fairness:** No one can get extra information about the tally result before the publication phase (Liaw, 2004).
9. **Receipt-freeness/Incoercibility:** The voter cannot be coerced into casting a particular vote by a coercer. He must neither obtain nor be able to construct a receipt proving the content of his vote (Lee and Kim, 2003), (Liaw, 2004).
10. **Non-cheating:** Voters can accuse the authority of cheating without revealing ballots to others (Liaw, 2004).
11. **Robustness:** The voting system should be successful regardless of the partial failure of the system (Lee and Kim, 2000).
12. **Convenience:** Voters to cast their ballots quickly, in one session, and with minimal equipment or special skills (Liaw, 2004).
13. **Efficiency:** The whole election should be held promptly, for instance, all computations done in a reasonable amount of time and voters are not required to wait for other voters to complete the process (Liaw, 2004).
14. **Mobility:** Voters are not restricted by physical location from which they can cast their votes (Liaw, 2004).
15. **Auditability:** The system must be technically sufficiently simple so that a widest possible range of specialists could audit it (of Estonia, 2017).

Table 2 shows the list of the EV security requirements that are compiled from six EV schemes and four EV systems. The presence of **+** indicates that the given requirement is addressed. The requirement is considered as addressed if the author explicitly defines the given requirement in literature and justifies how the given EV protocol satisfies the requirement. **-** indicates that the given scheme/system does not address the requirement. It is also important to note that different schemes/systems address a security requirement under the different assumption and adversary models. For instance, the Hi10 (Hirt, 2010) scheme addresses 'soundness' for K-out-of-L voting structure, and Zu02 scheme (Rjašková, 2002) addresses 'soundness' for 1-out-of-L voting structure. Similarly, the *uniqueness* requirement is addressed by LE02 (Lee and Kim, 2003) scheme under the assumption that an adversary cannot access the randomness and any internal information saved inside the tamper-resistant randomizer distributed to the voters. The Li04 scheme (Liaw, 2004) addressed *uniqueness* requirement under the assumption that an adversary

cannot obtain a random number generated by the voting center.

5 MAPPING OF A BENCHMARKING TO AN EV SYSTEM

The aim of this section is to answer RQ2. We demonstrate how a benchmarking system can be mapped to the electronic voting system. To achieve our goal, we first map the benchmark protocol to the EV protocol, then we map the structure of benchmark to the structure of EV system. Finally, we map the overall concepts of benchmark to the EV concepts using ontology.

5.1 Mapping of the Benchmark Protocol to EV Protocol

The protocol mapping consists of the mapping of the benchmark phases and actors to the EV system phases and the actors. Table 3 shows the mapping of benchmark protocol to EV protocol. The main entities involved in the benchmark protocol are: a Benchmark Administrator BA, N Benchmark Calculating Agents BCA_j ($j = 1,...,N$), and M Benchmark submitter BS_i ($i = 1,...,M$). The roles of each entity are as follows:

- Benchmark Administrator - BA verifies the identities and the eligibility of M submitters. BA manages the whole benchmarking process (creates questions and announces the benchmark result).
- Benchmark Submitter - There are M submitter BS_i ($i = 1,...,M$). They have their digital signature keys certified by a certification authority (CA).
- Benchmark Calculating agent - There are N calculating agents BCA_j ($j = 1,...,N$) who cooperatively decrypt the collected responses to open the result of benchmarking. A threshold t denotes the lower bound of the number of authorities that is guaranteed to remain honest during the protocol.

The main entities involved in the electronic voting protocol are: an Election administrator EA, N Tallier T_j ($j = 1,...,N$), and M voter V_i ($i = 1,...,M$). The roles of each entity are as follows:

- Election Administrator - EA verifies the identities and the eligibility of M voters. EA manages the whole voting process (creates candidacy and announces the election result).
- Voter - There are M voter V_i ($i = 1,...,M$). They have their own digital signature keys certified by a certification authority (CA).
- Tallier - There are N Tallier T_j ($j = 1,...,N$) who cooperatively decrypt the collected ballots to open the result of the election. A threshold t denotes the lower bound of the number of authorities that is guaranteed to remain honest during the protocol.

Table 3 shows the mapping of the protocol between benchmark and EV system. It is clear from the table that the activity of Benchmark calculation can be mapped to Tallying, a benchmark submitter can be mapped to the voter.

5.2 Mapping of the Benchmark Structure to EV Structure

We map the structure of benchmarking system to an EV system with the help of the mapping of ballot, vote, candidacy, and candidates to response, answer, questions, and options respectively. Question QU is mapped to Candidacy Cd, Option o is mapped to Candidate C, answer a is mapped to vote v, response B is mapped to ballot BT. It is important to note that there is only one candidacy in an election, but a benchmark needs to have more than one question. Therefore, a benchmarking system needs the x number of EV instances to execute, where x is the number of questions in the benchmark.

In the electronic voting scheme ω, a candidacy Cd consists of L number of candidates C_i (where $i = 1,...,L$) who participate in the election to be elected to some position based on the outcome of the election. A voter can decide to vote for only 1 candidate (1-out-of-L voting) or more than 1 candidate (K-out-of-L voting) based on the requirement of the election. A voter casts his ballot in the election. A ballot BT consists of a vector of votes, $\overrightarrow{v} = (v_1,...,v_K)$, where v_i is the vote for the $i-$th candidate in the election. In K-out-of-L election, the following condition holds $(1,L) \Rightarrow \{K \in \mathbb{N} : 1 \leq K \leq L\}$.

A benchmarking system β consists of a number of questions Q_i (where $i = 1,...,x$). The idea of having the questions is to collect the feedback from the submitters to establish a performance standard. Each question Q_i comes with the list of options o_i (where $i = 1,...,L$). BS generates a vector of answers, $\overrightarrow{a} = (a_1,...,a_L)$, where a_i is the answer of the $i-$th option and $a_i \in \{0,1\}$. BS finally generates a response B consists of the answer vector $\overrightarrow{a}$. The number of response is equal to the number of questions available in the benchmark. The final response B_fin contains all the responses B_i ($i = 1,...,x$). Table 4 shows how the structure of benchmark can be completely mapped to the structure of EV. The structure of benchmarking system can be constructed using the K-out-of-L voting structure where $(1,L) \Rightarrow \{K \in \mathbb{N} : 1 \leq K \leq L\}$.

Table 2: The security properties of EV system, AB: Applicability to Benchmark, + indicates that the given security requirement is implemented in the scheme, - indicates that the given security requirement is not implemented in the scheme.

ID	Property	AB	Zu02	Le02	Le00	Hi10	Ch05	Li04	Ch	Be	eV	IV
1	Completeness/ Correctness	Y	-	+	+	+	+	+	-	-	+	-
2	Uniqueness/ Unreusability	Y	-	+	+	+	-	+	-	-	+	+
3	Universal Verifiability	Y	+	+	+	+	-	-	+	+	+	-
4	Individual Verifiability	Y	+	+	-	+	-	+	+	+	+	+
5	Eligibility	Y	+	+	+	+	+	-	-	-	-	-
6	Anonymous/ Secrecy/privacy	Y	+	+	+	+	+	+	+	-	+	+
7	Soundness	Y	-	+	+	+	-	-	-	-	+	-
8	Fairness	N	+	+	+	+	-	+	-	-	+	-
9	Receipt-freeness/ Incoercibility	N	+	+	+	+	-	+	-	-	-	+
10	Non-cheating	N	-	-	-	-	-	+	-	-	+	-
11	Robustness	N	+	+	+	-	-	+	-	-	-	-
12	Convenience	N	-	-	-	-	-	+	-	-	-	-
13	Efficiency	N	-	-	-	-	-	+	-	-	-	-
14	Mobility	N	-	-	-	-	-	+	-	-	-	-
15	Auditability	N	-	-	-	-	+	-	+	-	-	+

Table 3: Mapping of the protocol.

Phase		Actor	
Benchmark β	**EV** ω	**Benchmark** β	**EV** ω
Benchmark Administration [*BAdm*]	Election Administration [*EAdm*]	Benchmark Administrator [*BA*]	Election Administrator [*A*]
Benchmark calculation [*Bcal*]	Tallying [*ETal*]	Benchmark calculating agent [*BCA*]	Tallier [*T*]
Benchmark submission [*BSub*]	Voting [*Vo*]	Benchmark submitter [*BS*], user [*BU*]	Voter [*V*]

It is important to notice that 1-out-of-L voting structure is not suitable for the mapping between benchmark and electronic voting. 1-out-of-L voting structure expects only one vote in the ballot unlike K-out-of-L voting where a ballot contains a vector of votes. Therefore, *BCA* cannot calculate the frequency of individual option in the benchmark result using 1-out-of-L voting structure.

The structure of the benchmark for different question types are as follows:

5.2.1 Yes/No or True/False Questions

For this type of question in the benchmark $L = 2$, i.e., there are two options o_1 and o_2 available for the question. The answer vector will consist of $\overrightarrow{a} = (a_1, a_2)$. As submitter can select only option in the answer, the $\Sigma a_i = 1$. Therefore, the structure of $B = (a_1, a_2)$ The total response for this question is $M * B$ (where M is the number of submitter). The total number of *yes* can be counted by adding the a_1 answer vector and total number of *No* can be counted by adding the a_2 answer vector from all the submitters.

$$\text{Result of } Q_j = \{\text{Frequency of } Yes \text{ , Frequency of } No \}$$
$$= \Sigma_{i=1}^{M} BS_i[B_j(a_1)], \Sigma_{i=1}^{M} BS_i[B_j(a_2)] \quad (1)$$

where $BS_i[B_j(a_1)]$ denotes the response B_J submitted by BS_i; $B_j(a_1)$ denotes the answer component a_1 of response B_j

This type of question in benchmark is mapped to a K-out-of-L voting system (where $K = 1$) according to the mapping presented in Table 4. Yes, and No options are presented with candidate c_1 and c_2 respectively. The ballot BT contains the vote vector $\{v_1, v_2\}$ against the candidate c_1 and c_2. The frequency of yes and no can be counted by adding the votes cast by M voters in the favor of the candidates. Equation 1 takes the following form in EV.

$$\text{result of } Cd_j = \{\text{votes received by } c_1 \text{ , votes received by } c_2 \}$$
$$= \Sigma_{i=1}^{M} V_i[BT_j(v_1)], \Sigma_{i=1}^{M} V_i[BT_j(v_2)] \quad (2)$$

where $V_i[BT_j(v_1)]$ denotes the ballot BT_J cast by V_i; $BT_j(v_1)$ denotes the vote component v_1 of Ballot BT_j

Table 4: Mapping of the benchmark structure to EV structure. There are M number of voters and submitters, x number of questions and candidacy, L number of option, answer, candidates, and votes.

Benchmark					**EV**			
Question	Option	Answer	Response	$\Rightarrow$	Candidacy	Candidate	Vote	Ballot
Q_1	$o_1...o_L$	$a_1....a_L$	B_1		Cd_1	$c_1...c_L$	$v_1...v_L$	BT_1
Q_2	$o_1...o_L$	$a_1....a_L$	B_2		Cd_2	$c_1...c_L$	$v_1...v_L$	BT_2
...	...	...	...		...	...	...	...
...	...	...	...		...	...	...	...
Q_x	$o_1...o_L$	$a_1....a_L$	B_x		Cd_x	$c_1...c_L$	$v_1...v_L$	BT_x

5.2.2 Multiple Option Question

This type of question contains L possible option to choose from where $L > 2$. The answer vector will consist of $\overrightarrow{a} = (a_1,..,a_L)$. As submitter can select only one valid option out of L option, the $\Sigma a_i = 1$. Therefore, the structure of $B = (a_1,..,a_L)$ The total response for this question is $M * B$ (where M is the number of submitter). The frequency histogram can be generated by adding the answer vectors from all the submitters.

$$\text{Result of } Q_j = \{\text{Frequency of } o_1 \text{ ,....., Frequency of } o_L\}$$

$$\Sigma_{i=1}^{M} BS_i[B(a_1)], ..., \Sigma_{i=1}^{M} BS_i[B(a_L)] \quad (3)$$

where $BS_i[B_j(a_1)]$ denotes the response B_J submitted by BS_i; $B_j(a_1)$ denotes the answer component a_1 of response B_j

This type of question in benchmark is mapped to K-out-of-L voting system according to the mapping presented in Table 4. L possible options are mapped to L candidates. The ballot BT contains the vote vector $\{v_1,...,v_L\}$ against the candidate $c_1,...c_L$.The frequency of the $i-$th option is calculated by adding the votes received to $i-$th candidate. Therefore, the equation 3 takes the following form in EV.

$$\Sigma_{i=1}^{M} V_i[BT_x(v_1)], ..., \Sigma_{i=1}^{M} V_i[BT_x(v_L)] \quad (4)$$

where $V_i[BT_j(v_1)]$ denotes the ballot BT_J cast by V_i; $BT_j(v_1)$ denotes the vote component v_1 of Ballot BT_j

5.2.3 Open Question (Numerical)

This type of question does not provide any predefined options to the submitters. However, the submitter can enter a numeric value in the option field. Option field consists of a number of empty bits based on the numerical range provided to the submitter. The value of L in the option field is calculated as the ceiling function of $log_2 MX$, i.e., $L = \lceil log_2 MX \rceil$ where MX is the range. The number entered by the submitter is converted into the equivalent binary string to be saved into the answer vector $\overrightarrow{a}$. Let's take the case of question 2 in the appendix, "What percentage of the employee recognize a security issue? [range 0-100]". The valid values this question takes is 101. Therefore, the value of L can be calculated by applying the ceiling function to $\lceil log_2 101 \rceil$, i.e., $L = 7$. Let's assume that BS submit 50 as the answer of the question. The answer vector $\overrightarrow{a} = (0100110)$. The total number of the response for question Q_j is $M * B$ (where M is the number of submitter). For option i (where $i = 1,...,L$), the $i-$th components of each valid response of M submitters are summed up, i.e, $aa_i = \Sigma_{w=1}^{M} BS_w[B_j(a_i)]$, where aa_i is a count of the number of answers that has been received for the $i-$th bit of the binary representation to the question by all the submitters. The mean value of the Question Q_j is calculated by adding all aa_i in the following equation

$$\mu = \frac{1}{M} \Sigma_{i=1}^{L} aa_i 2^{i-1} \quad (5)$$

Open numerical question in benchmark is mapped to K-out-of-L voting system according to the mapping presented in Table 4. L possible options are mapped to L candidates. The ballot BT contains the vote vector $\{v_1,...,v_L\}$ against the candidate $c_1,...c_L$. The mean of the Question Q_x is calculated by firstly adding the $i-$th components of each valid ballot in vv, and then adding all vv and converting them to the decimal value. Equation 5 takes the following form:

$$vv_i = \Sigma_{w=1}^{M} V_w[BT_j(v_i)]; \mu = \frac{1}{M} \Sigma_{i=1}^{L} vv_i 2^{i-1} \quad (6)$$

where vv_i is a count of the number of votes that has been received for the $i-$th bit of the binary representation of the candidates to the candidacy by all the voters.

5.3 Mapping of Overall Concepts

We map the concepts of benchmarking system to electronic voting system using an ontology. The idea, of using and developing an ontology to explain the concepts, is derived from (Agrawal, 2016). Figure 5 presents ontologies of benchmarking system and electronic voting system. In our proposed ontology, there are ten main concepts (circular boxes) and ten

relationships (solid arrow lines). The text above the horizontal dotted line corresponds to the benchmarking system, while the text below the horizontal dotted line corresponds to the electronic voting system. The dotted horizontal line also demonstrates how can a concept and relationship from benchmark be mapped to electronic voting. Thus, figure 5 helps to understand the relationship between the benchmark and electronic voting clearly. It is evident from the given ontology that the concepts of the benchmark can be mapped to EV system.

The ontology of benchmark states that Benchmark Administrator performs benchmark administration by creating Benchmark. Benchmark has some Questions that consists of options. Submitter from different Organization participates in the Benchmark by submitting their response. A response contains answer of the questions. A response can be considered valid or invalid on the basis of the benchmark rules. Benchmark calculating agent (BCA) counts response based on a given methodology, and finally, BA publishes the Benchmark result.

The ontology also depicts that an election administrator (EA) performs administration by creating an election. The election has Candidacy that consists of some Candidate running for a certain post in the election. Voters from different Constituency area participate in the election by submitting their ballot which contains the vote for the candidates. A ballot can be valid or invalid based on the election rule. A tallier collects and counts the valid ballot. EA finally declares the election result.

6 SECURE BENCHMARK ON UNRIZKNOW

In this section, we answer our final research question RQ3 by demonstrating the practical application of EV scheme to benchmarking system using Hi10 scheme (Hirt, 2010). We present the model, set-up, response submission, benchmark calculation using the EV approach. The aim of this section is to present how we can conduct a secure benchmark on **UnRizkNow** platform. The members of UnRizkNow are information security practitioners who possess knowledge about their organization regarding people, process, and technology. We use the cryptography tools mentioned in (Hirt, 2010) to establish our model.

6.1 Preliminaries

$\Sigma-$**proofs**- A $\Sigma-$proof is a three-move special honest-verifier zero-knowledge proof of knowledge. A $\Sigma-$proof is called *linear* if the verifier's test predicate is linear, i.e., the sum of two accepting conversations is accepting as well. The details of $\Sigma-$proof s is given in section 2.1 of (Hirt, 2010). *BA* acts as a verifier in our benchmark model.

Identification Scheme - An identification scheme is an interactive protocol between two parties, a prover (benchmark submitter) and a verifier (Benchmark Administrator). If the protocol is successful, then at the end of the protocol the *BA* is convinced he is interacting with the *BS*, or more precisely, with someone who knows the secret key that corresponds to the prover's public key. For benchmark submitter identification, we assume an identification scheme where the identification protocol can be written as a linear $\Sigma-$proof . It is easy to verify that Schnorr's identification scheme (Schnorr, 1991) satisfies this requirement. The secret key of *BS* is denoted by z_v, and the corresponding public key by $Z_v = g^{z_v}$ for an appropriate generator g.

Designated-Verifier Proofs- A designated-verifier proof is a proof which is convincing for one particular (designated) verifier, but completely useless when transferred from this designated verifier to any other entity (Jakobsson et al., 1996). The requirements of the encryption function are drawn from (Hirt, 2010). A semantically-secure probabilistic public-key encryption function $E_Z : \mathbb{V}X\mathbb{R} \Rightarrow \mathbb{E}, (a,\alpha) \mapsto e$, where Z denotes the public key, $\mathbb{V}$ denotes a set of answers, $\mathbb{R}$ denotes the set of random strings, and $\mathbb{E}$ denotes the set of encryptions. The decryption function is $D_z : \mathbb{E} \Rightarrow \mathbb{V}, e \mapsto a$, where z denotes the secret key. It is also required to have E to be $q-$ invertible for a given $q \in \mathbb{Z}$. It implies that for every encryption e, the decryption a and the randomness α of qe can be efficiently computed. It is also required that there is a number $u \leqslant q$, large enough that $1/u$ is considered negligible (Hirt, 2001). Furthermore, we use modified ElGamal and Pailier homomorphic encryption function.

Modified ElGamal Encryption - A traditional ElGamal system with an encryption function E with the property: $E(M_1)XE(M_2) = E(M_1 + M_2)$.

Pailier Encryption - As mentioned in section 3.3 of (Hirt, 2010).

Re-encrypting and Proving Re-encryptions - A random re-encryption $e^{'}$ of a given encryption $e = E(a,\alpha)$ is an encryption with the same answer a, but a new (independently chosen) randomness $\alpha^{'}$. Such a re-encryption can be computed by adding a random encryption of 0 to e. The rest of the details can be obtained from (Hirt, 2010) by substituting vote v with answer a, .

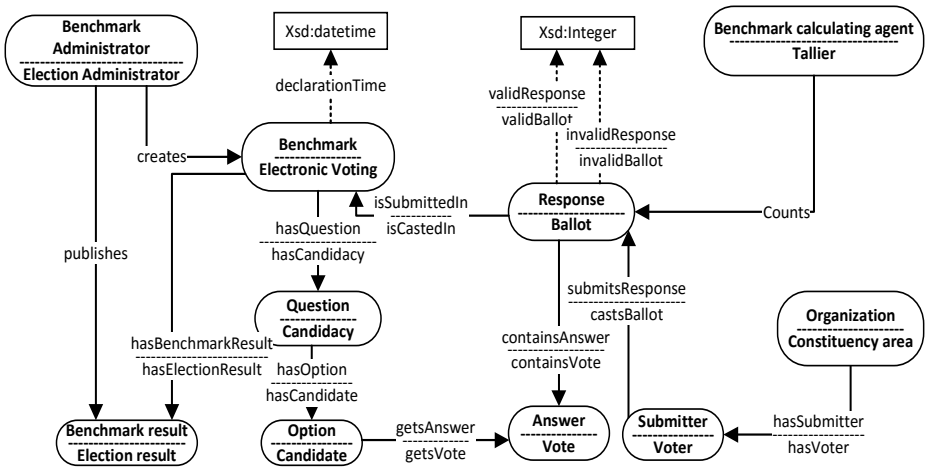


Figure 5: An ontology of benchmarking system and electronic voting system. The diagram shows that the concepts, actors, phases of benchmarking system can be mapped to electronic voting system.

6.2 Details of The Benchmark Protocol

We use the non-receipt free K-out-of-L voting protocol of (Hirt, 2010) to establish our benchmark protocol. Figure 6 shows the various steps involved in carrying out the benchmark on UnRizkNow platform.

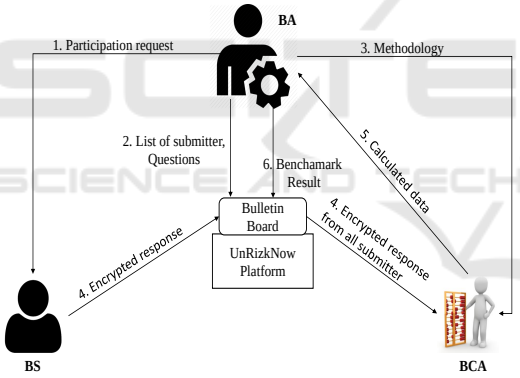


Figure 6: An overview of the benchmark model on UnRizkNow portal.

Model - We use the benchmark entities as mentioned in the section 5.1. The communication among the benchmark entities happens through UnRizkNow platform. The platform has a bulletin board to post any announcement. *BS* post their encrypted response on the bulletin board with their signature. This also prevents re-submission of the responses on the bulletin board. Anyone can read and verify the posted response on bulletin board, but nobody can delete from. The bulletin board can be considered as an authenticated public channel with memory. The communication channel between *BA* and *BS* is secured using TLS. A threshold t denotes the lower bound of the number of authorities that is guaranteed to remain honest during the protocol.

Benchmark Structure - The structure of question follows the structure mentioned in the section 2.2. We assume that we have yes/no, multiple choice, and open question (numerical) in the benchmark. A sample of the list of questions is given in Appendix. The mapping of the question to candidacy is performed as mentioned in the mapping section 5.2. The UnRizkNow platform maintains a double array $a[x][y]$ to save the label for the question format, and bit requires to generate the option for the question. The labels are *yn* for yes/no question, *mc* for multiple choice question, and *op* for the open numerical question. The submitter sees the questions in the form as presented in Appendix. The platform has a program module m that reads the value from the array $a[x][y]$ and takes care of the translation of option o to the required bits.

Benchmark Administration - N calculating agents $(BCA_1, ...BCA_N)$ execute the key generation protocol using ElGamal encryption scheme. The resulting public key of the benchmarking system is announced to the registered members of UnRizkNow community, and the corresponding secret key is shared among *BCA*. BA also publishes the questions, and response format on the bulletin board of UnRizkNow.

Benchmark Submission - Benchmark submitter constructs a random encryption $\overrightarrow{e} = E(\overrightarrow{a}, \overrightarrow{\alpha})$ for his answer vector $\overrightarrow{a}$ and randomness $\overrightarrow{\alpha} \in_R \mathbb{R}^K$, and posts it onto the bulletin board of UnRizkNow. The submitter also posts a proof of validity. A response $B = \overrightarrow{a} = (a_1, ...a_K)$ is valid if and only if $a_i \in \{0,1\}$ for $i = 1, ..., K$ and $\Sigma a_i = K$. A validity proof for the encrypted response $\overrightarrow{e} = (e_1, ..., e_K)$ is also constructed. The details of the construction of validity proof is given in section 5.4 in (Hirt, 2010). The encrypted response is submitted by *BS* to the bulletin board of UnRizkNow.

Benchmark Calculation - *BCA* collects the encrypted responses from the bulletin board. The benchmark result Π is performed for each question separately. For Question Q_i, the $i-$th components of each valid encrypted response from M submitters are summed up using the homomorphic property of the encryption scheme and decrypted using the verifiable decryption protocol of the encryption scheme.
Benchmark Result - The result of the benchmark is published for the individual questions. The result is calculated according to the equation 1, 3, 5.

7 DISCUSSION

In this section, we answer the final research question RQ4. Firstly, we address the adversary model and the security assumption that we considered in this study. The adversary model highlights what are the capabilities of an adversary. Secondly, we perform the security analysis on the main security requirements which are typical for benchmarking systems. Finally, we mention the behavior of the benchmarking system using EV concepts towards the considered adversary model.

7.1 Adversary Model and Trust Assumption

The adversary model depicts the attack potential that is a measure of the minimum effort to be expended in an attack to be successful (Idrees et al., 2014). The behavior of an adversary can change largely according to the implemented protocols and the capabilities of the adversary. An *internal* attacker is equipped with cryptographic keys and credentials that enable them to participate in the execution of the processes in the system. An *external* attacker does not possess such keys and credentials. In this section, we provide a general model of the adversary for benchmarking system and EV system and map them.

In our adversary model, *BS*, *BCA*, and *BA* can act as an internal attacker to break the system secrecy, but not to influence the election outcome via bribery or coercion. We assume that all the parties involved in the benchmarking scenario are polynomially bounded and thus incapable of solving hard problems or breaking cryptographic primitives such as contemporary hash functions. Adversaries cannot efficiently decrypt ElGamal ciphertexts without knowing the private keys. For preparing and conducting a benchmark event, as well as for computing the final result, we assume that at least one honest benchmark authority does not collude. We take into the consideration that dishonest *BCA* may collude with the adversary, but not all of them in the same benchmark event. A threshold t denotes the number of *BCA* that is required to decrypt the responses, and which is also able to break the secrecy of an answer. *BS* cannot create an invalid response that can pass the validity proof. An external or internal adversary cannot delete any content from the UnRizkNow bulletin board.

7.2 Fulfillment of the Security Requirements of Benchmarking System

In this section, we show how the security requirements of the benchmarking system stated in section 2.3.3 can be fulfilled by adopting the EV approach. We establish the security of our proposed benchmark model using the established security proofs from the electronic voting scheme. We utilize the security proof and concepts given in (Hirt, 2010).

1. **Completeness:** The dishonest submitter BS_i may create an invalid response, but the probability that the validity proof of encrypted response is negligible. Therefore, the invalidity of the encrypted response is detected in the validity proof of the scheme and the invalid vote will not be counted.
2. **Uniqueness:** The encrypted response along with the proof of the validity is posted on the bulletin board of UnRizkNow platform. Therefore, the submitter can submit only once, and the double submission is detected easily.
3. **Universal Verifiability:** Anyone can read the encrypted response posted on the bulletin board. One can check its validity by verifying the K-out-of-L encryption proof. Since the encryption function uses the homomorphic property, he can also sum up all valid encrypted response to obtain the encryption of sum of the answers. Since the decryption is verifiable, he can also check whether the sum of the answers has been correctly decrypted (Hirt, 2001), (Hirt, 2010).
4. **Individual Verifiability:** The individual verifiability of the benchmarking system is guaranteed by the homomorphic property of the encryption function and the verifiable decryption of the encryption scheme (Hirt, 2001), (Hirt, 2010).
5. **Eligibility:** The eligibility of the benchmarking system is ensured by the use of Schnorr's identification scheme. It is essential that each submitter know his secret key, and this is ensured by the public-key infrastructure. A protocol for ensuring knowledge of the secret key for Schnorr's iden-

tification scheme is provided in (Hirt and Sako, 2000).

6. **Secrecy:** The secrecy of the benchmarking system is guaranteed under the assumption that no t *BCA* can maliciously pool their information and the assumption that the encryption scheme is semantically secure.
7. **Soundness:** The soundness of the benchmarking system can be proved using the proofs given in the re-encrypting and proving re-encryption of (Hirt, 2010).

8 LIMITATION AND FUTURE WORK

The security requirements of benchmarking system are formulated mainly to address the *secrecy* of the sensitive information shared by the benchmark submitter and the *transparency* of the benchmark process. There could be an extra requirement of receipt-freeness for an enhanced version of benchmarking system. Receipt-freeness property ensures that the submitter cannot prove to a third party that they submitted a particular set of responses. A secure electronic voting scheme usually addresses the receipt-freeness requirement because the selling of the vote is a serious problem in the election. The selling of the vote is often initiated by the entity who wants a certain candidate to win in the election. However, in our benchmark model, we do not think this problem is widespread as there is no candidate involved in it. However, the significance of this requirement needs further investigation by producing a use-case scenario.

The mapping of the benchmark structure to EV system uses K-out-of-L voting structure. We constructed a response as an answer vector $\overrightarrow{a}$ where $a_i \in \{0,1\}$. The benchmark result is constructed by adding the $i-$th components of each valid response using the homomorphic property of the encryption function. Therefore, it is not possible to get the actual number entered by the submitter in the open numerical question as we cannot combine all the answers in the response and decrypt it. The system can apply homomorphic operation on the $i-$th bit of the answer. This property helps to ensure the confidentiality of the answer submitted, but at the same time, it does not allow to get all the actual numbers submitted by *BS*. The presence of an actual number in the benchmark could help to create a distribution graph of all the value submitted. In other words, it would provide how many submission lies below and above his submission. However, in our proposed model, one can only see if his performance is either below or above the average performance.

Our proposed solution is still prone to a vulnerability of conflicts of interests in and incentives to manipulate the benchmark process where the benchmark submitters are also the market participants with stakes in the level of the benchmarks. The conflicts in the interest can create an incentive for abusive conduct of the benchmark process. Benchmark submitters may attempt to manipulate a benchmark by submitting false or misleading data to break the credibility of the benchmark result. Our future work will consist of conducting a risk analysis of the benchmarking system. We will adopt CIRA method (Agrawal and Szekeres, 2017) to conduct the risk analysis exercise. The aim of this exercise will be to assess the conflict in the interest of the stakeholders involved in the benchmarking system and propose the treatment plan to reduce the conflict.

The EV schemes and system that we analyzed in this study is far from the complete list. There might be more relevant EV schemes and system available that can be suitable for our benchmark model on UnRizkNow platform. As our future work, we would like to implement different electronic voting schemes on the UnRizkNow platform and test their performance in the benchmark context. We are also interested in conducting similar studies with Group Signature, the Secure Multi-Party computation to analyze their role in conducting secure benchmark on UnRizkNow.

The ontology of benchmark and electronic voting presents an overview of the concepts and relationships involved in the system. The ontology needs to be formalized with Web Ontology Language (OWL) for modeling the ontology. The formal ontology will enable the possibility to be used by an automated tool to perform the mapping between benchmark and EV.

The future work also includes the assessment of other EV schemes to conduct secure benchmark on UnRizkNow. For instance, the LE02 schemes also meet all the requirements of secure benchmark. Therefore, Le02 can also be a good candidate to adopt for a future secure benchmark solution. However, there is a concern with the efficiency of the LE02 scheme. This scheme has the overall performance complexity of $O(xL^2B)$ where B represents the number of bits used to store one group element, x represents the number of questions, and L is the number of bits in the answer. In other words, every submitter sends his encrypted response using (xL^2B) bits. On the other hand, the overall performance complexity of Hi10 scheme is $O(xLB)$. In other words, every submitter sends his encrypted response using the (xLB) bits.

9 CONCLUSION

We have presented the model of a benchmarking system that is typically used by an organization to establish the benchmark standard and provide benchmark as a service. We highlighted the security challenges that the current benchmark model face, and therefore, a need to develop more secure benchmarking system is also justified. The security limitation of current benchmarking systems may hinder sharing of important information between the submitters and the benchmark authorities. Therefore, the requirements of a secure benchmarking system are established. We proposed a novel approach to solving the security limitation of benchmarking systems by adopting the secure cryptographic proofs from the field of secure electronic voting. We demonstrated how a benchmarking system could be mapped to the electronic voting system by mapping its protocol, structure, and concepts. We also demonstrated how the different formats of benchmark question can be presented and how the benchmark result can be calculated using the concepts of electronic voting. Our solution is based on the electronic voting protocol that provides secure transmission of the benchmark responses throughout the system. Furthermore, the identity of the response submitter is preserved by secrecy provided by the cryptographic protocols. The members who participate in the benchmark process can ensure that their responses have been counted correctly while calculating the benchmark result. Afterward, we demonstrated that how a secure benchmark can be designed for UnRizkNow platform using the concepts of EV system. We showed that a benchmarking system is more secure if it follows EV system approach as it can satisfy the necessary security requirements. We adopted Hi10 scheme to demonstrate the feasibility of our approach for UnRizkNow platform, but other relevant EV schemes can also be adapted to perform the benchmark on UnRizkNow platform.

REFERENCES

ABB (2017). Cyber security benchmark.

Agrawal, V. (2016). Towards the ontology of iso/iec 27005: 2011 risk management standard. In *HAISA*, pages 101–111.

Agrawal, V. and Snekkenes, E. A. (2017). *Factors Affecting the Willingness to Share Knowledge in the Communities of Practice*, pages 32–39. Springer International Publishing, Cham.

Agrawal, V. and Szekeres, A. (2017). Cira perspective on risks within unrizknow - a case study. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 121–126.

Agrawal, V., Wasnik, P., and Snekkenes, E. A. (2017). Factors influencing the participation of information security professionals in electronic communities of practice. In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 50–60.

Chaum, D., Ryan, P. Y. A., and Schneider, S. (2005). A practical voter-verifiable election scheme. In *Proceedings of the 10th European Conference on Research in Computer Security*, ESORICS'05, pages 118–139, Berlin, Heidelberg. Springer-Verlag.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90.

Chen, C.-L., Chen, Y.-Y., Jan, J.-K., and Chen, C.-C. (2014). A secure anonymous e-voting system based on discrete logarithm problem. *Applied Mathematics & Information Sciences*, 8(5):2571.

Cortier, V., Galindo, D., Glondu, S., and Izabachene, M. (2014a). Election verifiability for helios under weaker trust assumptions. In *European Symposium on Research in Computer Security*, pages 327–344. Springer.

Cortier, V., Galindo, D., Glondu, S., and Izabachène, M. (2014b). *Election Verifiability for Helios under Weaker Trust Assumptions*, pages 327–344. Springer International Publishing, Cham.

ESMA-EBA (2013). Final report:esma-eba principles for benchmark-setting processes in the eu. Technical report.

Forum, I. S. (2017). Benchmark as a service - information security forum. https://www.securityforum.org/products-services/benchmark-as-a-service/. Online; accessed 28 November 2017.

Gerlach, J. and Gasser, U. (2009). Three case studies from switzerland: E-voting. *Berkman Center Research Publication No*, 3:2009.

Gregor, S. and Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Q.*, 37(2):337–356.

Gritzalis, D. A. (2002). Principles and requirements for a secure e-voting system. *Comput. Secur.*, 21(6):539–556.

Haenni, R., Koenig, R. E., Locher, P., and Dubuis, E. (2017). Chvote system specification. *IACR Cryptology ePrint Archive*, 2017:325.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Q.*, 28(1):75–105.

Hidalgo, A. and Albors, J. (2008). Innovation management techniques and tools: a review from theory and practice. *R&D Management*, 38(2):113–127.

Hirt, M. (2001). *Multi Party Computation: Efficient Protocols, General Adversaries, and Voting*. Hartung-Gorre.

Hirt, M. (2010). Towards trustworthy elections. chapter Receipt-free K-out-of-L Voting Based on Elgamal En-

cryption, pages 64–82. Springer-Verlag, Berlin, Heidelberg.

Hirt, M. and Sako, K. (2000). *Efficient Receipt-Free Voting Based on Homomorphic Encryption*, pages 539–556. Springer Berlin Heidelberg, Berlin, Heidelberg.

Idrees, M. S., Roudier, Y., and Apvrille, L. (2014). Model the System from Adversary Viewpoint: Threats Identification and Modeling. EPTCS 165, 2014, pp. 45-58. arXiv:1410.4305v1.

IOSCO (2013). Principles for financial benchmarks. Technical report.

ISF (2017). The isf benchmark and benchmark as a service. https://www.securityforum.org/tool/the-isf-benchmark-and-benchmark-as-a-service/. online; accessed 19 November 2017.

Jakobsson, M., Sako, K., and Impagliazzo, R. (1996). Designated verifier proofs and their applications. In Maurer, U., editor, *Advances in Cryptology — EUROCRYPT '96*, pages 143–154, Berlin, Heidelberg. Springer Berlin Heidelberg.

Johannesson, P. and Perjons, E. (2014). *An introduction to design science*. Springer.

Kanno, Y. (2009). Information security measures benchmark (ism-benchmark). Technical report, T Security Center, Information-technology Promotion Agency (IPA).

Kelessidis, V. (2000). Innoregio: dissemination of innovation management and knowledge techniques.

Lee, B. and Kim, K. (2000). Receipt-free electronic voting through collaboration of voter and honest verifier. In *Proceeding of JW-ISC2000*, pages 101–108.

Lee, B. and Kim, K. (2003). Receipt-free electronic voting scheme with a tamper-resistant randomizer. In *Proceedings of the 5th International Conference on Information Security and Cryptology*, ICISC'02, pages 389–406, Berlin, Heidelberg. Springer-Verlag.

Liaw, H.-T. (2004). A secure electronic voting protocol for general elections. *Comput. Secur.*, 23(2):107–119.

Madise, Ü. and Martens, T. (2006). E-voting in estonia 2005. the first practice of country-wide binding internet voting in the world. *Electronic voting*, 86(2006).

Mathwick, C., Wiertz, C., deRuyter, K., served as editor, J. D., and served as associate editor for this article., E. A. (2008). Social capital production in a virtual p3 community. *Journal of Consumer Research*, 34(6):832–849.

Neff, C. A. (2001). A verifiable secret shuffle and its application to e-voting. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, CCS '01, pages 116–125, New York, NY, USA. ACM.

of Estonia, S. E. O. (2017). General framework of electronic voting and implementation thereof at national elections in estonia.

O'Rourke, L., Board, N. R. C. U. T. R., Program, N. C. F. R., of Transportation. Research, U. S. D., and Administration, I. T. (2012). *Handbook on Applying Environmental Benchmarking in Freight Transportation*. English short title catalogue Eighteenth Century collection. Transportation Research Board.

Pierro, M. D. (2017). evote tutorials.

Rjašková, Z. (2002). Electronic voting schemes. *Diplomová práca, Bratislava*.

Schnorr, C. P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174.

Schoenmakers, B. (1999). *A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting*, pages 148–164. Springer Berlin Heidelberg, Berlin, Heidelberg.

Vinkel, P. (2012). *Internet Voting in Estonia*, pages 4–12. Springer Berlin Heidelberg, Berlin, Heidelberg.

Whitman, M. and Mattord, H. (2014). *Management of information security*. Cengage learning.

Wiertz, C. and de Ruyter, K. (2007). Beyond the call of duty: Why customers contribute to firm-hosted commercial online communities. *Organization Studies*, 28(3):347–376.

APPENDIX

List of Benchmarking Questions

1. Do you perform background checks on all employees with access to sensitive data, areas, or access points?
   - Yes
   - No
2. What percentage of the employee recognize a security issue? [range 0-100]
3. Where do you store your sensitive information?
   - laptop
   - Paper document
   - Data server (internal)
   - Data Server (external)