International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 02 | Feb-2016 www.irjet.net p-ISSN: 2395-0072

A Usable Android Application Implementing Distributed Cryptography for Election Authorities

Ms. Minal Chandrakant Landge[1], Ms. Sabiha Jainuddin Subhedar[2], Ms. Anuja Arvind Patil[3], Ms. Neha Laxmanrao Koli[4] Prof. Chandrashekhar Shankar Shinde[5]

[1]Department of Computer Science and Engineering, Student at Dr. J. J. Magdum College of Engineering, Jaysingpur, Maharashtra, India.
minallandge1993@gmail.com

[2]Department of Computer Science and Engineering, Student at Dr. J. J. Magdum College of Engineering, Jaysingpur, Maharashtra, India.
sabihasubhedar123@gmail.com

[3]Department of Computer Science and Engineering, Student at Dr. J. J. Magdum College of Engineering, Jaysingpur, Maharashtra, India.
patilanuja11@gmail.com

[4]Department of Computer Science and Engineering, Student at Dr. J. J. Magdum College of Engineering, Jaysingpur, Maharashtra, India.
kolineha8@gmail.com

[5]Department of Computer Science & Engineering
Dr. J. J. Magdum College of Engineering, Jaysingpur, Maharashtra, India.
csshinde7769@rediffmail.com

Abstract:

The advancement in the mobile devices, wireless and web technologies given rise to the new application that will make the voting process very easy and efficient. The e-voting promising the possibility of convenient, easy and safe way to capture and count the votes in an election. This paper provides the specification and requirements for E-Voting using an Android platform. The e-voting means the voting process in election by using electronic device. We also described how the android mobile phones are efficient and can be used for voting. The android platform is used to develop an application. Using the API's provided by the android SDK (software development kit) the login can be done very efficiently. Many electronic voting protocols have been proposed, their practical application faces various challenges. One of these challenges is, that these protocols require election authorities to perform complex tasks like generating keys in distributed manner and decrypting votes in a distributed and verifiable manner.

Key Words: *Applied Cryptography, Distributed Key Generation, Electronic Voting, Understandability, Usability.*

1. INTRODUCTION

Voting for any social issue is essential for modern democratic societies now a day. So it is becoming very important to make the voting process more easy and efficient. In other hand the rapid development in operating system of the mobile phones gives rise to the application development on the large scale. The main reason behind the tremendous development in android application development is that the android is an open source operating system. It means that the software developers can have customization rights. As well as the software development kit provides tools to build and run android applications. The paper will be describing the basic idea of the project E-voting system on android and its advantages, disadvantages and applications. The paper is divided in five parts. The first part describes the literature survey i.e. the previous work done on the voting process. Then the further parts will describe about the E-voting methodology, architecture, advantages, disadvantages and its applications.

1.1 LITERATURE SURVEY

A. *Electoral system in India*

The technology used in India for voting is Electronic voting machines. There are 2 systems developed for conducting an electronic voting machine. These are the DRE (Direct Recording Electronic) and Identical Ballot Boxes. A DRE voting system records votes by means of an electronic display provided with mechanical or electro-optical components that can be activated by the voter, that processes voter selections by means of a computer program, and that records that processed voting data in memory components. It produces a tabulation of the voting data that is stored in a removable memory component and may also provide printed renditions of the data. The system may further provide a means for transmitting the processed vote data to a central location in individual or accumulated forms for consolidating and reporting results from precincts at a central location. DRE systems additionally can produce a paper ballot printout that can be verified by the voter before they cast their ballot.

B. Id*entical Ballot Boxes*

The Identical Ballot Boxes hold the ciphered vote, encrypted with the PMA voting key and the ciphered Identification Card Number, encrypted with their personal 4 digit key. It is designed to accept connections from the vote distribution server, and ensures an acceptable level of Security as far as remote vote manipulation is concerned. In the current version of the system, it has been implemented in SQL Server 2000. The connection the voting distributor, and the administration server is established through JDBC 3.0.

C. *Integrated Election Software package*

Integrated Election Software package, running on a Microsoft Windows computer, allows the election official to set up and record the details of an election. When voting is completed, it counts the votes and displays the outcome of the count results in the format Irish voters are familiar with. The PC's used are stand alone and security hardened for the election software only. Access to the PC's is also controlled by a security key.

1.2 ELECTION SCENARIO

Our project setting is close to real-world elections: we consider elections (or handle election districts) with one thousand voters, since 824 is an average number of postal voters in a district in Germany and five election authorities, because there must be at least five election authorities per electoral districts in Germany. Additionally, election authorities are citizens without special information security knowledge. Furthermore, we assume election authorities not to have (national) eID, as not all countries have issued eID cards, and even in those that have, the percentage of population possessing the eID is not big enough. We consider in our research project an abstract, yet widely implemented voting protocol based on El-Gamal encryption. In the election setup phase, an El-Gamal election key (together with the corresponding private key shares) is generated by the Election authorities with a corresponding distributed key generation protocol. During the vote casting, votes are encrypted with this public El-Gamal election key and submitted to a so called web bulletin board (WBB); i.e. a remote web server with a database connection with the public having read access and observers taking copies in order to observe whether the WBB is not deleting votes. There, the encrypted votes are stored together with some information identifying the voter.

We refer to these encrypted votes as personalized votes. In the tallying phase these personalized votes are anonymized using a verifiable re-encryption mix-net, e.g. the mix-net proposed in [16]. The choice of mix-net as a way to anonymize the votes has an impact on the efficiency of the decryption process.

This stems from the fact that as opposed to one cipher text decryption (as in the case of homomorphic sum tallying), all individual anonymized cipher texts need to be decrypted. Yet we decided to build upon the mix-net approach because it has been used in parliamentary elections in Estonia and Norway. The election authorities download the anonymized votes from the WBB and run the verifiable distributed decryption protocol for calculating the election result.

The goal of our project is to support election authorities during election setup and tallying for the described abstract voting protocol. We assume that they either meet in person or make phone/video calls in both phases, i.e. there is an out-of-band channel to interchange information. However, it might be that individual election authorities are not able to show up or no longer behave honestly. Therefore, we need to implement a threshold distributed key generation protocol and thus a threshold verifiable distributed decryption protocol, enabling the tallying of results even if only a threshold of election authorities is participating. For our particular setting we decided to choose a threshold of three, which we consider to provide an optimal trade-off between secrecy and robustness. For practicability, it was decided to develop a smartphone application, as these devices are nowadays widespread, and the number of people using them still grows4. Furthermore,
Smartphones are equipped with mobile Internet that can be used for communications in case setting up a wireless network is not possible or requires too much organizational effort.

1.3. ELECTION PROCEDURE

Below we describe the election setup and for the tallying phase with the developed application. The interfaces have been developed in an iterative process involving technical and non-technical potential election authorities, furthering the goal of making the application usable.

A. Setup Phase

Stage 1 - Tutorial: The election authorities get a tutorial on the functionality and security of the application and the Internet voting system.

Stage 2 - Announcing members and head: The election authorities announce their names and Jabber IDs. They also announce the head (after having discussed who should take the duties of the head). The administration staffs of the WBB announce this information via the WBB.

Stage 3 - Installing the application: The election authorities download the application from the trusted public institution (TPI) they trust via secure HTTPS connection, or from the Google Play Store, if this institution is registered as a certified developer, and install it on their smartphone.

Stage 4 - Starting the application: The election authorities start the application and enter their Jabber credentials, and a WBB web address provided to them by the election organizers(See Figure 1(a)).

Stage 5 - Communication key exchange: After a successful login into the application (i.e. the election URL and the credentials are correct), the application displays the name of the election, as well as the name of the head of the election authorities as announced on the WBB. The election authorities have to confirm that they agree to participate in this election with the named person being the head of election authorities. After the confirmation, a random RSA key pair is generated by each of the applications. The public keys are exchanged with all other authorities. At the end of this stage, each smartphone displays a passphrase (according to the protocol mentioned in Section III-B) (see Figure 1(b)). The instructions on the screen for the head are to read aloud this passphrase. The instructions for the others are to compare and only continue if all have the same passphrase. In case the verification was successful, the authorities may continue to the next step.

Stage 6 - Distributed (election) key generation: In this stage, the head of the commission initiates the distributed key generation protocol. This is done by inviting the other election authorities (by selecting them from the displayed list). At the same time, the

other authorities wait for receiving an invitation. Once they have received the invitation, the list of invited authorities, as well as the name and threshold value for the election is displayed (see Figure 1(c)). Then, they are asked whether they agree on this list and on the threshold. This is necessary to avoid the head inviting other people then the ones he is supposed to invite, or setting the wrong threshold value. After all election authorities accepted the invitation (as no misbehavior was detected), the distributed key generation protocol is executed. After its completion, each authority's smartphone sends a generated public key to the WBB.

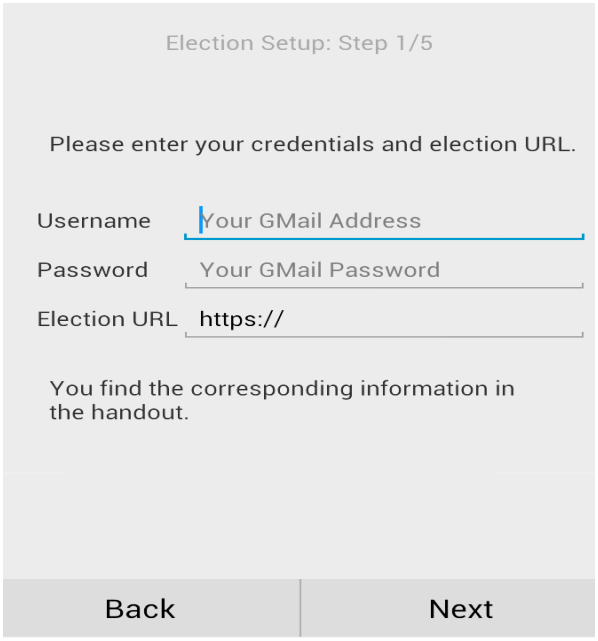


Fig1 (a). Login.

Fig 1(b). Verification of communication keys

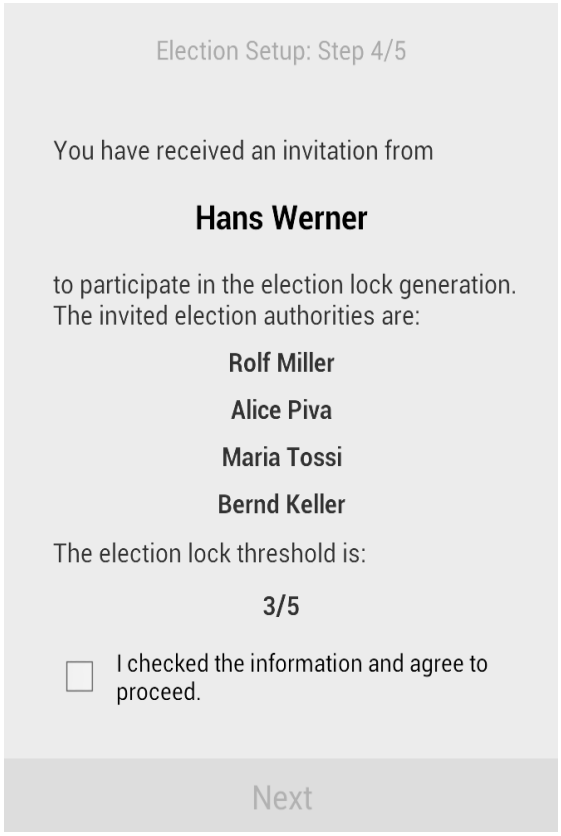


Fig1 (c). Accepting invitation to participate in distributed key generation

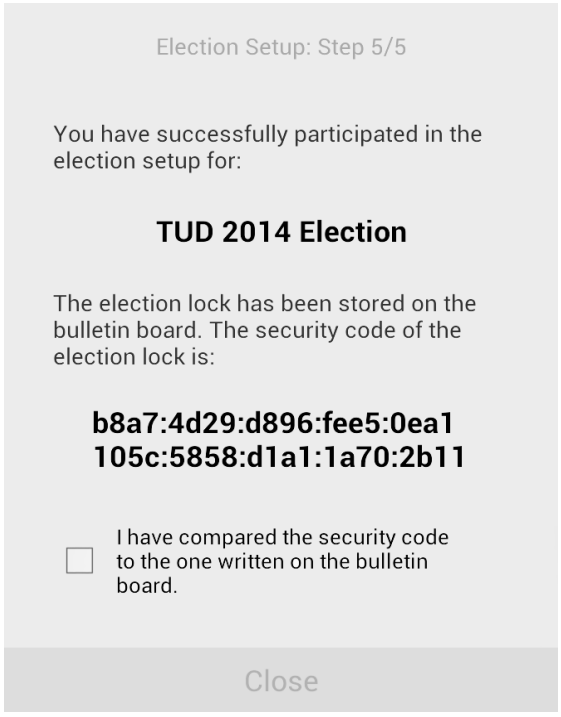


Fig 1(d). Verification of election key.

Stage 7 - Verifying (election) key generation: The smartphone displays the hash value of the public key it computed during the distributed key generation protocol. The instructions for all election authorities are to compare the displayed value with the one displayed on the WBB. If the value matches and all authorities confirm the matching, this key is loaded into the voting application to be used to encrypt votes. The election authorities may now close the application.

B. Tallying Phase

Stage 1 - Starting the application and running the distributed verifiable decryption: The election authorities start the application and all anonymized votes are downloaded from the WBB. The head initiates the decryption by inviting all other election authorities to participate. At the same time the other election authorities wait for this invitation. Once all election authorities accepted the invitation14, the decryptions started. The result of its successful completion is a list of decrypted votes, the number of votes per candidate together with corresponding proofs. The result is sent to the WBB from the head. For increasing robustness, the authorities are then encouraged to create a backup of the data generated by the application by exporting the data to external storage.

Stage 2 - Announcing the results: The result is displayed on the WBB but also by each application of the election authorities.

The election authorities compare the result displayed on their smartphones with the one displayed by the WBB. They then close the application and in case they want to get access to the result again they just open the application again.

1.4. SECURITY PROPERTIES AND SECURITY MODEL

In order to check whether the authorities understand the security properties and security model of the application, it is important to be very precise about both. The identified properties relevant for the Internet voting system – namely robustness, secrecy, and integrity and the corresponding assumptions are determined and discussed in this section. For the analysis we consider each of the entities involved with the proposed application (each election authority and the WBB) as well as outsider attackers (reading, deleting, modifying messages that are interchanged between the involved entities) as well as combinations being able to violate each of these three properties. We first consider each election authority as one unit including the person, the smartphone, the operating system, and the application. We later discuss the different parts and their impact on the security model. Robustness means that it is possible to decrypt the anonymized votes from the WBB. This is ensured if at least a threshold amount of election authorities is available and behaves correctly during the distributed decryption protocol. In addition, it needs to be assumed that a communication network with enough throughputs is available, and we rely on the used cryptographic primitives in place. Secrecy means that it should be impossible to decrypt the personalized, encrypted votes. This is ensured if at least a threshold amount of election authorities is honest and behaves correctly in any stage. This includes that they check the passphrase and that the hash value of the election key displayed on their smartsphone and the one displayed on of the WBB match. In addition, secrecy can currently only be ensured if the WBB is trustworthy (if not, the WBB could send the personalized votes and would obtain the corresponding decrypted ones). From the voting system's components we need to trust that at least one mix node is honest, and, finally, we rely on the cryptographic primitives in place.

Integrity means that it should be detected if the decrypted and anonymized votes do not match. This is ensured if at least a threshold amount of election authorities is honest and behaves correctly at any stage. This includes that they check that the proper result is published on the WBB. In addition, integrity can currently only be ensured if the WBB is trustworthy. Finally, we rely on the used cryptographic primitives. Different 'Parts' of the Election Authority. So far we only considered each election authority as a unit. However, actually it is not necessarily the person herself who might be dishonest and for instance not join the decryption phase or export the key share; it might also be the application which is not implemented by the election authority due to the necessary lack of technical knowledge; thus the downloaded application could be malicious in arbitrary ways. Therefore, it is recommended to have different trustworthy institutions programming their own application. Then each election authority downloads the application from a different institution. With this, the above security model also holds on the application level. The same problems also exist on the operating system and the smartphone manufacturer level. However, here it is not easy to find a solution matching the above mentioned security model. As the election authorities are supposed to use their own smartphones, we rely on whatever smartphones and operating systems they have. Thus, the security model on these levels actually depends on the concrete setting of an election, i.e. which smartphones and which operating systems are used by the election authorities.

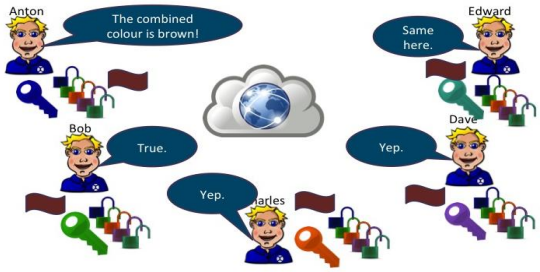
Trustworthiness of the WBB.As seen, the current implementation builds upon the trustworthiness of the WBB. We assume, however, that the security of the WBB is taken care of outside of our application in real-world elections, especially the high-stake ones. As such, for ensuring integrity and secrecy of the votes, the PKI established among the voters and the mix nodes can be used. In such a case, for example, for ensuring secrecy the election authorities could be directed to decrypt the votes only if these votes are signed by a threshold amount of mix nodes. Furthermore, the WBB can be constantly supervised by a trusted third party in order to ensure that it does not arbitrarily change its content, thereby ensuring integrity.

1.5EDUCATION MATERIAL

Transparency is an important property of elections as for instance the German constitutional court stated referring to the use of voting machines in Germany. In recent elections, most constituencies' provide education material for poll workers, such as for instance the state of Berlin. This material proves to be particularly crucial for Internet elections, since the election authorities in Internet elections usually have a high education, but are not specialists in computer science or information security. We therefore prepared, in addition to the application itself, training for election authorities. This training aimed for communicating the motivation for this application and a basic

understanding of the application's objectives and security properties (including the underlying security model). We use a metaphoric approach like many others in security education literature. The challenge is to select appropriate metaphors, i.e. the election authorities build a mental model on how the application works which allows them to deduce the security properties and the security model properly.

(a) Locks, keys and seals**.**



(b) Verification of communication keys.

(c) Partially opening encrypted votes.

Fig2. Metaphors used for cryptography in the education material**.**

Verification of communication keys: The exchanged public keys must be validated by the authorities by comparing a hash value (of all other public keys). In order to avoid explaining second pre-image resistance of cryptographic hash functions, we use the following metaphor: First, authorities exchange colored public locks. Second, after all colored locks have been exchanged; the mix color of all lock colors is obtained. The mix color represents the outcome of the second pre-image resistant hash function applied to the concatenation of all public keys (see Figure 2(b)). This metaphoric approach has been used to communicate the concept of one-way functions used to teach Diffie-Hellman secret sharing. Due to the conceptual similarity of one-way functions and second pre-image resistant hash functions, we consider the color metaphor to be a proper approach.

Cryptographic key sharing: Many works, see for instance, rely on the idea of splitting keys physically apart as metaphor for cryptographic key sharing. However, this concept does not work for schemes being robust against single entities being malicious. Therefore, we introduced the concept of each share being part of a description used to build the key. As these descriptions are not disjoint it is possible to reconstruct the key based on a subset of all shares. As it is necessary to build the key, we also explain that a computer program is necessary to be involved and the lock is stored as key information in a file.

Distributed decryption: One particular challenge of our approach, as opposed to the Estonian Internet voting system, has been the explanation of distributed decryption. We discussed several approaches and decided eventually that each election authority holding a key reconstruction description opens the locked envelope partially on a specific position.
Positions of different authorities overlap each other in order to communicate robustness of the protocol. If an envelope is partially opened by a sufficient (threshold) amount of election authorities, the envelope can be opened on the WBB and the election result can be computed (see Figure 2(c)).

1.6 Algorithm of Distributed Key Generation:

In this section, we survey the concepts of VSS and DKG.

Verifiable Secret Sharing: The notion of secret sharing was introduced independently by Shamir and Blakley. Since then, it has remained an important topic in security research.

Definition :(n; t+d; t)-Secret Sharing. For integers n, t and d such that n>=t+d>t>=0, an (n; t+d; t)-secret sharing scheme is a protocol used by a dealer to share a secret s among a set of n nodes in such a way that any subset of t+d or more nodes can compute the secret s, but subsets of size t or fewer have no information about s.

A secret sharing scheme with d= 1 is called a threshold secret sharing scheme, and for d>1, it is called a ramp secret sharing scheme. In this work, we concentrate on threshold secret sharing and denote it as (n; t)-secret sharing instead of (n; t+1; t)-secret sharing. Note that all polynomial-based threshold secret sharing schemes can easily be converted to ramp secret sharing schemes. In some secret sharing applications, clients may need to verify a consistent dealing to prevent malicious behavior by the dealer. A scheme with such a verifiability guarantee is known as verifiable secret sharing (VSS) scheme. Feldman developed the first non-interactive and efficient VSS scheme and Pedersen presented a modification to it.

Definition: An (n; t)-VSS scheme consists of two phases: the sharing (Sh) phase and the Reconstruction (Rec) phase.
Sh phase: A dealer Pd distributes a secret s 2 K among n nodes, where K is a sufficiently large key space. At the end of the Sh phase, each honest node Pi holds a share si of the distributed secret s.
Rec phase: In this phase, each node Pi broadcasts its secret share si and a reconstruction function is applied in order to compute the secret s = Rec(s1; s2; : : : ; sn) or output indicating that Pd is malicious. For honest nodes si= si, while for malicious nodes si may be different from si or even absent.

It has two security requirements: Secrecy (VSS-wS): A t-limited adversary who can compromise t nodes cannot computes during the Sh phase.
Correctness (VSS-C): The reconstructed value should be equal to the shared secret s or every honest node concludes that Pd is malicious by outputting.

We consider VSS schemes in the computational complexity setting. Here, any malicious behavior by Pd is caught by the honest nodes in the Sh phase itself and the VSS-C property simplifies to the following: the reconstructed value should be equal to the shared secret s. Further, many VSS applications avoid participation by all parties during the Rec phase. It is required that shares from any t + 1 honest node (or any 2t + 1 node) is sufficient to reconstruct s. Therefore, we mandate the correctness property that we refer as strong correctness requirement.

Strong Correctness (VSS-SC): The same unique value s is reconstructed regardless of the subset of nodes (of size greater than 2t) chosen by the adversary in the Rec algorithm. Further, some VSS schemes achieve a stronger secrecy guarantee.

Strong Secrecy (VSS-S): The adversary who can compromise t nodes does not have any more information about s except what is implied by the public parameters. Asynchronous VSS: Although the literature for VSS has been vast, VSS in the asynchronous communication model has not yet received the required attention. Canetti and Rabin developed the first complete asynchronous VSS scheme with unconditional security. However, this scheme and its successor (n5) due to their bit complexities are prohibitively expensive for any realistic use. Compromising the unconditional security assumption, Cachin etal. (AVSS), Zhou et al. (APSS), and more recently Schultz et al. (MPSS) suggested more practical asynchronous VSS schemes. Of these, the APSS protocol is impractical for any reasonable system size, as it uses a combinatorial secret sharing scheme by Ito, Saito and Nishizeki, which leads to an exponential factor in its message complexity. MPSS, on the other hand, is developed for a more mobile setting where set of the system nodes has to change completely between two consecutive phasesto maintain the secrecy and correctness properties.

AVSS is the most general and practical schema in asynchronous communication model against Byzantine adversaries, but it does not handle crash recoveries. It assimilates a bivariate polynomial into Bracha's reliable broadcast and can provide complete flexibility with the sets used without hampering the security. In asynchronous VSS, any two participants need to verify the dealer's commitment with each other to achieve correctness; thus, a protocol with o(n^2) message complexity does not seem to be possible. Therefore, AVSS, with its optimal message complexity, forms the basis for our Hybrid VSS and Hybrid DKG protocols.

2. CONCLUSION

This paper focused on the analysis of development of election authority application on an android platform. The usability of this system is very high if it will use in real life election process. It will definitely helpful for the users who wish to Vote and the voting process will be made very easy by using this application. The present work pursued two goals. The first goal was to develop a usable application for the tasks of election authorities that consists of distributed key generation and verifiable distributed decryption, thus furthering the implementation of electronic voting protocols that make use of these concepts. The second goal was to communicate the security model used in electronic voting protocols (with particular focus on our application) to laymen.

3. ACKNOWLEDGEMENT

It is our privilege to acknowledge with deep sense of gratitude towards our project guide, Prof. C. S. Shinde for his continuous guidance and encouragement throughout course of study and timely help given in the completion of our final year project work on "A Usable Android Application Implementing Distributed Cryptography For Election Authorities." It is needed a great moment of immense satisfaction to express out profound gratitude, indebtedness towards our H.O.D. Prof. Mrs. D. A. Nikam, whose real enthusiasm was a source of inspiration for us. We would also like to thank all other faculty members of Computer Engineering department who directly or indirectly kept the enthusiasm and momentum required to keep the work towards an effective project work alive in us and guided in their own capacities in all possible.

REFERENCES:

[1] Stephan Neumann, Oksana Kulyk, Melanie Volkamer"A Usable Android Application Implementing Distributed Cryptography for Election Authorities" Technische Universit¨at Darmstadt / CASED, Darmstadt, Germany, 8-12-2014, pp.207-216

[2] Estonian National Electoral Committee, "E-Voting System General Overview," 2010. [Online]. Available: http://www.vvk.ee/public/dok/ General Description E-Voting 2010.pdf

[3] B. Adida, O. De Marneffe, O. Pereira, and J.-J.Quisquater, "Electing a university president using open-audit voting: analysis of real-world use of helios," in Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections. USENIX Association, 2009, pp. 10–10.

[4] "International Association for Cryptologic Research 2012 Election,"http://www.iacr.org/elections/2012/, 2012.

[5] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," Journal of Cryptology, vol. 20, no. 1, pp. 51–83, 2007.

Author Profile

Ms.Minal Chandrakant Landge-
Completed 12th from L.G.R. Purohit College, Sangli, Pursuing BE degree in Computer Science and Engineeing from Dr. J.J. Magdum college of Engineering, Jaysingpur, Shivaji University, Kolhapur. Research interests are in Android Application.

Ms.Sabiha Jainuddin Subhedar-
Completed 12th from R.S.K College, Sangli, Pursuing BE degree in Computer Science and Engineeing from Dr. J.J. Magdum college of Engineering, Jaysingpur, Shivaji University, Kolhapur. Research interests are in Android Application.

Ms.Anuja Arvind Patil-
Completed 12th from K.W.C,Sangli,Pursuing BE degree in Computer Science and Engineeing from Dr. J.J. Magdum college of Engineering, Jaysingpur, Shivaji University, Kolhapur. Research interests are in Android Application.

Ms.Neha Laxmanrao Koli-
Completed 12th from Sangli Highschool, Sangli, Pursuing BE degree in Computer Science and Engineeing from Dr. J.J. Magdum college of Engineering, Jaysingpur, Shivaji University, Kolhapur. Research interests are in Android Application.

Mr.Prof. Chandrashekhar Shankar Shinde-
Perceived BE degree in Computer Science and Engg. from Walchand College of Engineering Sangli and completed M.Tech. degree from Shivaji University, Kolhapur, Currently working as an Assistant Professor in the Department of Computer Science & Engineering, Dr. J. J. Magdum College of Engineering, Jaysingpur. Research interests are in the System Programming, Water Marking.