Securing e-voting against MITM attacks

Dimitris Mitropoulos and Diomidis Spinellis
Department of Management Science and Technology
Athens University of Economics and Business
Email: {dds,dimitro}@aueb.gr

***Abstract*—Man in the middle attacks involve the interception and retransmission of electronic messages in a way that the original parties will presume that their communication is secure. Such an attack could be a threat to any electronic voting scenario. This paper proposes a novel method for preventing this kind of attacks by including in the transaction a challenge-response test. The human end-user is asked to vote through an image-based challenge that will foil a typical automated software-based attack. The image is crafted so as to include multiple challenge nonces as a way to select the user's vote. The approach's strength is based on the difficulty of malicious software to falsify the image or emulate the user's response.**

I. Introduction

The quintessence of an e-voting transaction is to be secure [1]. In the e-voting context, security issues are very subtle. This is because there are features that clash with each other. For example, guaranteeing anonymity, makes it harder to track election fraud [2], [1]. In addition, security in e-voting is highly related to the type of the technology used during the process. There are two basic forms of e-voting, namely: presence and distance [3]. The former takes place in a specific station, using an ad hoc machine and under the supervision of the election's administration. In distance e-voting, the voter can cast his vote from his personal computer by sending it to a central server via the internet. The electronic, network-based nature of the latter makes it susceptible to a wide range of attacks [4].

One of the most important and potentially damaging class of attacks that must be taken into account when designing and implementing secure e-voting systems are man-in-the-middle attacks (MITMAs) [5], [6], [7]. In a common MITMA, an intermediate party is placed between the client side and the server side wiretapping their communication and retransmitting messages as he chooses. During the attack neither side is aware that the private communication is being illegally monitored [6], [8], [9]. From the capturing of a session cookie to the altering of an online payment, MITMAs can cause considerable damage to both sides of a transaction.

PCI 2009: 13th Panhellenic Conference on Informatics, Corfu, Greece, September 2009.

Using CAPTCHAs[1] to defend network transactions is not a new idea [10]. At first, CAPTCHAs were introduced in order to prevent dictionary attacks and search engine bots [11]. But later they were also used to secure online transactions [12], prevent DoS attacks [10], fortify online gaming [13], enhance anti-spam protocols [14] and detect data input attacks [15].

In this paper we describe a technique that utilizes CAPTCHAs in conjunction with a *transaction authentication number* (TAN), to prevent MITMAs within the context of a network-based electronic voting transaction. Our proposed security layer fortifies the integrity of the user's vote on an end-to-end basis. Our approach does not deal with other aspects of e-voting security, such as voter anonymity and system availability [16], [17]. To prove the validity of our proposal, we focus on a simple e-voting transaction.

II. Attack Scenario

Consider a typical network-based, e-voting transaction. A legitimate voter connects to a remote server via her personal computer to vote for the elections. We assume that the MITM has successfully deceived the legitimate parties and we examine the transaction right before the vote casting. The goal for the attacker is to intercept the ballot and change the vote.

The attack can take place either locally or remotely. In the former case a downloaded Trojan horse could play the role of the MITM, hijacking an authenticated session [18].

In a remote attack the traffic is at first redirected to a rogue server via e-mail phishing or pharming. Another way to obtain packets intended for a legitimate server is by spoofing ARP or DNS responses to the client. Then the server run by the attacker, will act as proxy between the end user and the real website keeping the authenticated session active while modifying the transaction data [19].

The majority of internet transactions are made over HTTPS. Still, a MITM can impersonate each party to the satisfaction of the other over HTTPS [20]. Hence, our attack scenario is valid even for transactions secured over HTTPS.

As an example, consider a vote choosing between the "Little Endian" and the "Big Endian" political parties. In Figure 1 we show the steps taken by an attacker in order to alter a vote for the Little Endian party into a vote for the Big Endian party. When the voter sends the vote to be registered, the MITM replaces it with a different one. When the server sends back

[1] CAPTCHA stands for Completely Automatic Public Turing Test to Tell Computers and Humans Apart.

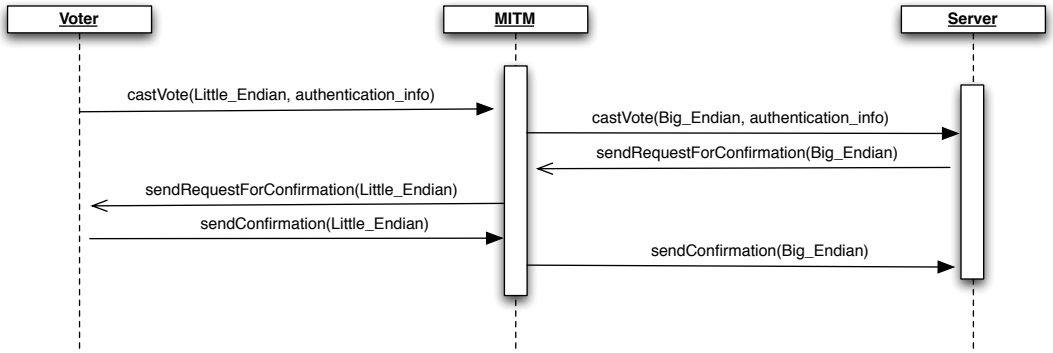

Fig. 1. Attack scenario.

a request of confirmation, the attacker conversely replaces the tokens and passes the request to the client. The ignorant user confirms the transaction leading to a successful attack.

III. Approach Description

Our approach is based on the idea of establishing an out-of-band end-to-end secure communications channel between the two transacting entities that is used for transaction signing. However, instead of establishing a physical secondary channel our scheme transfers over the existing channel data that the MITM software cannot easily decrypt or modify, ensuring the data's confidentiality and integrity. (The MITM can however generate a fake instance of such data; we discuss countermeasures in Section IV).

Initially, the users send to the server the vote's authentication details using a strong (e.g. two factor) authentication scheme—see Figure 2. The "two-factor authentication" is a system where two different factors are combined to verify a user [21]. Strong authentication is required, because otherwise a MITM can capture a user's credentials and rig the vote through a simple impersonation attack. Given the cost and the logistical problems of furnishing all voters with a hardware authentication token, the simplest way to provide two factor authentication is through a transaction authentication number (TAN) that will be furnished to all registered voters in a secure manner.

When the server receives the voter's details it verifies the transaction's authentication data; for instance, that the TAN matches the one expected for the specific voter. It then creates a challenge in the form of a CAPTCHA image. This contains a listing of the vote alternatives together with a nonce challenge for each one; see the example in Figure 3. The image is composed in a way that makes it very difficult for software to

1) decode the nonces' value, and
2) incorporate the nonces in a fake image.

The user reads the image with the nonces, selects the vote she wishes to cast and responds to the server's challenge by typing the corresponding nonce's value, which is then sent to the server. Finally, the server verifies that the nonce response

Fig. 3. An example of a MITM-resistant CAPTCHA.

is one of those offerred as the challenge, and, if it is, it casts the vote.

In a typical attack the MITM will intercept the user's vote request and substitute the details of a fraudulent vote for the vote the user requests. This allows the MITM to hijack the user's vote authentication data to cast a fraudulent vote. Owing to the way the image is constructed the MITM software is unable to decode a nonce and send it back to the server (property 1). Furthermore, the MITM software is also unable to construct an image containing false descriptions for the server's correct nonces (property 2), thus tricking the user to respond to the challenge.

IV. Weaknesses and Countermeasures

Our proposed scheme for protecting network-based e-voting can be attacked by deciphering the image or by placing a human accomplice in the loop. The complete tree of possible attacks and countermeasures appears in Figure 4. In all cases we assume that the malicious software is crafted to change a user's vote.

A. Image Processing Attacks

There are two possible attacks based on image analysis. In the first one, the malicious software doctors the image

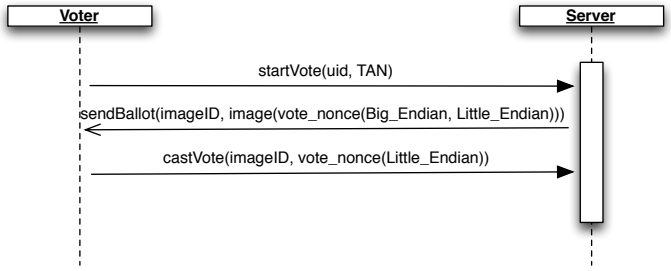

Fig. 2. A legitimate transaction with our proposal.

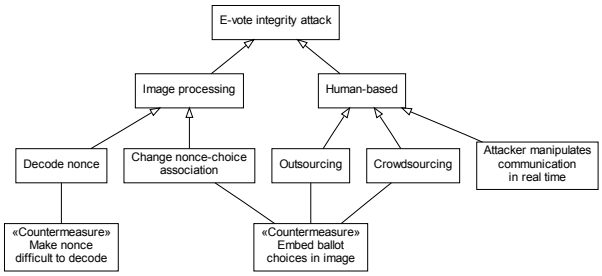

Fig. 4. Attack tree and countermeasures.

changing the association between the choices and the nonces to one that will favor the attacker. Since the attacker cannot know the voter's choice, this attack would entail putting the nonce of a known favorite choice next to the ballot choice the attacker wishes to promote. Our method hinders this approach by including in the image the names of the choices and a background pattern that makes it difficult to change any part of the image without destroying the pattern's continuity. In addition, we prevent locating the nonce in order to separate it by changing the font size of all letters and by adding a random number of spaces between words. These choices make the nonce appear in a place that is hard to determine in advance. In another attack the malicious software decodes the challenge and responds appropriately. It also presents the user with a new false ballot image that will be ignored. This attack can be resisted by hindering malicious software to locate the challenge nonces (using the methods we outlined), or recognize their text. Various methods have already been proposed for this; our prototype implementation uses various random fonts, font-sizes, and colors, and also skews, rotates and perturbs the location of each letter.

Both attacks we describe are based on image analysis. Considerable research has been published on the breaking of visual CAPTCHAs. Machine learning attacks against the Asirra CAPTCHA showed that the probability to break it is significantly high [22]. Microsoft CAPTCHAs are also prone to automated, low-cost attacks [23]. In addition, methods based on shape context matching seem to be another efficient way to identify words and phrases within CAPTCHAs [24]. Finally, reverse engineering CAPTCHA instances using simple image processing techniques is another proposed method [25].

B. Human-Based Attacks

An alternative, more devious form of attack involves putting an actual human into the loop. This can happen in three ways. First, the attackers can actively monitor e-vote transactions and respond to the challenge of a forged transaction. This however requires constant real-time monitoring and can be inconvenient, especially if voting takes place over a short period. Furthermore, it is a risky proposition, because it puts the attackers into synchronous direct contact with the malicious software, increasing the chance of tracing the attack back to them.

A human could in theory be put in the loop without directly involving the attackers. This can be done by outsourcing the nonce recognition either to a free crowdsourcing application [26] or to a payment-based system, such as the Amazon Mechanical Turk [27]. However, given that for the attack to work the nonces will have to be paired with candidate names, this attack is likely to arise suspicion among the humans performing the nonce recognition.

V. Related Work

In this section we cover two areas. In the first we discuss a variety of suggested solutions to counter MITMAs, and in the second we present a number of proposals that reinforce the security of e-voting.

A. Preventing MITMAs

A plethora of cryptographic techniques can be employed to prevent MITMAs. The first to be introduced was the *Interlock Protocol* [28]. To avert an eavesdropper attack this protocol presumes that the client and the server must use an anonymous key exchange protocol. Still, this protocol proved to be vulnerable when used for authentication [29].

Another cryptographic technique is the *password protection module* (PPM) [30]. For every transaction, a unique password is generated on the client side via hashing. The main flaw of this approach is that it speculates that the MITM would not be willing to perform a hashing in order to obtain a password for himself.

The Secure Sockets Layer (SSL) protocol and the Transport Layer Security (TLS) protocol are also designed in order to cryptographically protect communication channels between a client and a server. Based on the two aforementioned protocols, SSL/TLS *session-aware user authentication* is one of the most recent and promising proposals [6]. A mechanism that secures tunnelled authentication protocols is also related to this approach [31].

The Zurich Trusted Information Channel is another approach that utilizes SSL/TLS connections requiring minimal to no changes in both server and client-side [9]. The main disadvantage of this solution is its computational overhead.

Except for the cryptographic techniques, there are other ways to prevent MITMAs. One of them is channel hopping [32]. But this approach lacks implementation mechanisms that ensure its validity. Others involve the hardening of web browsers in a way that they will warn the user in case of a certificate verification error while visiting a secure site [19]. Furthermore, to secure the secret key exchanging scheme an approach that involves jigsaw puzzles was recently proposed [33]. To acquire the secret key that will be used for the transaction, both sides must embed their keys in an image, and form a jigsaw puzzle image. Then they have to post it on their websites and invite each other with an email in order to proceed. This proposal is close to ours since it involves a Turing test. Despite being firm, it may be time consuming in large-scale deployments.

Also close to our method is a patent that utilizes CAPTCHAs to shield e-banking against MITMAs [12]. According to this proposal, every transaction is watermarked with a server-generated CAPTCHA. This image contains a number that comes from a secure device that is already in the possession of the user. Our approach differs, because it uses a CAPTCHA as a way to select among various choice, and does not require the user to possess a secure two factor authentication device.

On the hardware front, a simple device called *"spies"*, that enables servers to establish client integrity is probably the oldest proposed solution [8]. There are also some systems that employ devices like smart phones, to make use of additional communication channels and as a result thwart MITMAs [34], [9]. For every transaction, a server-generated (TAN) is sent to the users mobile phone via SMS together with a transaction summary. If the summary is correct, the user sends the TAN back to the server in order to confirm the transaction.

Apart from the countermeasures, there is also an increased academic interest in MITM vulnerabilities in wireless networks [35], software updates [36], single sign-on protocols like Kerberos [37] and the *Universal Mobile Telecommunication Standard* (UMTS) [38].

B. Secure e-voting

The security aspects of building a sound e-voting system have been the subject of considerable research [39], [17], [40]. There are several systems that claim to provide security during electronic elections. The Zurich e-voting system is a flexible, modular e-voting system with a service-oriented architecture, that is used in Switzerland since 2004 [41]. Also using web services, and introducing the Election Markup Language (EML), the three-ballot-based secure electronic voting system, is similar to the aforementioned system, though it has not been deployed in a real large-scale experiment. The same applies to VOTEBOX, a direct recording electronic voting system (DRE) that assembles ideas and techniques from current research [42]. To evaluate systems like the ones mentioned above, the E-Voting System Security Optimization (EVSSO) method was recently proposed [3].

Apart from the completed systems, there are other schemes and protocols that protect the privacy of voter [43], [44], deal with authentication [45], [46], ensure their anonymity [47], [48], prevent double voting [49] and guarantee data integrity [50].

VI. CONCLUSION

The network-based nature of distance e-voting makes it prone to MITMAs, and especially the ones that take place locally on a user's inadequately secured PC. In that context is very easy for malicious software to hijack an e-voting session and replace a legitimate vote with a fraudulent one. In this paper, we have proposed an electronic voting scheme that secures the integrity of a user's vote from MITMAs. Our scheme is based on the idea that during electronic elections, a Turing test can be utilized to determine whether a vote is cast by a human or by malicious software. An advantage of our proposal is that it does not require specialized hardware or a physical secondary channel, thus making it suitable for large-scale elections. Our scheme as such does not provide user anonymity, because in the form we described it, its authentication is based on a TAN. Hence, the e-voting system that will benefit from our scheme must provide another layer that will deal with the separation of the users' authentication details from their vote and the provision of an appropriate auditing infrastructure. Finally, usability issues must be taken into account since CAPTCHAs are not the most efficient solutions especially for the elderly.

REFERENCES

[1] D. Evans and N. Paul, "Election security: Perception and reality," *IEEE Security and Privacy*, vol. 2, no. 1, pp. 24–31, 2004.

[2] M. Bishop and D. Wagner, "Risks of e-voting," *Commun. ACM*, vol. 50, no. 11, pp. 120–120, 2007.

[3] B. Ondrisek, "E-voting system security optimization," in *HICSS '09: Proceedings of the 42nd Hawaii International Conference on System Sciences*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–8.

[4] D. Frith, "E-voting security: hope or hype?" *Network Security*, vol. 11, pp. 14–16, 2007.

[5] J. Katz, "Efficient cryptographic protocols preventing "man-in-the-middle" attacks," Ph.D. dissertation, Columbia University, New York, 2002, publication number AAT3037726.

[6] R. Oppliger, R. Hauser, and D. Basin, "Ssl/tls session-aware user authentication," *Computer*, vol. 41, no. 3, pp. 59–65, 2008.

[7] D. Khusial and R. McKegney, "e-commerce security: Attacks and preventive strategies," IBM Toronto, Canada, Tech. Rep., April 2005.

[8] D. N. Serpano and R. J. Lipton, "Defense against man-in-the-middle attack in client-server systems," in *ISCC '01: Proceedings of the Sixth IEEE Symposium on Computers and Communications*. Washington, DC, USA: IEEE Computer Society, 2001, p. 9.

[9] T. Weigold, T. Kramp, R. Hermann, F. Höring, P. Buhler, and M. Baentsch, "The zurich trusted information channel — an efficient defence against man-in-the-middle and malicious software attacks," in *Trust '08: Proceedings of the 1st international conference on Trusted Computing and Trust in Information Technologies*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 75–91.

[10] C. Pope and K. Kaur, "Is it human or computer? defending e-commerce with captchas," *IT Professional*, vol. 7, no. 2, pp. 43–49, 2005.

[11] L. V. Ahn, M. Blum, and J. Langford, "CAPTCHA: Using hard ai problems for security," in *In Proceedings of Eurocrypt*. Springer-Verlag, 2003, pp. 294–311.

[12] D. J. Steeves and M. W. Snyder, "Secure online transactions using a captcha image as a watermark," US Patent 7200576, Apr. 2007. [Online]. Available: http://www.freepatentsonline.com/7200576.html

[13] P. Golle and N. Ducheneaut, "Preventing bots from playing online games," *Comput. Entertain.*, vol. 3, no. 3, pp. 3–3, 2005.

[14] S. Shirali-Shahreza and A. Movaghar, "A new anti-spam protocol using CAPTCHA," *Networking, Sensing and Control, 2007 IEEE International Conference on*, pp. 234–238, 15-17 April 2007.

[15] T. Schluessler, S. Goglin, and E. Johnson, "Is a bot at the controls?: Detecting input data attacks," in *NetGames '07: Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*. New York, NY, USA: ACM, 2007, pp. 1–6.

[16] A. Xenakis and A. Macintosh, "Procedural security analysis of electronic voting," in *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*. New York, NY, USA: ACM, 2004, pp. 541–546.

[17] E. Barr, M. Bishop, and M. Gondree, "Fixing federal e-voting standards," *Commun. ACM*, vol. 50, no. 3, pp. 19–24, 2007.

[18] T. W. Lauer, "The risk of e-voting," *Electronic Journal of e-Government*, vol. 2, pp. 177–186, 2004.

[19] H. Xia and J. C. Brustoloni, "Hardening web browsers against man-in-the-middle and eavesdropping attacks," in *WWW '05: Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005, pp. 489–498.

[20] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the HTTPS protocol," *IEEE Security and Privacy*, vol. 7, no. 1, pp. 78–81, 2009.

[21] B. Schneier, "Two-factor authentication: too little, too late," *Commun. ACM*, vol. 48, no. 4, p. 136, 2005.

[22] P. Golle, "Machine learning attacks against the asirra CAPTCHA," in *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2008, pp. 535–542.

[23] J. Yan and A. S. El Ahmad, "A low-cost attack on a microsoft captcha," in *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2008, pp. 543–554.

[24] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–134–I–141 vol.1, 18-20 June 2003.

[25] A. Hindle, M. W. Godfrey, and R. C. Holt, "Reverse engineering CAPTCHAs," *Reverse Engineering, Working Conference on*, vol. 0, pp. 59–68, 2008.

[26] D. C. Brabham, "Crowdsourcing as a model for problem solving: An introduction and cases," *Convergence*, vol. 14, no. 1, pp. 75–90, Feb. 2008. [Online]. Available: 10.1177/1354856507084420

[27] J. Barr and L. F. Cabrera, "AI gets a brain," *Queue*, vol. 4, no. 4, pp. 24–29, 2006.

[28] R. L. Rivest and A. Shamir, "How to expose an eavesdropper," *Commun. ACM*, vol. 27, no. 4, pp. 393–394, 1984.

[29] M. Bellovin, S.M.and Merritt, "An attack on the interlock protocol when used for authentication," *IEEE Transactions on Information Theory*, vol. 40, pp. 273–275, 1994.

[30] R. Security, "Enhancing one-time passwords for protection against real-time phishing attacks," Technology backgrounder, Tech. Rep., 2006.

[31] N. Asokan, V. Niemi, and K. Nyberg, "Man-in-the-middle in tunneled authentication protocols," In 11th Security Protocols Workshop, Tech. Rep., 2002.

[32] A. Alkassar and C. Stble, "Secure object identification - or: Solving the chess grandmaster problem," in *Proceedings of the 2003 Workshop on New Security Paradigms*. ACM Press, 2003, pp. 77–85.

[33] E.-J. Farn and C.-C. Chen, "A jigsaw puzzle based secret key exchange scheme," in *Proceedings of the 2008 International Conference on Machine Learning and Cybernetics*. IEEE, 2008, pp. 3067–3071.

[34] C. K. Bryan Parno and A. Perrig, "Phoolproof phishing prevention," in *Proc. Financial Cryptography and Data Security*. Springer Berlin / Heidelberg, 2006, pp. 1–19.

[35] H. Hwang, G. Jung, K. Sohn, and S. Park, "A study on MITM (man in the middle) vulnerability in wireless network using 802.1x and eap," in *ICISS '08: Proceedings of the 2008 International Conference on Information Science and Security*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 164–170.

[36] B. M. Luettmann and A. C. Bender, "Man-in-the-middle attacks on auto-updating software," *Bell Lab. Tech. J.*, vol. 12, no. 3, pp. 131–138, 2007.

[37] I. Cervesato, A. D. Jaggard, A. Scedrov, J.-K. Tsay, and C. Walstad, "Breaking and fixing public-key kerberos," *Inf. Comput.*, vol. 206, no. 2-4, pp. 402–424, 2008.

[38] U. Meyer and S. Wetzel, "A man-in-the-middle attack on UMTS," in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*. New York, NY, USA: ACM, 2004, pp. 90–97.

[39] C. Lambrinoudakis, E. Magkos, and V. Chrissikopoulos, "Electronic voting systems," in *(Chapter): J. Lopez, S. Furnell, A. Patel, S. Katsikas, (Ed.), "Securing Information and Communication Systems: Principles, Technologies and Applications"*. Artech House Publishers, Computer Security Series, 2008, pp. 307–323.

[40] M. Altman and G. M. Klass, "Current research in voting, elections, and technology," *Soc. Sci. Comput. Rev.*, vol. 23, no. 3, pp. 269–273, 2005.

[41] G. E. G. Beroggi, "Secure and easy internet voting," *Computer*, vol. 41, no. 2, pp. 52–56, 2008.

[42] D. Sandler, K. Derr, and D. S. Wallach, "Votebox: a tamper-evident, verifiable electronic voting system," in *SS'08: Proceedings of the 17th conference on Security symposium*. Berkeley, CA, USA: USENIX Association, 2008, pp. 349–364.

[43] Y. Mu and V. Varadharajan, "Anonymous secure e-voting over a network," in *ACSAC '98: Proceedings of the 14th Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society, 1998, p. 293.

[44] T. Rossler, H. Leitold, and R. Posch, "E-voting: A scalable approach using XML and hardware security modules," in *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 480–485.

[45] O. Cetinkaya and A. Doganaksoy, "A practical verifiable e-voting protocol for large scale elections over a network," in *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 432–442.

[46] F. Rodriguez-Henriquez, D. Ortiz-Arroyo, and C. Garcia-Zamora, "Yet another improvement over the mu-varadharajan e-voting protocol," *Comput. Stand. Interfaces*, vol. 29, no. 4, pp. 471–480, 2007.

[47] B. Kang, "Cryptanalysis on an e-voting scheme over computer network," in *CSSE '08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 826–829.

[48] T. Moran and M. Naor, "Split-ballot voting: everlasting privacy with distributed trust," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 246–255.

[49] W.-C. Ku and C.-M. Ho, "An e-voting scheme against bribe and coercion," in *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 113–116.

[50] n. Goirizelaia, I T. Selker, M. Huarte, and J. Unzilla, "An optical scan e-voting system based on n-version programming," *IEEE Security and Privacy*, vol. 6, no. 3, pp. 47–53, 2008.