AN ADAPTIVE INDEXED BINARY SEARCH TREE FOR EFFICIENT HOMOMORPHIC COERCION RESISTANT VOTING SCHEME

Vinodu George[1] and M P Sebastian[2]

[1]LBS College of Engineering, Kasaragod, Kerala, India
vinodu.george@gmail.com
[2]National Institute of Technology, Calicut, Kerala, India
sebasmp@nitc.ac.in

ABSTRACT

This paper presents a voting scheme that coalesces many of the advantageous features of an efficient e-voting scheme like receipt-freeness, uncoercibility and write-in ballot, without requiring untappable channels. Some of the previous schemes in the literature provide most of these features with a penalty of increased running time. The proposed scheme utilizes the advantages of a novel data structure known as " Indexed binary search tree (IBST)" for reducing the running time to linear order The self organizing nature of the data structure ensures an efficient voting process. It also satisfies the desirable features of the existing write-in and coercion resistant voting schemes, such as fair degree of efficiency and protection against any kind of adversarial behavior with lowest running time.

KEYWORDS

Coercion-Resistance, Electronic Voting, Homomorphic Encryption, Indexed Binary Search Tree, Mix Networks, Paillier Cryptosystem, Write-in Ballots.

1. INTRODUCTION

Electronic voting is a rising social application of cryptographic protocols. It promises the possibility of a convenient, efficient and secure facility for recording and tallying votes. Lot of literature on electronic voting has been developed over the last two decades. The uses of insecure Internet, incorrect implementations, and the resulting security breaches have caused substantial rework in this area. Several of these schemes were meant for secure electronic voting [21], [16].

One of the main requirements [23] of a voting protocol is the privacy of the voter. In a secret ballot, a vote must not identify a voter and any connection between the voter and his vote must be removed. Maximal privacy is achieved by a voting scheme, if the privacy of a voter is breached only with a collusion of all remaining entities (voters and authorities). Maximum privacy is accomplished using an anonymous channel – a communication channel where the voter (sender) is anonymous to the authority (receiver) and to any observer of the communication. In 10], a multistage system consisting of cryptography and shuffling/permutations, called mixnet, was proposed as an anonymous channel [16]. Receipt freeness is a stronger notion of privacy that restricts the voter to show how he has voted. A voter should not be provided with a receipt with which it may be able to prove vote to any other entity. One of the first receipt free protocol appeared in [8]. Since then, several schemes [8], [19] were proposed in order to meet the condition of receipt-freeness. Coercion resistant protocols, introduced by Juels, Catalano and Jakobsson [6],

give more freedom to the adversary and restrict any party to force another party to vote in a particular way. An adversary may attempt to coerce a voter and manipulate the manner in which a vote is cast. An adversary may also force a voter to abstain from casting a vote, or may even represent a valid voter at any stage of the voting scheme, by obtaining the voter's secret key. An incoercible voting scheme, will not allow such an adversary to coerce voters [16]. A coercion resistant and receipt free voting protocol prevents buying and selling of votes.

Cryptographic ballots do not support write-in [3] votes. Generally, when Alice wants to choose a name, other than from the predetermined list of candidates, she "writes-in" the pseudo-candidate option, and follows a separate process to specify her candidate. Kiayias and Yung [2] propose, a vector ballot approach, which provides write-in ballot.

This paper proposes a simple and practical voting scheme using the adaptive nature of the indexed binary search tree that is both receipt-free and coercion resistant. Also, it has the write-in property to cast the vote to a pseudo-candidate. Since it uses the Paillier crypto system [20], which is a homomorphic encryption scheme, the tallying of votes can also be done easily.

Rest of the paper is organized as follows. Section 2 discusses the previous work in the research area. Section 3 introduces the voting model. Section 4 introduces the proposed voting scheme and its detailed analysis. Section 5 concludes the paper.

2. PREVIOUS WORK

The early investigations of e-voting used a simple voting approach. Boardroom voting protocol is an example [12]. Failure of a single voter can cause an election failure in this case. Simple voting schemes like Voter Verified Paper Audit Trial (VVPAT) and vote by mail are inherently insecure and provide no guarantee for the security or privacy.

Receipt based voting schemes [10], [8], [9], [7] enable the voter to make sure that the vote cast by a person is accounted properly. By the introduction of the receipt, threats like buying and selling of votes become more serious. Receipts can be used as a solid proof for buyers and sellers.

Receipt free voting protocol was introduced by Benaloh [8]. A voter is incapable of producing any receipt as a proof for his/her choice of candidate. This makes a free environment for a voter to choose his/her candidate. This property of receipt freeness reduces vote buying and selling. The scheme in Sako [21] proposes a multi-authority scheme that uses re-encryption mixnets to mask the candidate choices and a homomorphic encryption scheme for the calculation of the final tally. This scheme was later refined by Michels and Horster [17]. Hirt and Sako [15] followed Sako and Kilian [21] present an efficient and correct voting scheme.

The voting schemes by David Chaum [11], Neff [4], [5] and Kiayias and Yung [1], [2] satisfy receipt freeness and allow write-in ballots. The protocol by David Chaum [11] has physical constraints of visual cryptography and voting stations. The scheme by Neff [5] is an efficient voting scheme based on shuffle mixnet protocol. This scheme also has a physical constraint known as "code book" which confirms the correspondence between the election codes and the voter's preferences. The protocol proposed by Kiayias and Yung [2] incorporates write-in choice into a homomorphic encryption scheme using "vector ballot" approach. This scheme lacks the coercion resistance property and hence is vulnerable to coercion.

The notion of coercion resistance [6] was proposed by Juels et al. This concept captures the fullest possible range of adversarial behavior in a real-world, Internet-based voting scheme. The voters access an anonymous channel during the voting stage. Anonymous channels can be realized practically by using of mixnets, while untappable channels require mainly unrealistic physical

assumptions. A drawback of this scheme is that the tallying overhead is quadratic with respect to the number of voters. The processing of V votes by N voters require $O(NV)$ steps for verification, which needs a minimum of N^2 steps. Thus, this scheme is practical only for small elections. Also, this scheme does not address the possibility of write-in ballots, allowing voters to choose their own ballots.

Acquisti [3] proposes a new voting scheme, which guarantees receipt-freeness, uncoercibility and write-in ballot without unrealistic physical assumptions and addresses some of the problems in [6]. This protocol allows flexible ballot formats to be used, including write-in ballots without the specific procedural constraints or physical assumptions needed as in [11] and [5]. Unlike in [11], [5], and [2], this protocol can neutralize "forced abstention and randomization attacks" [6]. This scheme also offers write-in ballot with coercion resistance. But for validation and tallying of votes, it takes all possible ballots for each credential. So the time complexity is more than [6]. Hence this scheme is practicable only for small elections.

2.1. Why coercion resistance?

Coercion resistance [6] is a stronger property for a voting scheme. Intuitively, an election protocol is coercion-resistant if a voter V cannot prove to a potential coercer C that he has voted in a particular way. (It is assumed that V cooperates with C interactively.) Receipt-freeness is a weaker property, for which we assume that V and C cannot interact during the protocol. To break the receipt-freeness, V later produces the evidence (the receipt) on how he has voted. Coercion resistance implies receipt-freeness, which implies privacy, a necessary property for any voting protocol.

Coercion-resistance [6] guarantees that a coercer cannot be convinced on how a voter has voted, even if the voter cooperates with him. Coercion resistance asserts that voter is incapable to prove it, so that the coercer has no reason to believe him. Intuitively, coercion resistance is a stronger property than privacy, since if it is possible for a coercer to detect the value of a voter's vote without the voter's cooperation, then it must also be possible with the voter's co-operation.

So we are in need of a voting scheme as in [9], which offers not only receipt-freeness, but also defends against randomization, forced-abstention, and simulation attacks [18], [15]. In addition, the stronger notions of privacy and coercion resistance will provide shields against the fullest possible range of adversarial behavior in a real-world Internet-based voting scheme.

3. OUR PROPOSAL

We propose a voting scheme that guarantees all the desirable properties as indicated above and by addressing the drawbacks of Kiayias and Yung, Juels et al. and Acquisti. The drawback with Kiayias and Yung scheme is lack of coercion resistance and with Juels et al. scheme is the additional time needed for the checking of the credential validity, thus making those schemes practicable only for small elections. The drawbacks are more serious with the scheme of Acquisti. Even though it accomplishes write-in ballot and coercion resistance, the time complexity for the same is $O(n^3)$, as it checks all possible ballots for each credential.

Our scheme is similar to Acquisti [3], but it performs the entire verification and elimination of duplications in linear order due to the adaptive nature of the storage mechanism for the vote cast. The elimination of duplicate votes and validation of the credential are combined to a single step. This reduces the running time. Since this scheme uses homomorphic encryption, the tallying time also gets reduced. Thus, our scheme will have all the features of Kiayias and Yung [2], Juels et al. [6] and Acquisti [3] with linear order complexity and is applicable for any type of election. This

protocol allows multiple casting, so that the coerced voter will get a chance to cast the intended vote. The reduction in running time is achieved by using a new adaptive data structure termed as indexed binary search tree.

3.1. Voting Model

An election system consists of several sets of entities

Authorities: Denoted by $A = \{A_1, A_2, \ldots\ldots A_{nA}\}$, the authorities are responsible for jointly issuing the keying material, i.e., credential to voters.

Validator: Denoted by D, responsible for the construction of the binary search tree and validation of the votes.

Talliers: The set of n_T Talliers, denoted by $T = \{T_1, T_2, \ldots\ldots T_{nT}\}$, are responsible for processing the ballots and jointly counting the votes and publishing a final tally.

Voters: The set of n_V voters, denoted by $V = \{V_1, V_2, \ldots\ldots V_{nV}\}$, are the entities participating in a given election.

3.2. Voting life cycle

A simplified life cycle of an electronic voting scheme may be divided into four main stages [21].

Setup. This stage involves the initialization of the technical part of the election system as well as the organizational structure.

Voting. In this stage the votes are cast. To do so, the voters need some form of authentication to validate the (digital) ballot completed by them. Furthermore, some (cryptographic) technique needs to be applied to ensure the ballot's secrecy.

Validation. This stage does the validation of votes cast by voters. If any of the votes are found to be invalid, they are discarded.

Tabulation. In this stage the Talliers compute the election result from the valid votes, after which the election officials announce it.

4. THE PROPOSED VOTING SCHEME

The Voting Authority consists of independently functioning servers that manage the registration, the adaptive indexed binary search tree and tallying of all the cast votes through the bulletin board. During the registration stage, each responsible authority creates shares of the voting credential, (includes indexing information for each eligible voter), which is used by the voter during the voting stage to authenticate his/her ballot. Each authority posts these credential shares, encrypted with the authority's public key and also with the voter's public key on a bulletin board at the place allotted for each voter. The authority also publishes the candidate slate on the bulletin board. The Validator multiplies the shares of the encrypted credential and creates the indexed binary search tree corresponding to the indexing information in the credential, and the resulting tree is also published on the bulletin board.

Each voter multiplies the decrypted shares he has selected from the bulletin board using the homomorphic property of the Paillier cryptosystem. The voter sends the encrypted credential along with the (encrypted) candidate choice and the proof to the bulletin board. At the end of the voting stage, the authority mixes all ciphertexts posted by eligible voters and stores the verified ballots in the binary search tree.

4.1. The Indexed Binary Search Tree (IBST)

The Indexed Binary Search Tree, which is an adaptive data structure used here, has two different parts, namely the Indexing part and the Binary Search Tree part.

4.1.1. The indexing part

This part of the tree holds the indexing information for each credentials in the voting system. Each node is a pointer, which points to the root of a fixed sized binary search tree. This part is an adaptively varying structure. Since the size of the binary search tree part is fixed, as the number of voters' increases, indexing part will adapt to that and it will add a pointer to a new binary search tree.

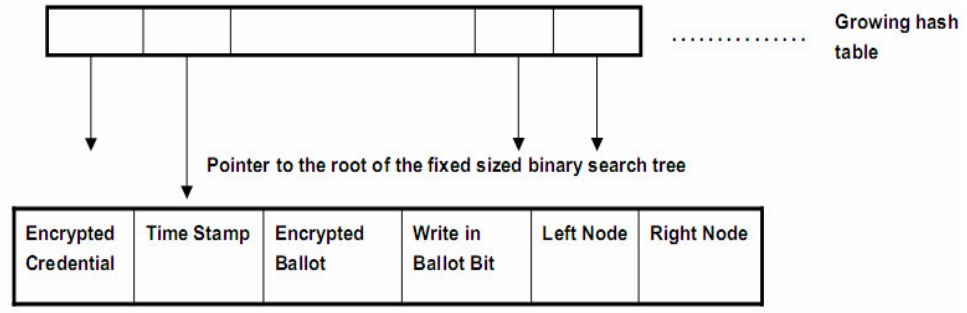

Figure 1. The Indexed Binary Search Tree structure

4.1.2. The Binary search tree part

This contains the randomly constructed binary search tree from the credentials of the voters with the same indexing value. Each nod contains the encrypted credential of the voter along with the other required information. During the validation process, the encrypted credential received along with the ballot, is compared with the credentials in the binary search tree for a match. In the matching process, to reduce the size of the binary search tree, we use the indexing information stored in the indexing part of the tree. Figure 1. illustrates the structure of the indexed binary search tree.

4.2. The Voting protocol

4.2.1. Set-up

The Paillier public/private key pairs (PA, SA, VA, VA_j) and (PT, ST, VT, VT_j) for the Authority and the Tallier, respectively, are generated in a reliable manner, and PA and PT are published along with the system parameters. The set of candidates is defined as Candidates = $\{1,B,B^2,\ldots.B^{nc-1}\}$ where B is an integer such that $B > n_v$ (number of voters) and n_c is the number of candidates.

4.2.2. Registration

The Authority A_j generates the string σ_{ij} for voter V_i that serves as the credential of the voter V_i. For each σ_{ij} the authority creates, A_j encrypts σ_{ij} using PA and the appropriate secret randomization, and then encrypts the resulting ciphertext with the public key of V_i, and publishes it on the bulletin board in a row publicly reserved for the shares of the credential of voter V_i. It is possible for A_j to furnish V_i with a proof that $(E^{PA}(\sigma_{ij}))$ is the ciphertext of σ_{ij}.

$S_i = E^{PVi}(E^{PA}(\sigma_{ij}))$

where E^{PVi} represents RSA encryption under V_i 's public key. The Authority A_j sends the encrypted credential for each voter V_i to a common trusted Validator in a random order so that the source of information about A_j remains unknown. So the Validator is with only the complete set of encrypted credential, and mapping between voter and credential is not possible for him. Since the Authority A_j sends the encrypted credential share in a random order, the Validator cannot find which Authority is responsible for each of the credentials. The Validator calculates the respective credential for each voter by multiplying the credential shares as follows;

$$\prod_{j=1,..m}(E^{PA}(\sigma_{ij})) = E^{PA}_{j=1,...m}(\sum \sigma_{ij}) \equiv E^{PA}(\sigma_i)$$

The Validator mixes the encrypted credentials in order to minimize the relation between the voters and credentials and then creates a indexed binary search tree with the encrypted credential as the key for the tree node.

4.2.3. Voting

A Voter receives the credential share from the space allotted for him on the bullet inboard and verifies the designated verifier proof of S_i. Upon successful verification, he multiplies together the shares $E^{PA}(\sigma_{ij})$. If the voter selects the candidate choice B^c from the bulletin board published by the Authority, then the write-in ballot bit is not set and he encrypts it with the public key of the Tallier, $E^{PT}(B^c)$. Otherwise, he sets the bit and casts his vote. Along with the encrypted credential, a voter will attach the non-interactive zero knowledge proof for the correctness of the credential, so that the validator can verify the authenticity of the credential. The voter V_i casts his vote as

$(E^{PK}(E^{PT}(B^c)), E^{PK}(E^{PA}(\sigma_i)))$.

In which the first part is the ciphertext of the candidate choice of the voter and the second part is the ciphertext of the credential of the voter. The voter wraps the resulting ciphertext with the RSA public key, *PK* of the Validator. The encryption using Paillier [20] public key *PT* of the Tallier should be semantically secure [14] for preventing any kind of passive attacks.

4.2.4. Tallying

To tally the ballots posted on the bulletin board, the Tallier performs the following steps:

Mixing of ballots: To minimize the relationship between the voter and his vote, the ballots are mixed.

Eliminating the duplicates and checking the credentials: The IBST is searched for the encrypted credential of the voter. A hit indicates a valid credential, otherwise, an invalid credential (which is discarded). Now check with the already stored vote for the valid credential, and if not, store the corresponding encrypted ballot and the time stamp of the vote. Otherwise, identify the last vote cast by comparing the time stamp of the ballot and store the corresponding vote. Before tallying, the time stamp of each ballot is removed in order to avoid uniquely identifiable ballots.

Tallying: The Tallier multiplies all ciphertexts of ballots in the tree nodes whose write-in ballot bit is not set and then decrypts the sum. The Tallier decrypts all encrypted ballots in the tree nodes whose bit is set since the write-in ballots need to be read individually and by combining the output of these two steps, the Tallier makes the final result. There is a threshold decryption by sharing the

secret key among the Talliers, i.e., using the shares of the secret key ST_j and the verification keys VT and VT_j. Each Tallier runs the decryption algorithm and produces a partial decryption of $E^{PT}(B^c)$, providing a proof of validity for the partial decryption. The combiner then produces the decryption of the ciphertext $E^{PT}(B^c)$, if enough partial decryptions (t or more) are valid.

4.3. Correctness of the Scheme

Our proposal follows the same procedure as defined in [6]. The advancement in our proposal is in the application of the data structure IBST for making the system more efficient. The correctness of the protocol is established in [6].

4.4. Complexity Analysis

The registration phase has a running time of O (No. of Voters ∗ No. of Authorities), as the Validator has to multiply the credential share of each Authority in order to get the credential of each voter. Since the number of Authority is constant, the running time of registration is O (No. of Voters) only. After issuing a credential, the Validator creates the IBST with a running time of O (No. of Voters ∗ log(No. of Voters in the fixed sized binary search tree)). Here again, as the number of voters in each binary search tree is constant, the total running time will be O(No. of Voters) only. See [13], [22] for a formal proof on this. The Validator takes the credential from the mix net, and hence the binary tree part of the indexed tree remains as a randomly built binary search tree. Traversing the binary tree part of the IBST does the validation and duplication elimination, and it needs a running time of O (No. of Voters ∗ log(No. of Voters in the binary tree)) only which is same as O(No. of Voters). Traversing through the tree also does tallying. Hence the total running time is O (No. of Voters ∗ a constant factor). Thus the entire process is completed in linear order. The write-in property does not cause any additional running time, as the tallying needs only a single traversal through the indexed binary tree. This makes the scheme applicable to any real time election.

5. CONCLUSION

The paper proposes an efficient electronic voting scheme that achieves privacy, uncoercibility, and receipt freeness with write-in property. It utilizes the advantages of the structure of a IBST for reducing the running time to linear order. Our scheme guarantees receipt freeness and uncoercibility with lowest running time compared to the other major coercion resistant receipt free voting schemes. The scheme can be used to cast different types of ballots that include yes/no, multi-candidate, 1 out of t choices, and write-in ballots in any type of election. In addition, our protocol allows for flexible ballot formats to be used in receipt-freeness and universal verifiability elections. One limitation is the possibility of regional analysis. Still the anonymity of the vote remains. So our scheme is expected to serve as a simple and secure means for any kind of real time electronic voting.

REFERENCES

[1] Aggelos Kiayias & Moti Yung, (2002) "Self-tallying elections and perfect ballot secrecy", *Proceedings of PKC '02*, volume 2274 of *LNCS*, Springer-Verlag, pp141–158.

[2] Aggelos Kiayias & Moti Yung, (2004) "The vector-ballot e-voting approach", *Proceedings of Financial cryptography*, volume 3110 of *LNCS,* Springer-Verlag, pp72–89.

[3] Alessandro Acquisti, (2004) "Receipt-Free Homomorphic Elections and Write-in Voter Verified Ballots", http://heinz.cmu.edu/~acquisti/papers/acquistielectronic_voting.pdf.

[4] C. A. Neff, (2001) "A verifiable secret shuffle and its application to e-voting", *Proceedings of SIGSAC: 8th ACM Conference on Computer and Communications Security,* ACM SIGSAC.

[5] Andrew Neff, (2003) "Detecting malicious poll site voting clients", http://www.votehere.net/.

[6] Ari Juels, Dario Catalano & Markus Jakobsson, (2005) "Coercion resistant electronic elections", *Proceedings of ACM Workshop On Privacy; The Electronic Society (WPES'05),* pp61–70.

[7] O. Baudron, P.A. Fouque, D. Pointcheval, G. Poupard, & J. Stern, (2001) "Practical Multi-Candidate Election System", *Proceedings of 20th ACM Symposium on Principles of Distributed Computing (PODC '01)*, pp274-283, ACM Press.

[8] J. Benaloh & D. Tuinstra, (1994) "Receipt-free secret-ballot elections", *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pp544-553.

[9] R. Cramer, R. Gennaro & B. Schoenmakers, (1997) "A secure and optimally efficient multi-authority election scheme", *Proceedings of EUROCRYPT'97*, volume 1233 of *LNCS*, Springer-Verlag, pp103-118.

[10] D. Chaum, (1981) "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM*, 24(2), pp84–88.

[11] David Chaum, (2002) "Secret-ballot receipts and transparent integrity", Draft, http://vreceipt.com/article.pdf.

[12] R. DeMillo, N. Lynch & M. C. Merritt, (1982) "Cryptographic Protocols", *Proceedings of 14th ACM Symp. On Theory of Computing, San Francisco*, pp383-400.

[13] Donald E. Knuth, (1973) *Sorting and Searching*, The Art of Computer Programming, volume 3,Addison-Wesley.

[14] S. Goldwasser & S. Micali, (1984) "Probabilistic Encryption", *Journal of Computer and System Sciences*, 28, pp270-299.

[15] M. Hirt & K. Sako, (2000) "Efficient receipt-free voting based on homomorphic encryption", *Proceedings of EUROCRYPT '00*, volume 1807 of *LNCS* , Springer-Verlag, pp539–556.

[16] R. Sampigethaya, & R. Poovendran, (2006) "A framework and taxonomy for comparison of electronic voting schemes", *Elsevier Computers & Security*, vol. 25, no. 2, pp137-153.

[17] M. Michels & P. Horster, (1996) "Some remarks on a receipt-free and universally verifiable mix-type voting scheme", *Proceedings of ASIACRYPT '96*, volume 1163 of *LNCS*, Springer-Verlag.

[18] T. Okamoto, (1996) "An electronic voting scheme", *Proceedings of IFIP World Conference on IT Tools*, Canberra, Australia, , pp21–30.

[19] T. Okamoto, (1997) "Receipt-free electronic voting schemes for large scale elections", *Proceedings of 5th Int. Security Protocols Workshop*, volume 1361 of *LNCS* , Springer-Verlag, pp25–35.

[20] Pascal Paillier, (1999) "Public-key cryptosystems based on composite degree residuosity classes", *Proceedings of EUROCRYPT '99*, volume 1592 of *LNCS* , Springer-Verlag, pp223–238.

[21] K. Sako & J. Kilian, (1995) "Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth", *Proceedings of EUROCRYPT '95*, volume 921 of *LNCS* , Springer-Verlag, pp393–403.

[22] T. H. Cormen, C. E. Leierson, R. L. Rivest & C. Stein, (2001) *Introduction to Algorithms,* MIT Press, Cambridge, Massachusetts - London.

[23] Warren D. Smith, (2005) "New cryptographic voting scheme with best known theoretical properties", *Proceedings of Workshop on Frontiers in Electronic Elections (FEE 2005)*, Milan, Italy.

Authors

Sebastian M P is a professor in Computer Science and Engineering Department of National Institute of Technology, Calicut, India. He received his B.Sc. Engg. degree from University of Kerala (1984). He received his M. Tech. degree in computer science and engineering (1990) and Ph. D. degree(1997) from the Department of Computer Science at Indian Institute of Science, Banglore. He received the 2009 Best Faculty Award for the outstanding technical contributions. His research interests are in mobile networks, cryptography and network security. He is on the editorial board of a number of international journals, and member of the program committees of several international conferences on computer networks and cryptography.

Vinodu George is an assistant professor in Computer Science and Engineering Department of LBS college of Engineering, Kasaragod, India. He received his B.Tech. degree from Cochin University of Science and Technology(1997) and his M.Tech. degree in computer science (2002) from the Computer Science and Engineering Department of National Institute of Technology, Calicut, India. His research interests are in cryptography, network security and algorithm complexity. His main interest right now is in electronic voting. He also has a record of industrial collaboration. He is a consultant for many industrial projects in the field of computer networks and security.