International Journal of Advances in Engineering & Scientific Research, Vol.4, Issue 3, May-2017,
pp 01-08 ISSN: 2349 –3607 (Online) , ISSN: 2349 –4824 (Print)

www.arseam.com
Impact Factor: 2.48

Cite this paper as : *Vikas K, Verma O.P , Deepak K & Sandeep J (2017). A NOVEL APPROACH FOR VERIFIABLE SECRET SHARING IN PROACTIVE NETWORK USING RSA, International Journal of Advances in Engineering & Scientific Research, Volume 4,(Issue 3, May-2017), pp 01–08. ISSN: 2349 –3607 (Online) , ISSN: 2349 –4824 (Print),*

A NOVEL APPROACH FOR VERIFIABLE SECRET SHARING IN PROACTIVE NETWORK USING RSA

Vikas Kumar
M.Tech. (CSE)
Delhi Technological University
Department of Computer Science, Delhi, India

Prof. O.P. Verma
Head of Department (CSE/IT), Department of Computer Science, Delhi Technological University, Delhi, India

Deepak Kumar
M.Tech (CSE), Scientist 'C', National Informatics Centre, Govt. of India

Sandeep Jain
M.Tech. (CSE), Scientist 'C', National Informatics Centre, Govt. of India

Abstract:

We consider perfect verifiable secret sharing (VSS) in a synchronous network of n processors (players) where a designated player called the dealer wishes to distribute a secret **s** among the players in a way that none of them obtain any information, but any **t + 1** players obtain full information about the secret. The round complexity of a VSS protocol is defined as the number of rounds performed in the sharing phase. Gennaro, Ishai, Kushilevitz and Rabin showed that three rounds are necessary and sufficient when **n > 3t**. Sufficiency, however, was only demonstrated by means of an inefficient (i.e., exponential-time) protocol and the construction of inefficient three-round protocol were left as an open problem. In this paper, we present an efficient three-round protocol for VSS. The solution is based on a three-round solution of so-called weak verifiable secret sharing (WSS), for which we also prove that three rounds are a lower bound. Furthermore, we also demonstrate that one round is sufficient for WSS when **n > 4t**, and that VSS can be achieved in 1 + " amortized rounds (for any " > 0) when **n > 3t**.

Keywords: Decryption, Encryption, Verifiable Secret Sharing, Hybrid Secure Communication, RSA.

I. INTRODUCTION

A networked environment is a key soul to the applications running in any enterprise. To maintain security in such applications involve more than a particular algorithm or protocol for encryption & decryption as well as for generation of sub-keys to be mapped to the plain text to generate cipher text. It means that participants should be in possession of some secret information (key), which can be used for protecting data from unauthorized users. The main purpose of this project is to provide an efficient way to the shareholders for secret reconstruction.
The RSA algorithm is the predominant mode used today for public-key cryptography and RSA threshold cryptographic systems have been devised by various authors. This paper explores several threshold RSA systems, which can be used for signing documents and decryption among other purposes. For simplicity, this paper focuses

on signature schemes. Threshold RSA cryptosystems are in use today, for example, the root CA key for MasterCard/VISA's Secure Electronic Transaction system is protected via a threshold RSA scheme[20].
Verifiable secret sharing (VSS) is an important primitive in distributed cryptography that allows a dealer to share a secret among **n** parties in the presence of an adversary controlling at most **t** of them. In the computational setting, the feasibility of VSS schemes based on commitments was established over two decades ago. Interestingly, all known computational VSS schemes rely on the homomorphic nature of the commitments or achieve weaker guarantees. As homomorphism is not inherent to commitments or to the computational setting in general, a closer look at its utility to VSS is called for.

In this paper, we demonstrate that homomorphism of commitments is not a necessity for computational VSS in the synchronous or in the asynchronous communication setting. We present new VSS schemes based only on the definitional properties of commitments that are almost as good as existing VSS schemes based homomorphic commitments. Furthermore, they have significantly lower communication complexities than their (statistical or perfect) unconditional counterparts. Considering the feasibility of commitments from any claw-free permutation, one-way function or collision-resistant hash function, our schemes can be an excellent alternative to unconditional VSS in the future.

Further, in the synchronous communication model, we observe that a crucial interactive complexity measure of round complexity has never been formally studied for computational VSS. Interestingly, for the optimal resiliency conditions, the least possible round complexity in the known computational VSS schemes is identical to that in the (statistical or perfect) unconditional setting: *three rounds*. Considering the strength of the computational setting, this equivalence is certainly surprising. In this paper, we show that three rounds are actually not mandatory for computational VSS. We present the rest two-round VSS scheme for $\mathbf{n = 2t + 1}$ and lower-bound the result tightly by proving the impossibility of one-round computational VSS for $\mathbf{t = 2}$ or $\mathbf{n = 3t}$. For the remaining condition of $\mathbf{t = 1}$ and $\mathbf{n = 4}$, we present a one-round VSS scheme. We also include a new two-round VSS scheme using homomorphic commitments that have the same communication complexity as the well-known three-round Feldman and Pedersen VSS schemes.

Objective:

- To review public-key cryptography
- To demonstrate that confidentiality and sender-authentication can be achieved simultaneously with PKC
- To review the RSA algorithm for public-key cryptography
- To present the proof of the RSA algorithm
- To go over the computational issues related to RSA
- To discuss the vulnerabilities of RSA

II. LITERATURE SURVEY

A method of encryption that combines two or more encryption schemes and includes a combination of symmetric and asymmetric encryption to take advantage of the strengths of each type of encryption is known as Hybrid Encryption. RSA is a cryptosystem for public-key encryption and is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet. RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total -- factoring -- is considered infeasible due to the time it would take even using today's supercomputers.

For our work in the synchronous setting, we closely follow the network and adversary model of the best known VSS schemes: Feldman VSS [11] and Pedersen VSS [4]. These schemes are called non-interactive as they require

unidirectional private links from the dealer to the parties; non-dealer parties speak only via the broadcast channel. Our protocol assumes nearly the same network model; however, in addition, we also allow parties to send messages to the dealer over the private channel. In practice, it is reasonable to assume that private links are bidirectional. Note that we do not need any private communication links between non-dealer parties. This network relaxation is an advantage of computational VSS over unconditional VSS as Pedersen [4] proved that unconditional VSS schemes are impossible in the network where only the dealer is connected to the parties by the private communication channel and a common broadcast medium is available.

It is also important to compare our work with unconditional VSS as we work towards reducing the cryptographic assumptions required for computational VSS. In unconditional or information theoretic settings, there are two different possibilities for the VSS properties; they can be held perfectly (i.e., error-free) or statistically with negligible error probability. Assuming a broadcast channel, perfect VSS is possible if and only if $\mathbf{n = 3t + 1}$ [2], while statistical VSS is possible for $\mathbf{n = 2t + 1}$ [5]. Gennaro et al. [13] initiated the study of the round complexity of unconditional VSS, which was extended by Fitzi et. al.[12] and Katz et. al.[9]. They concentrate on unconditional VSS with perfect security and show that three rounds in the sharing phase are necessary and sufficient for $\mathbf{n = 3t + 1}$. In the statistical scenario, Patra et al.[9] show that $\mathbf{n = 3t + 1}$ is necessary and sufficient for 2-round statistical VSS. Recently, Kumaresan et al.[6] extended the result to prove that 3 rounds are enough for designing statistical VSS with $\mathbf{n = 2t + 1}$.

III. PROBLEM STATEMENT

The major problem with existing cryptographic scenario is that it cannot achieve authentication and confidentiality along with integrity in a single step. In PKI, encryption and decryption are performed with different keys, where the private key is a non-sharable entity. Subsequently, there is a procedure to maintain authentication and confidentiality by implementation digital envelope for communication. A digital envelope is a secure electronic data container that is used to protect a message through encryption and data authentication. A digital envelope allows users to encrypt data with the speed of secret key encryption and the convenience and security of public key encryption. Rivest, Shamir and Adleman (RSA) Public-Key Cryptography Standard (PKCS) #7 governs the application of cryptography to data for digital envelopes and digital signatures. A digital envelope uses two layers of encryption: private key and public key encryption.
Public key encryption is used to send a secret key to a receiving party over a network. This technique does not require plain text communication.

Either of the following methods may be used to create a digital envelope:

- Secret key encryption algorithms, such as Rijndael or Twofish, for message encryption.
- PKC algorithm like RSA for secret key encryption with a receiver's public key.

A digital envelope may be decrypted by using a receiver's private key to decrypt a secret key, or by using a secret key to decrypt encrypted data. An example of a digital envelope is Pretty Good Privacy (PGP), popular data cryptography software that also provides cryptographic privacy and data communication authentication. A digital envelope is also known as a digital wrapper

IV. PROPOSED METHOD

The proposed solutions will not only give a way to establish secure communication but it will also help to improve the level of encryption by reducing security overhead. The system does not require any external system interface for

development.

Public-Key Requirements

•Public-Key algorithms rely on two keys where:

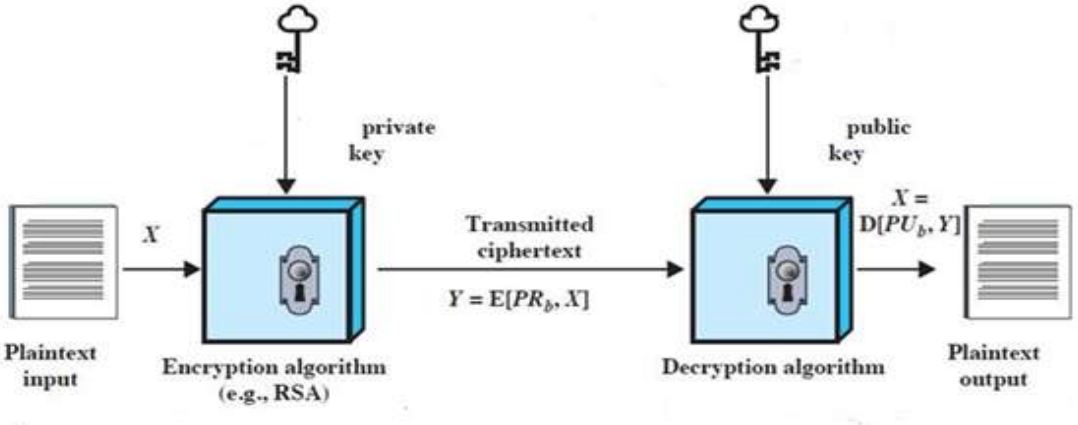
- It is computationally infeasible to find decryption key knowing only algorithm & encryption key
- It is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
- Either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

These are formidable requirements which only a few algorithms have satisfied

Public-Key Cryptography

PKC scheme uses one key for encryption and a different key for decryption. Modern PKC was first described using a two-key crypto system in which two parties could engage in a secure communication over a non-secure communications channel without having to share a secret key [5]. In PKC, one of the keys is designated the public key and may be advertised as widely as the owner wants. The other key is designated the private key and is never revealed to another party. RSA is one of the first and still most common PKC implementation that is in use today for key exchange or digital signatures.

The cardinal advantage of this method is that administration of keys on a network requires the presence of only a functionally trusted third party (TTP), as opposed to an unconditionally trusted TTP. Depending on the mode of usage, the TTP might only be required in an "off-line" manner, as opposed to in real time. Many public-key schemes yield relatively efficient signature mechanisms. The key used to describe the public verification function is typically much smaller than for the symmetric-key counterpart [6-9]



Security of Public Key Schemes

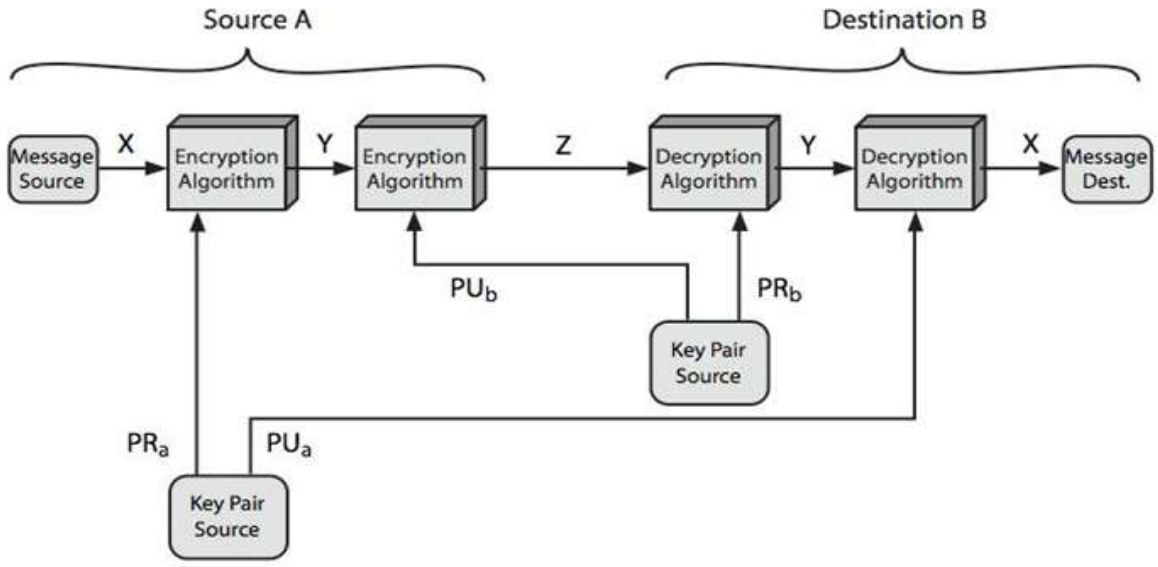
•like private key schemes brute force exhaustive search attack is always theoretically possible

•but keys used are too large … >512bits (PK schemes are generic and super-polynomial … can always choose a bigger instance, unlike block ciphers)

•security relies on a large enough difference in difficulty between easy (en/decrypt) and hard (cryptanalysis) problems

•more generally the hard problem is 'known', but is made hard enough to be impractical to break

Public-Key Cryptosystems



Chosen Ciphertext Attacks

•RSA is vulnerable to a chosen ciphertext attack (CCA)
•attacker chooses cipher-texts and gets decrypted plaintext back
•choose cipher-text to exploit properties of RSA to provide info to help cryptanalysis
•can counter with a random pad of plaintext
•or use optimal asymmetric encryption padding (OAEP)

V. VSS in the Synchronous Network Model

In our synchronous communication model, along with being adaptive and **t-bounded**, we allow the adversary to be rushing: in every round of communication it can wait to hear the messages of the honest parties before sending (or broadcasting) its own messages. By round complexity of VSS, we mean the number of rounds in the sharing and reconstruction phases of any execution. Although it is possible to have more than one round during the reconstruction phase [9], all of our protocols ask for a single round during reconstruction. Therefore, in this paper, we denote the round complexity of a VSS protocol as the number of rounds in its sharing phase.

1024 bit RSA Keys

The RSA algorithm is an asymmetric key cipher using a "public key" and a "private key". The parameters that describe the RSA public and private key pair are selected to satisfy the fundamental strict dependency requirements of the RSA cipher, enabling either key to decrypt content encrypted with the other key. The "public key" actually represents a pair of parameters (numbers): a **Modulus** and a **public exponent** E. The public exponent is usually chosen to be relatively small (often 3 bytes). The size of the Modulus in bits is referred to as the "key size". A Modulus of size 128 bytes represents a "1024 bit RSA key". The "private key" is usually described as a number pair consisting of the same key **Modulus** and a **private exponent** D. D is usually chosen to be about the same size as the modulus (~128 bytes). Random selection of Modulus E and D starts by random selection of two large prime numbers. The sample key data below shows typical prime numbers P, Q, the Modulus and Exponent displayed as

big-endian ordered byte arrays followed by the corresponding decimal integer number. The product P*Q below is calculated to verify that P*Q = Modulus.
Generation of RSA signatures or RSA encryption first typically requires formatting the data to be signed or encrypted into a PKCS #1 Signature (Type 1) or Encryption (Type 2) data block. The data block is the same size as the key modulus. Then exponentiation of the data with D (for RSA signatures) or E (for RSA encryption) (modulo Modulus) is performed producing the digital signature or RSA encrypted data.

P [64 bytes/512 bits]
[FA, F7, 2D, 97, 66, 5C, 47, 66, B9, BB, 3C, 33, 75, CC, 54, E0, 71, 12, 1F, 90, B4, AA, 94, 4C, B8, 8E, 4B, EE, 64, F9, D3, F8, 71, DF, B9, A7, 05, 55, DF, CE, 39, 19, 3D, 1B, EB, D5, FA, 63, 01, 52, 2E, 01, 7B, 05, 33, 5F, F5, 81, 6A, F9, C8, 65, C7, 65]

Decimal:
13144131834269512219260941993714669605006625743172006030529504645527800951523697
62014990305566325185422006702050378352478552367581915883654773477065606947 7
Digits: 155

Q [64 bytes/512 bits]
[EA, A0, F7, B0, 11, D8, 58, BC, 1F, E7, D9, EA, E6, 2B, E3, 68, 48, 39, 7A, 0C, 16, 5D, E3, 58, 95, DB, B7, CB, E8, F0, 24, B4, 65, 62, 5A, EB, 28, 08, 79, 0A, 30, 53, 18, C5, 36, 35, DC, 5C, F6, 66, 77, 44, F2, B4, BA, 46, CF, 30, 0A, DF, 05, AE, 40, 23]

Decimal:
12288506286091804108262645407658709962803358186316309871205769703371233115856772
65823682463109274040305712727192882036398381954429295019558590530369501597 1
Digits: 155

Modulus [128 bytes/1024 bits]
[E6, 03, BC, F9, FA, 9B, 40, 5C, D8, 51, AC, 0A, 3D, 33, F9, 12, 0C, 89, 57, E7, 98, 25, C2, A5, BD, AE, 35, 00, 0C, 5E, 6B, 1D, 30, 21, 62, 20, 0D, D3, 56, 59, C2, AE, 13, 8E, FF, 1E, 6B, B3, 94, A7, 45, F0, F8, 71, B8, AF, 86, 13, 71, 10, 6F, A0, DB, 08, 7C, 74, AC, 64, DF, 7C, 8B, 41, F3, 36, 3F, 7A, 79, 1D, 83, 3D, 68, 02, 90, 52, 3F, C7, 4D, 0B, 99, 26, 07, 44, 68, 1B, FE, 8C, C7, 0B, 67, 7D, 15, D1, 54, 6A, 34, F2, F4, D3, 61, A4, 3F, ED, 28, 55, 52, 39, 47, 14, 20, E4, 1A, 82, E7, 4D, 57, 69, 82, CF]

Decimal:
16152174667064029642647365822885998430666314431815268152405470907824573659036629
72483772980826569393306732864932303362619914669385966910731129686267107921489042
39628873374506302653492009810626437582587089465395941375496004739918498276676334
23824146549803003658606392990236819200423317203208018872696560061716 7
Digits: 309

Exponent = [01, 00, 01]

Decimal:
65537
Digits: 5

P*Q

161521746670640296426473658228859984306663144318152681524054709078245736590366297248377298082656939330673286493230336261991466938596691073112968626710792148904239628873374506302653492009810626437582587089465395941375496004739918498276676334238241465498030036586063929902368192004233172032080188726965600617167

VI. CONCLUSIONS AND FUTURE WORK

There were many more papers in the field of threshold RSA schemes than I could cover and it was hard to choose which to focus on, as a result some major work might have been omitted. If there is not a book yet on threshold RSA by a prominent expert in the field, then there is a need for one, as there are so many papers that claim to make one enhancement or another and it is very difficult to discern if the claim is truly novel or even correct, especially without a background in group theory. It is my hope that the schemes that I did examine are among the known works in the field. Embarking on this project deepened my understanding of the many difficulties encountered in threshold RSA schemes, yet many complexities I left uncovered, such as analysis from an adversarial or computational complexity perspective.

It seems that a merged approach of the schemes of sections 4 and 5 covers almost all the main TRSA security issues: no trusted dealer, protection of the private key, k out of N sharing, and measures for robustness against malicious participants during the key and modulus generation and during the signature generation. However, I am not convinced that it does this in a manner that is efficient enough to be commonly used unless N is very small, especially not in a system where the number of participants needed to sign a message could change arbitrarily. We found Boneh's (et.al.) work some of the most interesting in part because they have a project at Stanford (ITTC) that attempts to implement a non-trusted dealer TRSA scheme. The fastest scenario is when all servers are honest and k=N, as could be expected. Using various optimizations such as distributed sieving, they are able to generate 1024 bit keys in under 91 seconds on a 10Mbps Ethernet, and in less than 6 minutes on a wide area network with cross-country communication[18][19], which they consider acceptable, but will probably need enhancing before being adopted by the mainstream.

In this paper, we considered computational VSS as a standalone primitive. Our VSS schemes may also be easily leveraged in applications such as asynchronous Byzantine agreement protocols and multi-party communications. However, other VSS applications such as proactive share renewal and share recovery schemes[4] and distributed key generation[12] heavily rely on homomorphism of the commitments. It represents an interesting open problem if we can do better than in the unconditional case (e.g. [10]) for these applications. Further, most of the threshold cryptographic protocols also rely on homomorphism to verify the correctness. It will be interesting to check the feasibility of these threshold protocols based our VSS schemes without using expensive zero-knowledge proofs[21]. Finally, our protocols based on the definitional properties of commitment schemes are expensive (by a factor of n) in terms of communication complexity in comparison to the respective protocols employing homomorphic commitments. It is also worthwhile to study whether this gap in communication complexity is inevitable.

VII. ACKNOWLEDGMENT

We express our sincere thanks and a deep sense of gratitude to Prof. O.P. Verma, Head of Department, Department of Computer Science and Engineering, Delhi Technological University, for his valuable motivation and guidance, without which this study would not have been possible. We consider ourselves fortunate for having the opportunity to learn and work under his supervision and guidance over the entire period of association. We also thank our college for providing the motivating and encouraging environment so as to make our study successful.

REFERENCES

[1] Afolabi, A.O and E.R. Adagunodo, 2012. Implementation of an improved data encryption algorithm in a web-based learning system, International Journal of Research and Reviews in Computer Science. Vol. 3, No. 1.

[2] Bhoopendra, S.R., Prashanna, G. and S. Yadav, 2010. An Integrated encryption scheme used in Bluetooth communication mechanism. International Journal of Computer Tech. and Electronics Engineering (IJCTEE), vol. 1, issue 2.

[3] DI management (2005) "RSA algorithm", http://www.di-mgt.com.au/rsa_alg.html.

[4] Gaurav, S., 2012. Secure file transmission scheme based On hybrid encryption technique. International Journal of management, IT and Engineering. Vol. 2, issue 1.

[5] Hellman, M. and J. Diffie, 1976. New Directions in Cryptography. IEEE transactions on Information theory, vol. IT-22, pp: 644-654.

[6] Shinde, G.N. and H.S. Fade War, 2008. Faster RSA algorithm for decryption using Chinese remainder theorem. ICCES, Vol. 5, No. 4, pp. 255-261.

[7] Yang L. and S.H. Yang. 2007. A framework of security and safety checking for internet-based control systems. International Journal of Information and Computer security. Vol.1, No. 2.

[8] Washington, L.C. 2006. Introduction to Cryptography: with coding theory by Wade Trappe. Upper Saddle River, New Jersey, Pearson Prentice Hall.

[9] Wuling Ren College of Computer and Information Engineering Zhejiang Gongshang University. 2010. A hybrid encryption algorithm based on DES and RSA in Bluetooth communication. Second International Conference on Modeling, Simulation and Visualization methods.

[10] M. Ben-Or, R. Canetti, and O. Goldreich, "Asynchronous secure computation," in *Proc. 25th Annual ACM Symposium on Theory of Computing (STOC)*, 1993

[11] M. Ben-Or, R. Canetti, and O. Goldreich, "Asynchronous secure computation," in *Proc. 25th Annual ACM Symposium on Theory of Computing (STOC)*, 1993.

[12] G. Bracha, "An asynchronous [(n 1)=3]-resilient consensus protocol," in *Proc. 3rd ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 154–162, 1984.

[13] C. Cachin, K. Kursawe, F. Petzold, and V. Shoup, "Secure and efficient asynchronous broadcast protocols (extended abstract)," in *Advances in Cryptology: CRYPTO 2001* (J. Kilian, ed.), vol. 2139 of *Lecture Notes in Computer Science*, pp. 524–541, Springer, 2001.

[14] C. Cachin, K. Kursawe, and V. Shoup, "Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography," in *Proc. 19th ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 123–132, 2000.

[15] R. Canetti, *Studies in Secure Multiparty Computation and Applications*. Ph.D. thesis, Weizmann Institute, 1995.

[16] R. Canetti, R. Gennaro, A. Herzberg, and D. Naor, "Proactive security: Long-term protection against break-ins," *RSA Laboratories' CryptoBytes*, vol. 3, no. 1, 1997.

[17] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," in *Proc. 30th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 209–218, 1998.

[18] Malkin, M., Wu, T., Boneh, D., "Experimenting with Shared Generation of RSA keys" in proceedings of the *Internet Society's 1999 Symposium on Network and Distributed System Security (NDSS)*, pp. 43--56, 1999.

[19] M. Malkin, T. Wu, D. Boneh, "Building intrusion tolerant applications", in proceedings of the *8th USENIX Security Symposium*, pp. 79--91, 1999

[20] Y. Frankel, M. Yung, "Risk management using threshold RSA cryptosystems", (Usenix), http://www.usenix.org/publications/login/1998-5/frankel.html, 1998.

[21] K. Peng, F. Bao, "Efficient Publicly Verifiable Secret Sharing with Correctness, Soundness and ZK Privacy", LNCS 5932, Springer-Verlag, pp. 118–132, 2009.