Refik Molva
Gene Tsudik
Dirk Westhoff (Eds.)

LNCS 3813

Security and Privacy in Ad-hoc and Sensor Networks

Second European Workshop, ESAS 2005
Visegrad, Hungary, July 2005
Revised Selected Papers

Springer

Refik Molva Gene Tsudik
Dirk Westhoff (Eds.)

Security and Privacy in Ad-hoc and Sensor Networks

Second European Workshop, ESAS 2005
Visegrad, Hungary, July 13-14, 2005
Revised Selected Papers

Springer

Volume Editors

Refik Molva
Institut Eurécom
2229 Route des Crêtes
06560 Valbonne Sophia Antipolis, France
E-mail: molva@eurecom.fr

Gene Tsudik
University of California, Irvine
Computer Science Department
Irvine CA 92697-3425, USA
E-mail: gts@ics.uci.edu

Dirk Westhoff
NEC Europe Ltd., Network Laboratories
Kurfürsten-Anlage 36
69115 Heidelberg, Germany
E-mail: dirk.westhoff@netlab.nec.de

Library of Congress Control Number: 2005937512

CR Subject Classification (1998): E.3, C.2, F.2, H.4, D.4.6, K.6.5

ISSN 0302-9743
ISBN-10 3-540-30912-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-30912-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11601494 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3813

Preface

It was a pleasure to take part in the 2005 European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2005), held on July 13–14 in Visegrad (Hungary) in conjunction with the First International Conference on Wireless Internet (WICON) <http://www.wicon.org/>.

As Program Co-chairs, we are very happy with the outcome of this year's ESAS workshop. It clearly demonstrates the continued importance, popularity and timeliness of the workshop's topic: security and privacy in ad hoc and sensor networks. A total of 51 full papers were submitted. Each submission was reviewed by at least three expert referees. After a short period of intense discussions and deliberations, the Program Committee selected 17 papers for presentation and subsequent publication in the workshop proceedings. This corresponds to an acceptance rate of 33% — a respectable rate by any measure.

First and foremost, we thank the authors of ALL submitted papers. Your confidence in this venue is much appreciated. We hope that you will continue patronizing ESAS as authors and attendees. We are also very grateful to our colleagues in the research community who served on the ESAS Program Committee. Your selfless dedication is what makes the workshop a success.

Finally, we are very grateful to the ESAS Steering Group: Levente Buttyan, Claude Castelluccia, Dirk Westhoff and Susanne Wetzel. They had the vision and the drive to create this workshop in the first place; they also provided many insights and lots of help with this year's event. We especially acknowledge and appreciate the work of Levente Buttyan whose dedication (as Steering Committee member, PC member and Local Arrangements Chair) played a very important role in the success of the workshop.

September 2005

Refik Molva
Gene Tsudik

Organization

Program Chairs

Refik Molva, Eurecom, France
Gene Tsudik, UC Irvine, USA

Program Committee

Imad Aad, EPFL, Switzerland
N. Asokan, Nokia, Finland
Sonja Buchegger, UC Berkeley, USA
Laurent Bussard, Microsoft, Germany
Levente Buttyán, BUTE, CrySyS Lab, Hungary
Srdjan Capkun, UCLA, USA
Claude Castelluccia, INRIA, France
Hannes Hartenstein, University of Karlsruhe, Germany
Yih-Chun Hu, UC Berkeley, USA
Markus Jakobsson, Indiana University, Bloomington, USA
Yongdae Kim, University of Minnesota, Minneapolis, USA
Stefan Lucks, University of Mannheim, Germany
Breno de Medeiros, Florida State University, USA
Ludovic M, Supelec, France
Gabriel Montenegro, SunLabs, USA
Cristina Nita-Rotaru, Purdue University, USA
Guevara Noubir, Northeastern University, USA
Kaisa Nyberg, Nokia, Finland
Christof Paar, University of Bochum, Germany
Panagiotis Papadimitratos, Cornell University, USA
Andre Weimerskirch, University of Bochum, Germany
Dirk Westhoff, NEC Europe Network Lab., Germany
Susanne Wetzel, Stevens Institute of Technology, USA

Workshop Organizers

Levente Buttyán, Budapest University of Technology and Economics, Hungary (buttyan@crysys.hu)

Claude Castelluccia, INRIA, France (Claude.Castelluccia@inrialpes.fr)

Dirk Westhoff, NEC Europe Network Lab., Heidelberg, Germany (Dirk.Westhoff@netlab.nec.de)

Susanne Wetzel, Stevens Institute of Technology, USA (swetzel@cs.stevens.edu)

Lecture Notes in Computer Science

For information about Vols. 1–3742

please contact your bookseller or Springer

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2006.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3815: E.A. Fox, E.J. Neuhold, P. Premsmit, V. Wuwongse (Eds.), Digital Libraries: Implementing Strategies and Sharing Experiences. XVII, 529 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Sublibrary LNAI).

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3810: Y.G. Desmedt, H. Wang, Y. Mu, Y. Li (Eds.), Cryptology and Network Security. XI, 349 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Sublibrary LNAI).

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005. (Sublibrary LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

Vol. 3804: G. Bebis, R. Boyle, D. Koracin, B. Parvin (Eds.), Advances in Visual Computing. XX, 755 pages. 2005.

Vol. 3803: S. Jajodia, C. Mazumdar (Eds.), Information Systems Security. XI, 342 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005. (Sublibrary LNAI).

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005. (Sublibrary LNAI).

Vol. 3799: M. A. Rodríguez, I.F. Cruz, S. Levashkin, M.J. Egenhofer (Eds.), GeoSpatial Semantics. X, 259 pages. 2005.

Vol. 3798: A. Dearle, S. Eisenbach (Eds.), Component Deployment. X, 197 pages. 2005.

Vol. 3797: S. Maitra, C. E. V. Madhavan, R. Venkatesan (Eds.), Progress in Cryptology - INDOCRYPT 2005. XIV, 417 pages. 2005.

Vol. 3796: N.P. Smart (Ed.), Cryptography and Coding. XI, 461 pages. 2005.

Vol. 3795: H. Zhuge, G.C. Fox (Eds.), Grid and Cooperative Computing - GCC 2005. XXI, 1203 pages. 2005.

Vol. 3794: X. Jia, J. Wu, Y. He (Eds.), Mobile Ad-hoc and Sensor Networks. XX, 1136 pages. 2005.

Vol. 3793: T. Conte, N. Navarro, W.-m.W. Hwu, M. Valero, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XIII, 317 pages. 2005.

Vol. 3792: I. Richardson, P. Abrahamsson, R. Messnarz (Eds.), Software Process Improvement. VIII, 215 pages. 2005.

Vol. 3791: A. Adi, S. Stoutenburg, S. Tabet (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 225 pages. 2005.

Vol. 3790: G. Alonso (Ed.), Middleware 2005. XIII, 443 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005. (Sublibrary LNAI).

Vol. 3788: B. Roy (Ed.), Advances in Cryptology - ASIACRYPT 2005. XIV, 703 pages. 2005.

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3784: J. Tao, T. Tan, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction. XIX, 1008 pages. 2005.

Vol. 3783: S. Qing, W. Mao, J. Lopez, G. Wang (Eds.), Information and Communications Security. XIV, 492 pages. 2005.

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3779: H. Jin, D. Reed, W. Jiang (Eds.), Network and Parallel Computing. XV, 513 pages. 2005.

Vol. 3778: C. Atkinson, C. Bunse, H.-G. Gross, C. Peper (Eds.), Component-Based Software Development for Embedded Systems. VIII, 345 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3776: S.K. Pal, S. Bandyopadhyay, S. Biswas (Eds.), Pattern Recognition and Machine Intelligence. XXIV, 808 pages. 2005.

Vol. 3775: J. Schönwälder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3774: G. Bierman, C. Koch (Eds.), Database Programming Languages. X, 295 pages. 2005.

Vol. 3773: A. Sanfeliu, M.L. Cortés (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XX, 1094 pages. 2005.

Vol. 3772: M. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3771: J.M.T. Romijn, G.P. Smith, J. van de Pol (Eds.), Integrated Formal Methods. XI, 407 pages. 2005.

Vol. 3770: J. Akoka, S.W. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J. van den Heuvel, M. Kolp, J. Trujillo, C. Kop, H.C. Mayr (Eds.), Perspectives in Conceptual Modeling. XXII, 476 pages. 2005.

Vol. 3769: D.A. Bader, M. Parashar, V. Sridhar, V.K. Prasanna (Eds.), High Performance Computing – HiPC 2005. XXVIII, 550 pages. 2005.

Vol. 3768: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part II. XXVIII, 1088 pages. 2005.

Vol. 3767: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part I. XXVIII, 1022 pages. 2005.

Vol. 3766: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 231 pages. 2005.

Vol. 3765: Y. Liu, T. Jiang, C. Zhang (Eds.), Computer Vision for Biomedical Image Applications. X, 563 pages. 2005.

Vol. 3764: S. Tixeuil, T. Herman (Eds.), Self-Stabilizing Systems. VIII, 229 pages. 2005.

Vol. 3762: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. XXXI, 1228 pages. 2005.

Vol. 3761: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part II. XXVII, 653 pages. 2005.

Vol. 3760: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part I. XXVII, 921 pages. 2005.

Vol. 3759: G. Chen, Y. Pan, M. Guo, J. Lu (Eds.), Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. XIII, 669 pages. 2005.

Vol. 3758: Y. Pan, D.-x. Chen, M. Guo, J. Cao, J.J. Dongarra (Eds.), Parallel and Distributed Processing and Applications. XXIII, 1162 pages. 2005.

Vol. 3757: A. Rangarajan, B. Vemuri, A.L. Yuille (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. XII, 666 pages. 2005.

Vol. 3756: J. Cao, W. Nejdl, M. Xu (Eds.), Advanced Parallel Processing Technologies. XIV, 526 pages. 2005.

Vol. 3754: J. Dalmau Royo, G. Hasegawa (Eds.), Management of Multimedia Networks and Services. XII, 384 pages. 2005.

Vol. 3753: O.F. Olsen, L.M.J. Florack, A. Kuijper (Eds.), Deep Structure, Singularities, and Computer Vision. X, 259 pages. 2005.

Vol. 3752: N. Paragios, O. Faugeras, T. Chan, C. Schnörr (Eds.), Variational, Geometric, and Level Set Methods in Computer Vision. XI, 369 pages. 2005.

Vol. 3751: T. Magedanz, E.R.M. Madeira, P. Dini (Eds.), Operations and Management in IP-Based Networks. X, 213 pages. 2005.

Vol. 3750: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part II. XL, 1018 pages. 2005.

Vol. 3749: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part I. XXXIX, 942 pages. 2005.

Vol. 3748: A. Hartman, D. Kreische (Eds.), Model Driven Architecture – Foundations and Applications. IX, 349 pages. 2005.

Vol. 3747: C.A. Maziero, J.G. Silva, A.M.S. Andrade, F.M.d. Assis Silva (Eds.), Dependable Computing. XV, 267 pages. 2005.

Vol. 3746: P. Bozanis, E.N. Houstis (Eds.), Advances in Informatics. XIX, 879 pages. 2005.

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martín-Sánchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005. (Sublibrary LNBI).

Vol. 3744: T. Magedanz, A. Karmouch, S. Pierre, I.S. Venieris (Eds.), Mobility Aware Technologies and Applications. XIV, 418 pages. 2005.

¥396.48元

Table of Contents

Efficient Verifiable Ring Encryption for Ad Hoc Groups

Joseph K. Liu[1], Patrick P. Tsang[1], and Duncan S. Wong[2]

[1] Department of Information Engineering,
The Chinese University of Hong Kong Shatin, Hong Kong
{ksliu, pktsang3}@ie.cuhk.edu.hk
[2] Department of Computer Science,
City University of Hong Kong, Kowloon, Hong Kong
duncan@cityu.edu.hk

Abstract. We propose an efficient *Verifiable Ring Encryption* (VRE) for ad hoc groups. VRE is a kind of verifiable encryption [16,1,4,2,8] in which it can be publicly verified that there exists at least one user, out of a designated group of n users, who can decrypt the encrypted message, while the semantic security of the message and the anonymity of the actual decryptor can be maintained. This concept was first proposed in [10] in the name of *Custodian-Hiding Verifiable Encryption*. However, their construction requires the inefficient cut-and-choose methodology which is impractical when implemented. We are the first to propose an efficient VRE scheme that does not require the cut-and-choose methodology.

In addition, while [10] requires interaction with the encryptor when a verifier verifies a ciphertext, our scheme is non-interactive in the following sense: (1) an encryptor does not need to communicate with the users in order to generate a ciphertext together with its validity proof; and (2) anyone (who has the public keys of all users) can verify the ciphertext, without the help of the encryptor or any users. This non-interactiveness makes our scheme particularly suitable for ad hoc networks in which nodes come and go frequently as ciphertexts can be still generated and/or verified even if other parties are not online in the course. Our scheme is also proven secure in the random oracle model.

1 Introduction

A *Verifiable Encryption* [16,1,4,2,8] allows a prover to encrypt a message and sends to a receiver such that the ciphertext is publicly verifiable. That is, any verifier can ensure the ciphertext can be decrypted by the receiver yet knowing nothing about the plaintext. There are numerous applications of verifiable encryption. For example, in a publicly verifiable secret sharing scheme [16], a dealer shares a secret with several parties such that a third party can verify that the sharing was done correctly. This can be done by verifiably encrypting each shares under the public key of the corresponding party and proves to the third party that the ciphertext encrypt the correct shares. Another scenario is in a fair exchange environment [1], in which both parties want to exchange some

R. Molva, G. Tsudik, and D. Westhoff (Eds.): ESAS 2005, LNCS 3813, pp. 1–13, 2005.

information such that either each party obtain the other's data, or neither party does. One approach is to let both parties verifiably encrypt their data to each other under the public key of a trusted party and then to reveal their data. If one party refuses to do so, the other can go to the trusted party to obtain the required data. Verifiable encryption can be also applied in revokable anonymous credential [5]. When the administration organization issues a credential, it verifiably encrypts enough information under the public key of the anonymity revocation manager, so that later if the identity of the credential owner needs to be revealed, this information can be decrypted.

In an interactive *Custodian-Hiding Verifiable Encryption* (CHVE) [10], an *Encryptor* wants to send a public-key encrypted message to one among a group of *n users* through a *Verifier*. The Encryptor plays the role of a *Prover* and conducts an interactive protocol with the Verifier such that, if the Verifier is satisfied, at least one of the n possible decryptors can recover the message. At the same time, the message is semantically secure, even against the Verifier, and the identity of the actual decryptor is anonymous, again even to the Verifier. Custodian-Hiding Verifiable Encryption can be found useful in the applications of gateway system or receiver-oblivious transfer.

In ad hoc networks, nodes are highly dynamic and may switch from being online and being offline frequently from time to time. The verifiability of interactive Custodian-Hiding Verifiable Encryption schemes is virtually of no practical use if the encryptor goes offline (or leaves the networks forever) since no one can verify the validity of the ciphertext without the help of the encryptor. In the environment of ad hoc networks in which most users are highly mobile, it is unreasonable to require an encryptor to be always online and available to be contacted by a verifier. What we need is exactly a non-interactive approach to verify the ciphertext.

Let us spare a few words explaining the decision of naming our scheme as "Verifiable Ring Encryption" over "Custodian-Hiding Verifiable Encryption", as suggested by [10]. The word "Ring" is borrowed from Ring signatures [15] which is a signature scheme constructed in the structure of a ring in order to achieve 1-out-of-n anonymity of the signer. Analogously, Verifiable Ring Encryption implies an encryption scheme constructed in the structure of a ring, in which ciphertexts can be verified to be decryptable by some one, with the identity of that genuine decryptor hidden among a group of n members. Our choice of "Verifiable Ring Encryption" therefore better conveys the information on what the scheme actually does. Moreover, the non-interactiveness of our scheme suggests that a verifier is convinced by verifying the validity of some kind of proofs. These proofs can actually be thought of a kind of ring signatures in the sense that they convince verifiers of the fact that some 1 out of n users can decrypt a ciphertext, and yet hiding that decryptor's identity.

Finally we would like to note that the notion of "Verifiable Group Encryption" (VGE) has been used by [4] to mean something related but very different: VGE allows the prover to prove that any subset of t members of a group of n users can jointly recover the message behind a ciphertext, by making use of a secret sharing scheme. That is, the prover divides the message into n pieces of shares such that any t of them are enough to reconstruct m. Then he encrypts each share for each user using the user's encryption function, and sends all ciphertexts to the verifier. It is clear that the message m can be reconstructed if any t users decrypt their corresponding ciphertext to get the shares.

1.1 Contributions

We propose an efficient Verifiable Ring Encryption for ad hoc networks which is the first of its kind that is without the use of the inefficient cut-and-choose methodology. Furthermore, our proposed scheme is non-interactive. Unlike the previous one proposed in [10], in our scheme an encryptor does not need to communicate with the users in order to generate a ciphertext together with its validity proof. Also anyone who has got the public keys of all users can verify the ciphertext without the help of the encryptor or any users. Note that being non-interactive makes our scheme well-suited for ad hoc networks in which nodes are highly mobile. Ciphertexts can be still generated and/or verified even if other parties are not online in the course. We also prove the security of our proposed scheme in the random oracle model [3].

Organization: The rest of the paper is organized as follows. We give security definitions in Sec. 2. The details of our proposed scheme is presented in Sec. 3. Its security is analyzed in Sec. 4. We conclude the paper in Sec. 5.

2 Security Definition

2.1 Notations

Let a be a real number. We denote by $\lfloor a \rfloor$ the largest integer $b \leq a$, by $\lceil a \rceil$ the smallest integer $b \geq a$, and by $\lceil a \rfloor$ the largest integer $b \leq a + 1/2$. For positive real numbers a and b, let $[a]$ denote the set $\{0, 1 \ldots, \lfloor a \rfloor - 1\}$ and $[a, b]$ the set $\{\lfloor a \rfloor, \ldots, \lfloor b \rfloor\}$ and $[-a, b]$ denote the set $\{-\lfloor a \rfloor, \ldots, \lfloor b \rfloor\}$.

By $\text{neg}(\lambda)$ we denote a negligible function, i.e., a function f such that $f(\lambda) < 1/p(\lambda)$ holds for all polynomials $p(\lambda)$ and all sufficiently large λ.

We also use the shorthand notation $\{\mathsf{PK}\}_N$ and $\{\mathsf{SK}\}_N$, $N \in \mathbb{N}$, to mean the sets $\{\mathsf{PK}_1, \ldots, \mathsf{PK}_N\}$ and $\{\mathsf{SK}_1, \ldots, \mathsf{SK}_N\}$ respectively.

2.2 A High Level Description

Before giving a formal definition of verifiable ring encryption, we begin with a high level discussion of this notion in order to let readers understand more easily.

We start by the description of an ordinary verifiable encryption. A verifiable encryption scheme proves that a ciphertext encrypts a plaintext satisfying a certain relation $\mathcal{R}$. The relation $\mathcal{R}$ is defined by a generator algorithm $\mathcal{G}'$ which on input a security parameter λ outputs a binary relation $W \times \Delta$. For $\delta \in \Delta$, an element $w \in W$ such that $(w, \delta) \in \mathcal{R}$ is called a *witness* for δ. The encryptor will be given a value δ, a witness w for δ, then encrypts w to generate a ciphertext

ψ. Later, the encryptor may prove to another party that ψ decrypts to a witness for δ. In this system, the honest verifier will output accept or reject. If the system is sound, the verifier accepts a proof means that with overwhelming probability the ciphertext ψ can be decrypted to a witness for δ.

We extend this concept into a group of N designated receivers. In a verifiable ring encryption scheme, a prover proves that a ciphertext encrypts a plaintext satisfying one of the certain relation $\mathcal{R}$ which is corresponding to one of the receiver. The idea is that the encryptor will be given a value w, which is a witness for δ where $(w, \delta) \in \mathcal{R}$, and randomly generates other $N-1$ witnesses and the corresponding group elements.

Note that for an interactive proof system, both the prover and the verifier are required to interact in order to have the verifier convinced. If the proof system is non-interactive, the proof is carried out in a non-interactive fashion – the prover (or the encryptor) generates a proof transcript that can be used to convince a verifier at any later time that one (out of N) of the receivers can decrypt the corresponding witness of that group element δ. However, the verifier still cannot compute the identity of the actual decryptor.

2.3 Defining Verifiable Ring Encryption

A *Verifiable Ring Encryption* (VRE) scheme is actually a group encryption scheme with add-on *Verifiability*. A group encryption scheme is a generalization of a public key encryption scheme. Entities involved in such a scheme include an *encryptor* and a group of N *users*. The encryptor has a secret message m which he wants to send to a certain designated one out of the N users in the group, so that the secret message can be decrypted only by the designated member. In other words, a VRE scheme, apart from allowing a secret message to be encrypted to some designated members, provides with the encryptor the ability to prove that a ciphertext encrypts a plaintext satisfying certain relation $\mathcal{R}$.

The relation $\mathcal{R}$ is defined by a generator algorithm $\mathcal{G}'$ which on input 1^λ outputs a description $\Psi = \Psi[\mathcal{R}, W, \Delta]$ of a binary relation $\mathcal{R}$ on $W \times \Delta$. We require that the sets $\mathcal{R}$, W, and Δ are easy to recognize (given Ψ). For $\delta \in \Delta$, an element $w \in W$ such that $(w, \delta) \in \mathcal{R}$ is called a witness for δ. The idea is that the encryptor will be given a value δ, a witness w for δ, and a label L, and then encrypts w under L, yielding a ciphertext ψ. After this, the encryptor may prove to another party that ψ decrypts under L to a witness for δ. In carrying out the proof, the encryptor will need to make use of the random coins that were used by the encryption algorithm.

Now, a Ver-Gp-Enc scheme is a tuple of $(\mathcal{S}, \mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{P}, \mathcal{V})$ defined as follows:

- $\mathsf{param} \leftarrow \mathcal{S}(1^\lambda)$, the probabilistic polytime (PPT) *Setup* algorithm that on input security parameter $1^\lambda, \lambda \in \mathbb{N}$, outputs and publishes a set of system's parameters param that also includes the security parameter 1^λ, and a description $\Psi[\mathcal{R}, W, \Delta] \leftarrow \mathcal{G}'(1^\lambda)$.
- $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathcal{G}(\mathsf{param}, 1^{\lambda_i})$, the PPT *Key Generation* algorithm that on input the set of system's parameters param and security parameter $1^{\lambda_i}, \lambda_i \in \mathbb{N}$,

where $\lambda_i \geq \lambda$, outputs a public-key/private key pair $(\mathsf{PK}_i, \mathsf{SK}_i)$. PK_i includes also the security parameter 1^{λ_i}.

- $\psi \leftarrow \mathcal{E}(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L)$, the PPT *Encryption* algorithm that takes as input the set of system's parameters param, the group size $N \in \mathbb{N}$ of size polynomial in λ, a set of N public keys $\{\mathsf{PK}\}_N$, an index $\pi \in [1, N]$, a message $w \in W$ which is the witness of $\delta \in \Delta$, and a label $L \in \{0,1\}^*$, and outputs a ciphertext ψ. We denote by $\mathcal{E}'(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L)$ the pair $(\psi, coins)$, where ψ is the output of $\mathcal{E}(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L)$ and *coins* are the random coins used by $\mathcal{E}$ to compute ψ.
- $m/\perp \leftarrow \mathcal{D}(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, \mathsf{SK}_\pi, \psi, L)$, the polynomial-time *Decryption* algorithm that takes as input the set of system's parameters param, the group size $n \in \mathbb{N}$ of size polynomial in λ, a set $\{\mathsf{PK}\}_N$ of N public keys, an index $\pi \in [1, N]$, a private key SK_π, a ciphertext ψ, and a label $L \in \{0,1\}^*$, and outputs either a message $m \in \mathcal{M}$, or a special symbol $\perp$. The output of the algorithm implicitly defines the domain of m, that we denote by $\mathcal{M}$.
- $\mathsf{proof} \leftarrow \mathcal{P}(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L, \psi, coins)$, the PPT *Proof* algorithm that takes as input the tuple $(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L, \psi, coins)$ such that $(\psi, coins)$ is the output of some $\mathcal{E}'(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L)$, and outputs a proof proof.
- $0/1 \leftarrow \mathcal{V}(\mathsf{param}, N, \{\mathsf{PK}\}_N, L, \psi, \mathsf{proof})$, the polynomial-time *Verification* algorithm that takes as input the tuple $(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, L, \psi)$ such that ψ is the output of some $\mathcal{E}(\mathsf{param}, N, \{\mathsf{PK}\}_N, \pi, w, \delta, L)$ for some $\pi \in [1, N]$, $w \in \mathcal{M}$ and $\delta \in \Delta$, and outputs either 0 or 1, indicating accept or reject respectively.

Here we take a more relaxed approach in order to make it to be more convenient and adequate for practical applications. Instead of requiring the ciphertext to be decrypted to a witness, we only require that a witness can be easily reconstructed from the plaintext using some efficient reconstruction algorithm *recon*. We believe that this definition is more suitable for many applications.

Definition 1. *The above Ver-Gp-Enc scheme is a Verifiable Group Encryption scheme, if it is (1) correct, (2) sound, (3) zero-knowledge and (4) anonymous, as defined in the following.*

Correctness: *A Ver-Gp-Enc is correct if it satisfies both* Verification Correctness *and* Decryption Correctness *defined below.*

- (Verification Correctness.) *For all* $\mathit{param} \leftarrow \mathcal{S}(1^\lambda)$, *for all* $N \in \mathbb{N}$ *of size polynomial in* λ, *for all* $\lambda_i \geq \lambda$, $i \in [1, N]$, *for all* $(PK_i, SK_i) \leftarrow \mathcal{G}(\mathit{param}, 1^{\lambda_i})$, $i \in [1, N]$, *for all* $(w, \delta) \in \mathcal{R}$, *for all* $L \in \{0,1\}^*$, *for all* $\pi \in [1, N]$, *for all* $(\psi, coins) \leftarrow \mathcal{E}'(\mathit{param}, N, \{PK\}_N, \pi, w, \delta, L)$, *for all*

$$\mathit{proof} \leftarrow \mathcal{P}(\mathit{param}, N, \{PK\}_N, \pi, w, \delta, L, \psi, coins),$$

$$\Pr[x \leftarrow \mathcal{V}(\mathit{param}, N, \{PK\}_N, L, \psi, \mathit{proof}) : x = 1] = 1 - \mathrm{neg}(\lambda).$$

- (Decryption Correctness.) *For all* $\mathit{param} \leftarrow \mathcal{S}(1^\lambda)$*, for all* $N \in \mathbb{N}$ *of size polynomial in* λ*, for all* $\lambda_i \geq \lambda$*,* $i \in [1, N]$*, for all* $(\mathit{PK}_i, \mathit{SK}_i) \leftarrow \mathcal{G}(\mathit{param}, 1^{\lambda_i})$*,* $i \in [1, N]$*, for all* $\pi \in [1, N]$*, for all* $w \in \mathcal{M}$*, for all* $L \in \{0,1\}^*$*, for all*

$$\psi \leftarrow \mathcal{E}(\mathit{param}, N, \{\mathit{PK}\}_N, \pi, w, \delta, L),$$

$$\Pr[\tilde{m} \leftarrow \mathcal{D}(\mathit{param}, N, \{\mathit{PK}\}_N, \pi, \mathit{SK}_\pi, \psi, L) : m = \tilde{m}] = 1 - \mathrm{neg}(\lambda).$$

Soundness: *For all PPT adversaries* $\mathcal{A}_1, \mathcal{A}_2$*, and some reconstruction PPT algorithm recon,*

$$\begin{array}{l}
\Pr[\ \mathit{param} \leftarrow \mathcal{S}(1^\lambda); \\
\quad (N, \lambda_1, \ldots, \lambda_N) \leftarrow \mathcal{A}_1(\mathit{param}), \\
\quad \textit{where } N \textit{ has a size polynomial in } \lambda \textit{ and } \lambda_i \geq \lambda \textit{ for all } i \in [1, N]; \\
\quad (\mathit{PK}_i, \mathit{SK}_i) \leftarrow \mathcal{G}(\mathit{param}, 1^{\lambda_i}), \textit{ for all } i \in [1, N]; \\
\quad (\delta, \psi, L, \mathit{proof}) \leftarrow \mathcal{A}_2(\mathit{param}, N, \{\mathit{PK}\}_N, \{\mathit{SK}\}_N); \\
\quad x \leftarrow \mathcal{V}(\mathit{param}, N, \{\mathit{PK}\}_N, L, \psi, \mathit{proof}); \\
\quad m_j \leftarrow \mathcal{D}(\mathit{param}, N, \{\mathit{PK}\}_N, j, \mathit{SK}_j, \psi, L\}), \textit{ for all } j \in [1, N]; \\
\quad w_j \leftarrow \mathit{recon}(\mathit{param}, N, \{\mathit{PK}\}_N, \delta, m_j), \textit{ for all } j \in [1, N] : \\
\quad x = 1 \wedge (\forall j \in [1, N])((w_j, \delta) \notin \mathcal{R}) \qquad] \\
= \mathrm{neg}(\lambda).
\end{array}$$

Simply speaking, the definition of soundness above means that if a ciphertext is verified by a verifier to be valid, then there exists one user who can decrypt the ciphertext to the witness of δ, with overwhelming probability.

Zero knowledge: *There exists a PPT simulator Sim such that for all PPT adversaries* $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$*, we have*

$$\begin{array}{l}
\Pr[\ \mathit{param} \leftarrow \mathcal{S}(1^\lambda); \\
\quad (N, \lambda_1, \ldots, \lambda_N) \leftarrow \mathcal{A}_1(\mathit{param}), \\
\quad \textit{where } N \textit{ has a size polynomial in } \lambda \textit{ and } \lambda_i \geq \lambda \textit{ for all } i \in [1, N]; \\
\quad (\mathit{PK}_i, \mathit{SK}_i) \leftarrow \mathcal{G}(\mathit{param}, 1^{\lambda_i}), \textit{ for all } i \in [1, N]; \\
\quad (w, \delta, L, \pi) \leftarrow \mathcal{A}_2(\mathit{param}, N, \{\mathit{PK}\}_N, \{\mathit{SK}\}_N), \\
\quad \textit{where } (w, \delta) \in \mathcal{R},\ L \in \{0,1\}^* \textit{ and } \pi \in [1, N]; \\
\quad (\psi, \mathit{coins}) \leftarrow \mathcal{E}'(\mathit{param}, N, \{\mathit{PK}\}_N, \pi, w, \delta, L); \\
\quad b \leftarrow \{0, 1\}; \\
\quad \textit{if } b = 0 \\
\qquad \textit{then } \mathit{proof} \leftarrow \mathcal{P}(\mathit{param}, N, \{\mathit{PK}\}_N, \pi, w, \delta, L, \psi, \mathit{coins}) \\
\qquad \textit{else } \mathit{proof} \leftarrow \mathit{Sim}(\mathit{param}, N, \{PK\}_N, \delta, \psi, L); \\
\quad \hat{b} \leftarrow \mathcal{A}_3(\mathit{param}, N, \{\mathit{PK}\}_N, \{\mathit{SK}\}_N, w, \delta, L, \pi, \psi, \mathit{proof}) : \\
\quad b = \hat{b} \qquad] \\
= 1/2 + \mathrm{neg}(\lambda).
\end{array}$$

The definition above means that an adversary cannot distinguish a simulated proof from a proof generated from real execution of algorithms. In other words, the proof is zero-knowledge to a verifier.

Anonymity: *For all PPT adversaries* $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$,

$$\begin{array}{l}
\Pr[\ \mathit{param} \leftarrow \mathcal{S}(1^{\lambda});\\
\quad (N, \lambda_1, \ldots, \lambda_N) \leftarrow \mathcal{A}_1(\mathit{param}, \Psi),\\
\quad \textit{where } N \textit{ has a size polynomial in } \lambda \textit{ and } \lambda_i \geq \lambda \textit{ for all } i \in [1, N];\\
\quad (\mathit{PK}_i, \mathit{SK}_i) \leftarrow \mathcal{G}(\mathit{param}, 1^{\lambda_i}),\ \textit{for all } i \in [1, N];\\
\quad (w, \delta, L, \pi_0, \pi_1) \leftarrow \mathcal{A}_2(\mathit{param}, N, \{\mathit{PK}\}_N),\\
\quad \textit{where } (w, \delta) \in \mathcal{R} \textit{ and } \pi_0, \pi_1 \in [1, N] \textit{ are distinct;}\\
\quad b \leftarrow \{0, 1\};\\
\quad (\psi, \mathit{coins}) \leftarrow \mathcal{E}'(\mathit{param}, N, \{\mathit{PK}\}_N, \pi_b, w, \delta, L);\\
\quad \mathit{proof} \leftarrow \mathcal{P}(\mathit{param}, N, \{\mathit{PK}\}_N, \pi_b, w, L, \psi, \mathit{coins});\\
\quad \hat{b} \leftarrow \mathcal{A}_3(\mathit{param}, N, \{\mathit{PK}\}_N, w, \delta, L, \pi_0, \pi_1, \{\mathit{SK}_i | i \in [1, n] \backslash \{\pi_0, \pi_1\}\}, \psi, \mathit{proof});\\
\quad \hat{b} = b \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad]\\
= 1/2 + \mathrm{neg}(\lambda).
\end{array}$$

The definition of anonymity above means that an adversary cannot decide better than random guessing, given a ciphertext together with a corresponding proof transcript, who among the 2 possible designated members is actually designated, even he has corrupted all of the other $(N-2)$ members.

3 The Proposed Scheme

3.1 Key Generation

For each user, select two random ℓ-bit Sophie Germain primes p' and q', with $p' \neq q'$, and compute $p = (2p' + 1), q = (2q' + 1)$ and $n = pq$, where $\ell = \ell(\lambda)$ is a security parameter which is a polynomial in λ. Choose random $x_1, x_2, x_3 \in_R [n^2/4]$, choose a random $g' \in_R \mathbb{Z}^*_{n^2}$, and compute $g = (g')^{2n}$, $y_1 = g^{x_1}$, $y_2 = g^{x_2}$ and $y_3 = g^{x_3}$.

Let Γ be a cyclic group of order ρ generated by γ. We assume ρ and γ are publicly known, and that ρ is prime. Let $W = [\rho]$ and $\Delta = \Gamma$, and let $\mathcal{R} = \{(w, \delta) \in W \times \Delta : \gamma^w = \delta\}$.

Choose two other $\mathfrak{l}$-bit primes $\mathfrak{p}', \mathfrak{q}'$ and compute $\mathfrak{p} = 2\mathfrak{p}' + 1$, $\mathfrak{q} = 2\mathfrak{q}' + 1$ and $\mathfrak{n} = \mathfrak{p}\mathfrak{q}$, and choose $\mathfrak{g}, \mathfrak{h}$ as two generators of $\mathfrak{G}_{\mathfrak{n}'} \subset \mathbb{Z}^*_{\mathfrak{n}}$, where $\mathfrak{n}' = \mathfrak{p}'\mathfrak{q}'$ and $\mathfrak{G}_{\mathfrak{n}'}$ is the subgroup of $\mathbb{Z}^*_{\mathfrak{n}}$ of order $\mathfrak{n}'$, and $\mathfrak{l} = \mathfrak{l}(\lambda)$ which is a polynomial in λ.

The public key of this user is $(n, g, y_1, y_2, y_3, \mathfrak{n}, \mathfrak{g}, \mathfrak{h}, h, \rho, \gamma)$ and the secret key is (x_1, x_2, x_3, p, q) where $h = (1 + n \bmod n^2) \in \mathbb{Z}^*_{n^2}$. We further define $H : \{0,1\}^* \rightarrow \{0,1\}^k$ be a collision resistant hash function and $\mathsf{abs} : \mathbb{Z}^*_{n^2} \rightarrow \mathbb{Z}^*_{n^2}$ maps $(a \bmod n^2)$, where $0 < a < n^2$, to $(n^2 - a \bmod n^2)$ if $a > n^2/2$, and to $(a \bmod n^2)$, otherwise.

For a list of N users, we denote PK_i, the public key of user i be $(n_i, g_i, y_{1_i}, y_{2_i}, y_{3_i}, \mathfrak{n}_i, \mathfrak{g}_i, \mathfrak{h}_i, h_i, \rho_i, \gamma_i)$ and the corresponding secret key SK_i is $(x_{1_i}, x_{2_i}, x_{2_i}, p_i, q_i)$. For simplicity, we let L denote the list of the public keys of N users.

3.2 Encryption and Ciphertext Validity Proof

The prover sends an encrypted message to one of the N receivers such that only one of them can decrypt the message. At the same time, any verifier having the