

Transactions on Computational Science
and Computational Intelligence

Kevin Daimi · Hamid R. Arabnia
Leonidas Deligiannidis
Min-Shiang Hwang
Fernando G. Tinetti *Editors*

Advances in Security, Networks, and Internet of Things

Proceedings from SAM'20, ICWN'20,
ICOMP'20, and ESCS'20

Transactions on Computational Science and Computational Intelligence

Series Editor

Hamid Arabnia

Department of Computer Science

The University of Georgia

Athens, GA, USA

Computational Science (CS) and Computational Intelligence (CI) both share the same objective: finding solutions to difficult problems. However, the methods to the solutions are different. The main objective of this book series, “Transactions on Computational Science and Computational Intelligence”, is to facilitate increased opportunities for cross-fertilization across CS and CI. This book series publishes monographs, professional books, contributed volumes, and textbooks in Computational Science and Computational Intelligence. Book proposals are solicited for consideration in all topics in CS and CI including, but not limited to, Pattern recognition applications; Machine vision; Brain-machine interface; Embodied robotics; Biometrics; Computational biology; Bioinformatics; Image and signal processing; Information mining and forecasting; Sensor networks; Information processing; Internet and multimedia; DNA computing; Machine learning applications; Multi-agent systems applications; Telecommunications; Transportation systems; Intrusion detection and fault diagnosis; Game technologies; Material sciences; Space, weather, climate systems, and global changes; Computational ocean and earth sciences; Combustion system simulation; Computational chemistry and biochemistry; Computational physics; Medical applications; Transportation systems and simulations; Structural engineering; Computational electro-magnetic; Computer graphics and multimedia; Face recognition; Semiconductor technology, electronic circuits, and system design; Dynamic systems; Computational finance; Information mining and applications; Biometric modeling; Computational journalism; Geographical Information Systems (GIS) and remote sensing; Ubiquitous computing; Virtual reality; Agent-based modeling; Computational psychometrics; Affective computing; Computational economics; Computational statistics; and Emerging applications. For further information, please contact Mary James, Senior Editor, Springer, mary.james@springer.com.

More information about this series at <http://www.springer.com/series/11769>

Kevin Daimi • Hamid R. Arabnia
Leonidas Deligiannidis • Min-Shiang Hwang
Fernando G. Tinetti
Editors

Advances in Security, Networks, and Internet of Things

Proceedings from SAM'20, ICWN'20,
ICOMP'20, and ESCS'20

Editors

Kevin Daimi
Electrical and Computer Engineering, and
Computer Science
University of Detroit Mercy
Detroit, MI, USA

Hamid R. Arabnia
Department of Computer Science
University of Georgia
Athens, GA, USA

Leonidas Deligiannidis
School of Computing and Data Sciences
Wentworth Institute of Technology
Boston, MA, USA

Min-Shiang Hwang
Computer Science and Information
Engineering
Asian University
Taichung City, Taiwan

Fernando G. Tinetti
Facultad de Informática – CIC PBA
Universidad Nacional de La Plata
La Plata, Argentina

ISSN 2569-7072

ISSN 2569-7080 (electronic)

Transactions on Computational Science and Computational Intelligence

ISBN 978-3-030-71016-3

ISBN 978-3-030-71017-0 (eBook)

<https://doi.org/10.1007/978-3-030-71017-0>

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It gives us great pleasure to introduce this collection of papers that were presented at the following international conferences: Security and Management (SAM 2020); Wireless Networks (ICWN 2020); Internet Computing & IoT (ICOMP 2020); and Embedded Systems, Cyber-physical Systems, and Applications (ESCS 2020). These four conferences were held simultaneously (same location and dates) at Luxor Hotel (MGM Resorts International), Las Vegas, USA, July 27–30, 2020. This international event was held using a hybrid approach, that is, “in-person” and “virtual/online” presentations and discussions.

This book is composed of seven parts. Parts 1 through 4 (composed of 33 chapters) include articles that address various challenges with security and management (SAM). Part 5 (composed of 8 chapters) presents novel methods and applications in the areas of wireless networks (ICWN). Part 6 (composed of 8 chapters) discusses advancements in Internet computing and Internet of Things (ICOMP). Lastly, Part 7 (composed of 11 chapters) presents emerging trends in the areas of embedded systems and cyber-physical systems (ESCS).

An important mission of the World Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE (a federated congress to which this event is affiliated with), includes “*Providing a unique platform for a diverse community of constituents composed of scholars, researchers, developers, educators, and practitioners. The Congress makes concerted effort to reach out to participants affiliated with diverse entities (such as: universities, institutions, corporations, government agencies, and research centers/labs) from all over the world. The congress also attempts to connect participants from institutions that have **teaching** as their main mission with those who are affiliated with institutions that have **research** as their main mission. The congress uses a quota system to achieve its institution and geography diversity objectives.*” By any definition of diversity, this congress is among the most diverse scientific meetings in the USA. We are proud to report that this federated congress had authors and participants from 54 different

nations, representing variety of personal and scientific experiences that arise from differences in culture and values.

The program committees (refer to subsequent pages for the list of the members of committees) would like to thank all those who submitted papers for consideration. About 50% of the submissions were from outside the USA. Each submitted paper was peer reviewed by two experts in the field for originality, significance, clarity, impact, and soundness. In cases of contradictory recommendations, a member of the conference program committee was charged to make the final decision; often, this involved seeking help from additional referees. In addition, papers whose authors included a member of the conference program committee were evaluated using the double-blind review process. One exception to the above evaluation process was for papers that were submitted directly to chairs/organizers of pre-approved sessions/workshops; in these cases, the chairs/organizers were responsible for the evaluation of such submissions. The Congress (the joint conferences) received many good submissions. The overall acceptance rate for regular papers was 20%; 18% of the remaining papers were accepted as short and/or poster papers.

We are grateful to the many colleagues who offered their services in preparing this book. In particular, we would like to thank the members of the Program Committees of individual research tracks as well as the members of the Steering Committees of SAM 2020, ICWN 2020, ICOMP 2020, and ESCS 2020; their names appear in the subsequent pages. We would also like to extend our appreciation to over 500 referees.

As sponsors-at-large, partners, and/or organizers, each of the following (separated by semicolons) provided help for at least one research track: Computer Science Research, Education, and Applications (CSREA); US Chapter of World Academy of Science; American Council on Science and Education & Federated Research Council; and Colorado Engineering Inc. In addition, a number of university faculty members and their staff, several publishers of computer science and computer engineering books and journals, chapters and/or task forces of computer science associations/organizations from three regions, and developers of high-performance machines and systems provided significant help in organizing the event as well as providing some resources. We are grateful to them all.

We express our gratitude to all authors of the articles published in this book and the speakers who delivered their research results at the congress. We would also like to thank the following: UCMSS (Universal Conference Management Systems & Support, California, USA) for managing all aspects of the conference; Dr. Tim Field of APC for coordinating and managing the printing of the programs; the staff at Luxor Hotel (MGM Convention) for the professional service they provided; and Ashu M. G. Solo for his help in publicizing the congress. Last but not least, we would like to thank Ms. Mary James (Springer Senior Editor in New York) and Arun Pandian KJ (Springer Production Editor) for the excellent professional service they provided for this book project.

Book Co-editors and Chapter Co-editors: SAM 2020, ICWN 2020, ICOMP 2020, ESCS 2020

Detroit, MI, USA

Kevin Daimi

Athens, GA, USA

Hamid R. Arabnia

Boston, MA, USA

Leonidas Deligiannidis

La Plata, Argentina

Fernando G. Tinetti

Security and Management

SAM 2020 – Program Committee

- Dr. Jacques Bou Abdo, Computer Science Department, Notre Dame University – Louaize, Lebanon
- Dr. Hanaa Ahmed, Computer Science Department, University of Technology, Iraq
- Dr. Mohammed Akour, Department of Computer and Information Systems, Yarmouk University, Jordan
- Professor Emeritus Nizar Al Holou, Department of Electrical and Computer Engineering, University of Detroit Mercy, USA
- Professor Nadia Alsaidi, Department of Applied Mathematics & Computing, University of Technology, Iraq
- Allen Ashourian, ZRD Technology, USA
- Professor Emeritus Hamid Arabnia, (Vice Chair and Coordinator, SAM'20), Department of Computer Science, University of Georgia, USA
- Dr. David Arroyo, Researcher, Spanish National Research Council (CSIC), Spain
- Dr. Shadi Banitaan, (Sessions/Workshops Co-Chair, SAM'20), Computer Science and Software Engineering, University of Detroit Mercy, USA
- Dr. Clive Blackwell, Innovation Works, Airbus Group, United Kingdom
- Dr. Violeta Bulbenkiene, Department of Informatics Engineering, Klaipeda University, Lithuania
- Dr. María Calle, Department of Electrical and Electronics Engineering, Universidad del Norte, Barranquilla, Colombia
- Eralda Caushaj, School of Business Administration, Oakland University, USA
- Dr. Feng Cheng, Internet Technologies and Systems, Hasso-Plattner-Institute, University of Potsdam, Germany
- Professor Emeritus Kevin Daimi, (Conference Chair, SAM'20), Computer Science and Software Engineering, University of Detroit Mercy, USA
- Dr. Ioanna Dionysiou, Department of Computer Science, University of Nicosia, Cyprus

- Dr. Hiroshi Dozono, Faculty of Science and Engineering, Saga University, Japan
- Dr. Luis Hernandez Encinas, (Program Co-Chair, SAM'20), Department of Information Technologies and Communications, Institute of Physical and Information Technologies (ITEFI-CSIC), Spain
- Professor Levent Ertaul, Department of Computer Science, California State University East Bay, USA
- Dr. Ken Ferens, Department of Electrical and Computer Engineering, University of Manitoba, Canada
- Professor Guillermo Francia, Center for Cybersecurity, University of West Florida, USA
- Steffen Fries, Siemens AG, Corporate Technology, CT RDA ITS, Germany
- Dr. Víctor Gayoso Martínez, Spanish National Research Council (CSIC), Spain
- Dr. Bela Genge, University of Medicine, Pharmacy, Science and Technology of Tg. Mures, Romania
- Professor Danilo Gligoroski, Norwegian University of Science and Technology (NTNU), Norway
- Dr. Michael R. Grimaila, Department of Systems Engineering and Management, Center for Cyberspace Research, Air Force Institute of Technology, USA
- Dr. Diala Abi Haidar, Management Information Systems Department, Dar Al Hekma University, Saudi Arabia
- Dr. Hicham H. Hallal, College of Engineering, American University of Sharjah, UAE
- Dr. Hanady Hussein Issa, Arab Academy for Science, Technology and Maritime Transport (ASTMT), Egypt
- Christian Jung, Security Engineering Department, Fraunhofer IESE, Germany
- Nesrine Kanieche, University of Sheffield, United Kingdom
- Dr. Marie Khair, Computer Science Department, Notre Dame University – Louaize, Lebanon
- Professor Hiroaki Kikuchi, (Program Co-Chair, SAM'20), Department of Frontier Media Science, School of Interdisciplinary Mathematical Sciences, Meiji University, Japan
- Professor Irene Kopalani, Princeton University Research Computing, USA
- Dr. Arash Habibi Lashkari, Canadian Institute for Cybersecurity (CIC), University New Brunswick (UNB), Canada
- Dr. Flaminia Luccio, (Sessions/Workshops Co-Chair, SAM'20), Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Italy
- Dr. Giovanni L. Masala, Computing, Mathematics & Digital Technology, Manchester Metropolitan University, UK
- Dr. Wojciech Mazurczyk, Faculty of Electronics and Information Technology, Warsaw University of Technology, Poland
- Dr. Suzanne Mello-Stark, Rhode Island College, USA
- Dr. Sherry Michael, Enterprise Partners, Bahrain
- Dr. Alexandra Michota, Open University of Cyprus, Cyprus

- Dr. Esmiralda Moradian, (Posters Co-Chair, SAM'20), Department of Computer and Systems Sciences, Stockholm University, Sweden
- Dr. Nader M Nassar, Innovation for Security and Compliance Group, IBM Corp, USA
- Dr. Ana Nieto, Computer Science Department, University of Malaga, Spain
- Dr. Mais Nijim, Department of Electrical Engineering and Computer Science, Texas A&M University-Kingsville, USA
- Dr. Eugenia Nikolouzou, Internet Application Department, Hellenic Authority for Communication Security and Privacy, Greece
- Dr. Saibal K. Pal, DRDO & University of Delhi, India
- Dr. Cathryn Peoples, (Posters Co-Chair, SAM'20), School of Computing and Communications, Faculty of Science, Technology, Engineering & Mathematics, The Open University, United Kingdom
- Dr. Junfeng Qu, Department of Information Technology, Clayton State University, USA
- Dr. Peter Schartner, System Security Research Group, Alpen-Adria-Universität Klagenfurt, Austria
- Dr. Karpoor Shashidhar, Computer Science Department, Sam Houston State University, USA
- Dr. Nicolas Sklavos, Computer Engineering & Informatics Department, University of Patras, Greece
- Ashu M.G. Solo, (Publicity Chair, SAM'20) Maverick Technologies America, USA.
- Dr. Cristina Soviany, (Program Co-Chair, SAM'20), Features Analytics SA, Belgium
- Professor Hung-Min Sun, Information Security, Department of Computer Science, National Tsing Hua University, Taiwan
- Professor Woei-Jiunn Tsaor, Department of Computer Science, National Taipei University, Taiwan
- Professor Shengli Yuan, Department of Computer Science and Engineering Technology, University of Houston-Downtown, USA

Wireless Networks

ICWN 2020 – Program Committee

- Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); ECE Department; Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA
- Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).
- Dr. Travis Atkison; Director, Digital Forensics and Control Systems Security Lab, Department of Computer Science, College of Engineering, The University of Alabama, Tuscaloosa, Alabama, USA
- Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de Valencia (UPV), Department of Computer Engineering (DISCA), Valencia, Spain
- Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA
- Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA
- Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA
- Prof. Tai-hoon Kim; School of Information and Computing Science, University of Tasmania, Australia
- Prof. Louie Lolong Lacatan; Chairperson, Computer Engineering Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of CS & IT (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria

- Prof. Dr. Guoming Lai; Computer Science and Technology, Sun Yat-Sen University, Guangzhou, P. R. China
- Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations
- Prof. Salahuddin Mohammad Masum; Computer Engineering Technology, Southwest Tennessee Community College, Memphis, Tennessee, USA
- Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria
- Prof. James J. (Jong Hyuk) Park (Congress Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IIJTCC; Head of DCSE, SeoulTech, Korea
- Dr. Akash Singh (Congress Steering Committee); IBM Corporation, Sacramento, California, USA; Chartered Scientist, Science Council, UK; Fellow, British Computer Society; Member, Senior IEEE, AACR, AAAS, and AAAI; IBM Corporation, USA
- Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.
- Prof. Fernando G. Tinetti (Congress Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina
- Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine
- Prof. Shiuh-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.
- Dr. Yunlong Wang; Advanced Analytics at QuintilesIMS, Pennsylvania, USA
- Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA
- Prof. Hyun Yoe; Director of Agrofood IT Research Center and Vice President of Korea Association of ICT Convergence in the Agriculture and Food Business (KAICAF); Director of Agriculture IT Convergence Support Center (AITCSC); Department of Information and Communication Engineering, Sunchon National University, Suncheon, Republic of Korea (South Korea)
- Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Internet Computing & IoT

ICOMP 2020 – Program Committee

- Prof. Afrand Agah; Department of Computer Science, West Chester University of Pennsylvania, West Chester, PA, USA
- Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); Professor and Chair, Electrical and Computer Engineering Department; Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA
- Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).
- Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de Valencia (UPV), Department of Computer Engineering (DISCA), Valencia, Spain
- Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA
- Prof. Zhangisina Gulnur Davletzhanovna; Vice-rector of the Science, Central-Asian University, Kazakhstan, Almaty, Republic of Kazakhstan; Vice President of International Academy of Informatization, Kazakhstan, Almaty, Republic of Kazakhstan
- Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA
- Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA
- Prof. Houcine Hassan; Department of Computer Engineering (Systems Data Processing and Computers), Universitat Politècnica de Valencia, Spain

- Prof. Tai-hoon Kim; School of Information and Computing Science, University of Tasmania, Australia
- Prof. Louie Lolong Lacatan; Chairperson, Computer Engineering Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of CS & IT (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria
- Prof. Dr. Guoming Lai; Computer Science and Technology, Sun Yat-Sen University, Guangzhou, P. R. China
- Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations
- Dr. Ali Mostafaiepour; Industrial Engineering Department, Yazd University, Yazd, Iran
- Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria
- Prof. James J. (Jong Hyuk) Park (Congress Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea
- Dr. Xuwei Qi; Research Faculty & PI, Center for Environmental Research and Technology, University of California, Riverside, California, USA
- Dr. Akash Singh (Congress Steering Committee); IBM Corporation, Sacramento, California, USA; Chartered Scientist, Science Council, UK; Fellow, British Computer Society; Member, Senior IEEE, AACR, AAAS, and AAI; IBM Corporation, USA
- Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.
- Prof. Fernando G. Tinetti (Congress Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina
- Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine
- Varun Vohra; Certified Information Security Manager (CISM); Certified Information Systems Auditor (CISA); Associate Director (IT Audit), Merck, New Jersey, USA
- Prof. Shiuh-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.
- Dr. Yunlong Wang; Advanced Analytics at QuintilesIMS, Pennsylvania, USA

- Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA
- Prof. Hyun Yoe; Director of Agrofood IT Research Center and Vice President of Korea Association of ICT Convergence in the Agriculture and Food Business (KAICAF); Director of Agriculture IT Convergence Support Center (AITCSC); Department of Information and Communication Engineering, Sunchon National University, Suncheon, Republic of Korea (South Korea)
- Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
- Dr. Farhana H. Zulkernine; Coordinator of the Cognitive Science Program, School of Computing, Queen's University, Kingston, ON, Canada

Embedded Systems, Cyber-physical Systems, & Applications

ESCS 2020 – Program Committee

- Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); Professor and Chair, ECE Department; Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA
- Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).
- Dr. P. Balasubramanian; School of CSE, Nanyang Technological University, Singapore
- Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de Valencia (UPV), Department of Computer Engineering (DISCA), Valencia, Spain
- Prof. Emeritus Kevin Daimi (Congress Steering Committee); Director, Computer Science and Software Engineering Programs, Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA
- Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA
- Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA
- Prof. Houcine Hassan; Department of Computer Engineering (Systems Data Processing and Computers), Universitat Politècnica de Valencia, Spain
- Prof. Dr. Guoming Lai; Computer Science and Technology, Sun Yat-Sen University, Guangzhou, P. R. China
- Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of

BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations

- Dr. Ali Mostafaeipour; Industrial Engineering Department, Yazd University, Yazd, Iran
- Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Tech., Ambrose Alli University, Edo State, Nigeria
- Dr. Benaoumeur Senouci; Embedded Systems Department, LACSC Laboratory-Central Electronic Engineering School, ECE, Paris, France
- Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.
- Prof. Fernando G. Tinetti (Congress Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina
- Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine
- Prof. Shih-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.
- Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA
- Prof. Hyun Yoe; Director of Agrofood IT Research Center and Vice President of Korea Association of ICT Convergence in the Agriculture and Food Business (KAICAF); Director of Agriculture IT Convergence Support Center (AITCSC); Department of Information and Communication Engineering, Sunchon National University, Suncheon, Republic of Korea (South Korea)
- Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Contents

Part I Authentication, Biometrics, and Cryptographic Technologies	
Statistical Analysis of Prime Number Generators Putting Encryption at Risk	3
Aykan Inan	
Secure Authentication Protocol for Drones in LTE Networks	17
Dayoung Kang, Gyuhong Lee, and Jin-Young Choi	
Memorable Password Generation with AES in ECB Mode	33
Timothy Hoang and Pablo Rivas	
A Comprehensive Survey on Fingerprint Liveness Detection Algorithms by Database and Scanner Model	39
Riley Kiefer and Ashokkumar Patel	
Suitability of Voice Recognition Within the IoT Environment	53
Salahaldeen Duraibi, Fahad Alqahtani, Frederick Sheldon, and Wasim Alhamdani	
Chor-Rivest Knapsack Cryptosystem in a Post-quantum World	67
Raúl Durán Díaz, Luis Hernández-Álvarez, Luis Hernández Encinas, and Araceli Queiruga-Dios	
An Effective Tool for Assessing the Composite Vulnerability of Multifactor Authentication Technologies	85
Adam English and Yanzhen Qu	
Part II Computer and Network Security and Related Issues	
Phishing Prevention Using Defense in Depth	101
Joel Williams, Job King, Byron Smith, Seyedamin Pouriyeh, Hossain Shahriar, and Lei Li	

Phishing Detection using Deep Learning 117
Beatrice M. Cerda, Shengli Yuan, and Lei Chen

Enhancing Data Security in the User Layer of Mobile Cloud Computing Environment: A Novel Approach 129
Noah Oghenfego Ogwara, Krassie Petrova, Mee Loong (Bobby) Yang, and Stephen MacDonell

Vulnerability of Virtual Private Networks to Web Fingerprinting Attack..... 147
Khaleque Md Aashiq Kamal and Sultan Almuhammadi

Intrusion Detection Through Gradient in Digraphs 167
S. S. Varre, Muhammad Aurangzeb, and Mais Nijim

A Practice of Detecting Insider Threats within a Network 183
Jeong Yang, David Velez, Harry Staley, Navin Mathew, and Daniel De Leon

Toward Home Area Network Hygiene: Device Classification and Intrusion Detection for Encrypted Communications 195
Blake A. Holman, Joy Hauser, and George T. Amariuca

Part III Security Education, Training, and Related Tools

The Impact of Twenty-first Century Skills and Computing Cognition Cyber Skills on Graduates’ Work Readiness in Cyber Security..... 213
Anna J. Griffin, Nicola F. Johnson, Craig Valli, and Lyn Vernon

Enhancing the Cybersecurity Education Curricula Through Quantum Computation..... 223
Hisham Albataineh and Mais Nijim

CyberCheck.me: A Review of a Small to Medium Enterprise Cyber Security Awareness Program 233
Craig Valli, Ian Martinus, Jayne Stanley, and Michelle Kirby

Part IV Security, Forensics, Management and Applications

A Hybrid AI and Simulation-Based Optimization DSS for Post-Disaster Logistics..... 245
Gonzalo Barbeito, Dieter Budde, Maximilian Moll, Stefan Pickl, and Benni Thiebes

A Posteriori Access Control with an Administrative Policy 261
Farah Dernaika, Nora Cuppens-Boulahia, Frédéric Cuppens, and Olivier Raynaud

An Analysis of Applying STIR/SHAKEN to Prevent Robocalls 277
James Yu

Supervised Learning for Detecting Stealthy False Data Injection Attacks in the Smart Grid 291
Mohammad Ashrafuzzaman, Saikat Das, Yacine Chakhchoukh, Salahaldeen Duraibi, Sajjan Shiva, and Frederick T. Sheldon

Vulnerability Analysis of 2500 Docker Hub Images 307
Katrine Wist, Malene Helsem, and Danilo Gligoroski

Analysis of Conpot and Its BACnet Features for Cyber-Deception 329
Warren Z. Cabral, Craig Valli, Leslie F. Sikos, and Samuel G. Wakeling

Automotive Vehicle Security Metrics 341
Guillermo A. Francia, III

Requirements for IoT Forensic Models: A Review 355
Nawaf Almolhis, Abdullah Mujawib Alashjaee, and Michael Haney

Mobile Malware Forensic Review: Issues and Challenges 367
Abdullah Mujawib Alashjaee, Nawaf Almolhis, and Michael Haney

The Organizational Cybersecurity Success Factors: An Exhaustive Literature Review 377
Nancy Poehlmann, Kevin Matthe Caramancion, Irem Tatar, Yueqi Li, Mehdi Barati, and Terry Merz

A Hybrid Recommender System for Cybersecurity Based on a Rating Approach 397
Carlos Ayala, Kevin Jimenez, Edison Loza-Aguirre, and Roberto O. Andrade

Secure Stor: A Novel Hybrid Secure Edge Server Architecture and CDN to Enhance the Security and Response Time for Edge Devices . 411
Mais Nijim, Raghava Reddy Marella, Muhammad Aurangzeb, and Moustafa Nasralla

Leveraging Security Management with Low-Level System Monitoring and Visualization 421
Karlen Avogian, Basel Sababa, Ioanna Dionysiou, and Harald Gjermundrød

Lightweight Network Steganography for Distributed Electronic Warfare System Communications 437
Tim Lei, Jeremy Straub, and Benjamin Bernard

Security of DBMSs 449
Suhair Amer

Static Analysis for Software Reliability and Security 463
Hongjun Choi, Dayoung Kang, and Jin-Young Choi

Part V Wireless Networks, Novel Technologies and Applications

A Tool for the Analysis of MANET Routing Protocols Based on Abstract State Machines	473
Alessandro Bianchi, Emanuele Covino, Giovanni Pani, and Sebastiano Pizzutilo	
A New Real-Time Geolocation Tracking Tool Enhanced with Signal Filtering	491
Erkan Meral, Mehmet Serdar Guzel, Mehrube Mehrubeoglu, and Omer Sevinc	
A Self-adaptivity Indoor Ranging Algorithm Based on Channel State Information with Weight Gray Prediction Model	503
Jingjing Wang and Joon Goo Park	
Autonomous Vehicle Security Model	513
Noha Hazzazi, Kevin Daimi, and Hanady Issa	
Wi-Fi Direct Issues and Challenges	525
Rabiah Alnashwan and Hala Mokhtar	
RFID Assisted Vehicle Navigation Based on VANETs	541
Yang Lu and Miao Wang	
Regular Plans with Differentiated Services Using Cuckoo Algorithm	555
John Tsiligaridis	
Using Multimodal Biometrics to Secure Vehicles	567
Kevin Daimi, Noha Hazzazi, and Mustafa Saed	

Part VI Internet Computing, Internet of Things, and Applications

Per-user Access Control Framework for Link Connectivity and Network Bandwidth	587
Shogo Kamata, Chunghan Lee, and Susumu Date	
Comparative Study of Hybrid Machine Learning Algorithms for Network Intrusion Detection	607
Amr Attia, Miad Faezipour, and Abdelshakour Abuzneid	
Unquantize: Overcoming Signal Quantization Effects in IoT Time Series Databases	621
Matthew Torin Gerdes, Kenny Gross, and Guang Chao Wang	
Information Diffusion Models in Microblogging Networks Based on Hidden Markov Theory and Conditional Random Fields	637
Chunhui Deng, Siyu Tang, and Huifang Deng	

ISLSTM: An Intelligent Scheduling Algorithm for Internet of Things	655
Fred Wu, Jonathan Musselwhite, Shaofei Lu, Raj Vijeshbhai Patel, Qinwen Zuo, and Sweya Reddy Dava	
The Implementation of Application for Comparison and Output of Fine Dust and Public Database Using Fine Dust Sensor	669
YunJung Lim	
Dynamic Clustering Method for the Massive IoT System	683
Yunseok Chang	
A Network Traffic Reduction Method for a Smart Dust IoT System by Sensor Clustering	693
Joonsuu Park and KeeHyun Park	
 Part VII Embedded Systems, Cyber-physical Systems, Related Tools, and Applications	
On the Development of Low-Cost Autonomous UAVs for Generation of Topographic Maps	701
Michael Galloway, Elijah Sparks, and Mason Galloway	
Wireless Blind Spot Detection and Embedded Microcontroller	717
Bassam Shaer, Danita L. Marcum, Curtis Becker, Gabriella Gressett, and Meridith Schmieder	
BumpChat: A Secure Mobile Communication System	731
Brian Kammourieh, Nahid Ebrahimi Majd, and Ahmad Hadaegh	
Data Collection and Generation for Radio Frequency Signal Security	745
Tarek A. Youssef, Guillermo A. Francia, III, and Hakki Erhan Sevil	
Real-Time Operating Systems: Course Development	759
Michael Rivnak and Leonidas Deligiannidis	
Piano Player with Embedded Microcontrollers	777
Bassam Shaer, Garrick Gressett, Phillip Mitchell, Joshua Meeks, William Barnes, and Stone Hewitt	
Software-Defined Global Navigation Satellite Systems and Resilient Navigation for Embedded Automation	791
Jeffrey Wallace, Angelica Valdivia, Srdjan Kovacevic, Douglas Kirkpatrick, and Dubravko Babic	
Smart Automation of an Integrated Water System	805
F. Zohra and B. Asiabanpour	
Quadratic Integer Programming Approach for Reliability Optimization of Cyber-Physical Systems Under Uncertainty Theory	821
Amrita Chatterjee and Hassan Reza	

Brief Review of Low-Power GPU Techniques 829
Pragati Sharma and Hussain Al-Asaad

Ethical Issues of the Use of AI in Healthcare 843
Suhair Amer

Index..... 855

Part I
Authentication, Biometrics, and
Cryptographic Technologies

Statistical Analysis of Prime Number Generators Putting Encryption at Risk



Aykan Inan

1 Introduction

When it comes down to investigating the security properties of a cryptographic procedure there are different methods to do so, depending on the cryptoscheme itself. This includes inter alia, protocol, side-channel, and mathematical attacks. But the security of a strong cryptographically system is primarily based on the secure management of the secret key. If this key can be easily accessed or even worse guessed by the attacker the system is compromised. No matter how strong the used encryption method itself is. Therefore, it is of great importance that the secret key cannot be revealed in any case. Storing the key securely is one matter but the unpredictability is another [1].

Some cryptoschemes, such as RC4, rely on a random stream. Others, such as RSA need a PNG in order to generate two primes for generating the public and corresponding private key. Therefore, randomness for both “normal” random numbers and primes plays a major role.

Random number and prime number generators (RNGs and PNGs) are typically used when a cryptographic scheme needs a random number or random prime number to some extent. Random prime number generators have additional features as general random number generators: Any number generated needs to be odd and needs to be tested for primality afterwards. The following section gives an overview over the current state of PNGs. Section 3 demonstrates the approach used in this paper to verify the PNGs randomness. Sections 4 and 5 analyze specific outliers

A. Inan (✉)

Ravensburg-Weingarten University, Weingarten, Baden-Württemberg, Germany
e-mail: aykan.inan@hs-weingarten.de

© Springer Nature Switzerland AG 2021

K. Daimi et al. (eds.), *Advances in Security, Networks, and Internet of Things*,
Transactions on Computational Science and Computational Intelligence,
https://doi.org/10.1007/978-3-030-71017-0_1

within the randomness spectrum. Section 6 looks for patterns within primes, and Sect. 7 for patterns in the last 32 to 64 bits. The last Sect. 8 concludes and gives an outlook.

2 Related Work and Basics

This section is splitted in two parts: while Sects. 2.1 and 2.2 discusses deterministic (Sect. 2.1) and non-deterministic (Sect. 2.2) random number generators, Sect. 2.3 gives an overview on prime-number generators. A key requirement for both types of random number generators is that their output cannot be reproduced or predicted [2]. There are many mathematical tests such as the chi-square test, which can verify the statistical behavior of RNGs or PNGs sequences.

2.1 Deterministic RNG

A deterministic random number generator is always producing the same sequences of random numbers under the same circumstances. That is why they are also called pseudo-random number generators (PRNG). But the produced consecutive numbers appear to be random enough for most applications. The generated sequence of a PRNG is computed recursively from an initial seed value initializing a function $f(s_0)$:

$$\begin{aligned} s_0 &= \text{seed} \\ s_{i+1} &= f(s_i), \quad i = 0, 1, \dots, i \in \mathbb{N}_0 \end{aligned} \tag{1}$$

In general, the generated sequence can be described as:

$$s_{i+1} = f(s_i, s_{i-1}, \dots, s_{i-t}) \tag{2}$$

In this case t is representing an integer constant.

Consequently, a PRNG does not generate true random numbers in a proper or true sense because it is computing its random numbers initialized from a starting (seed) value. Thus, it is completely deterministic [2]. According to Manuel Blum and Silvio Micali [3] a polynomial algorithm should not be capable of predicting and computing the next sequence better than 0.5 (50%) chance of success without knowing the initial seed value. For this, different mathematical tests are being used to prove the correctness.

2.2 *Non-deterministic RNG*

In contrast to the deterministic RNG a non-deterministic random number generator includes external source of randomness (entropy) such as hardware noise or the current time [1]. They are also known as cryptographically secure pseudo-random number generators (CSPRNG) and can be seen as a special type of PRNG which represents an unpredictable PRNG [2].

Assuming we have the following output sequence of n bits, where n is representing some integer:

$$S_i, S_{i+1}, \dots, S_{i+n-1} \quad (3)$$

Then it must be computationally infeasible to compute the subsequent bits:

$$S_{i+n}, S_{i+n+1} \dots \quad (4)$$

PNRGs and CSPRNGs are described and defined as an algorithm that is producing an unpredictable sequence of random numbers in such a way that an attacker is not capable of computing or guessing them. This means that all generated random numbers must have the same likelihood of occurrence.

The characteristic of fully randomness can only be fulfilled with a One-time-pad which is, however, unsuitable for practical applications. So, the solution is to use pseudo-random numbers or pseudo-random sequences which are based on a deterministic process. Despite of this deterministic behavior and the use of an initialization seed value the produced output still must have the property of a truly random sequence [1].

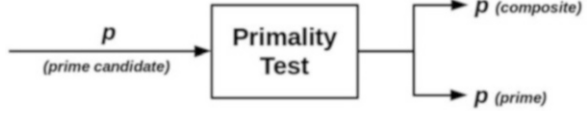
In this context it is important to point out that the key generation in asymmetrical procedures usually requires some more effort than the generation process of pseudo-random numbers used in symmetrical systems. This is because an asymmetric cryptoscheme, such as RSA, requires large primes. Thus, the produced random numbers must fulfill the above-mentioned properties as well as the category of being prime at the same time [1, 4]. For this purpose Prime Number Generators are needed.

2.3 *Prime Number Generator*

In practice it is common to work with pseudo-prime numbers which fulfill most basic requirements for primes such as producing an odd number. Then a primality test, usually Miller-Rabin [5], is applied as depicted in Fig. 1.

The likelihood that a randomly picked or generated integer p is a prime is of further interest. In case of RSA, e.g., in order to generate a 1024-bit modulus n , the two primes p and q each should have a length of about 512 bits [4]. The chance that

Fig. 1 Approach to generate primes



a random integer of that size is prime is still sufficiently high based on the prime number theorem and is approximately $\frac{1}{\ln(p)}$ as shown below [2, 6]:

$$p \text{ is prime} \approx \frac{2}{\ln(p)} = \frac{2}{\ln(2^{512})} = \frac{2}{512 \ln(2)} \approx \frac{1}{177} \quad (5)$$

This means on average that 177 random numbers must be generated and tested before finding a prime. The density of primes, for even much larger bit numbers, is still adequate high [2].

This is relevant if similar prime numbers are generated. Often a value is then added to the result if it fails the primality test until a prime number is finally found. But this can lead to similar generated numbers.

Consequently, a prime used in RSA must be unpredictable. However, if at least one of the primes is easily obtained, RSA would be broken. This paper therefore analyses whether our current process of determining the quality of randomness for primes is still valid.

2.4 Evaluation of PRNG

Every published analysis dealing with PNGs or RNGs such as [3, 7–10] are just focusing on this unpredictability of subsequent sequences. As long as the produced output, e.g., a pair of two primes, is unique compared to the previous one, then the primes are sufficient enough for cryptographical use.

However, by generating more than one billion primes of specific bit lengths, (32, 64, 128, 256, and 512) and displaying the result into different statistics as presented in the following Sects. 3–7. The generated prime numbers show similar characteristics and seem related to each other.

A statistical analysis of the PNG used in LibreSSL was conducted. The results demonstrate suspicious behavior indicating that the numbers are not fully random as they seem.

3 Statistical Analysis

The statistical analysis is based on two essentials aspects:

- (a) Prime Numbers
- (b) Prime Distances

Each aspect itself is separated into sub-aspects again.

- (a.1) Smallest prime number
- (a.2) Largest prime number
- (a.3) Mean prime number

The exact same statistical approach applies to the generated corresponding distances between prime numbers:

- (b.1) Smallest distance
- (b.2) Largest distance
- (b.3) Maximum distance between (b.1) and (b.2)
- (b.4) Mean distance

In general the statistics are showing the following relationship between two consecutive generated primes, as depicted in Fig. 2 and all generated primes in general:

Furthermore a variance analysis and standard deviation will be presented and explained in Sect. 3.3 for the prime numbers and the prime distances in Sect. 3.7. Due to the large amount of data and the large numbers involved the following subsections are going to present the most outstanding properties with regard to the above-mentioned listing in (a) and (b) and the specific bit lengths of 32, 64, 128, 256, and 512 bit because they are most commonly used.

3.1 Largest and Smallest Prime Numbers

The largest and smallest prime numbers that have ever occurred are listed in Tables 1 and 2.

Although these specific numbers did not occur very frequently they give a good reference point to search for boundaries and patterns within and among other primes. This includes, among other properties, the occurrences of numbers near threshold values (see Sects. 4 and 5), such as the largest and smallest prime.

Fig. 2 Distance between p and q



Table 1 Largest prime numbers

Bit	Value			Occurrence
32	4,294,967,291			13
64	18,446	...	876,649	2
128	340,282	...	715,813	2
256	115,792	...	191,919	1
512	13,407	...	162,089	1

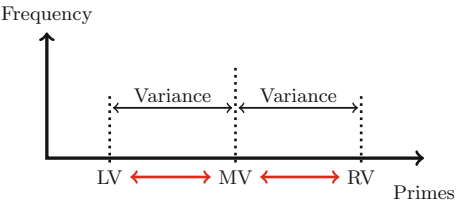
Table 2 Smallest prime numbers

Bit	Value	Occurrence
32	3,221,225,473	3
64	13,835 ... 607,023	2
128	255,211 ... 346,557	1
256	86,844 ... 462,243	1
512	100,558 ... 779,483	1

Table 3 Mean value of all prime numbers

Bit	Mean value
32	3,756,397,796
64	16,138 ... 636,565
128	297,724 ... 539,463
256	101,315 ... 281,316
512	11,731 ... 587,394

Fig. 3 Variance analysis



3.2 Mean Value of Prime Numbers

The mean value is calculated via all generated prime numbers which then was used to determine the standard deviation which will be explained in the following subsection. Table 3 shows the results.

3.3 Standard Deviation of Prime Numbers

The variance analysis as shown in Fig. 3 was conducted with regard to the standard deviation. This applies to both the distance and the prime analysis. The evaluation of the standard deviation provides good matches with the previous results.

The abbreviations used in the following tables and figures as well as in the above Fig. 3 are:

- SP: Smallest prime

LP: Largest prime

SD: Smallest distance

LD: Largest distance

LV: Left value to MV

RV: Right value to MV
- MV: Mean value

xDR: x delta right

xDL: x delta left

Sx: Sector x

LD: Last digit

Fig. 4 Variance analysis of prime numbers

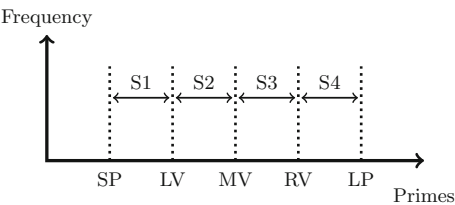


Table 4 Statistical distribution of all prime numbers

Bit	Sector 1 (S1)	Sector 2 (S2)	Sector 3 (S3)	Sector 4 (S4)
32	21.13%	28.95%	28.78%	21.13%
64	21.13%	28.90%	28.84%	21.13%
128	21.13%	28.88%	28.85%	21.13%
256	21.14%	28.87%	28.86%	21.13%
512	21.55%	28.72%	28.71%	21.02%

To get the first overview the huge number space was divided into four different sectors to analyze the distribution as shown in Fig. 4.

The largest and smallest prime numbers found marked the upper and lower boundaries while the mean value is marking the approximate center. The left and right value to the mean value helped again to separate the number range. Table 4 shows the proportional distribution of the primes for each sector and bit length.

First of all Table 4 reveals that the distribution for every bit length is almost identical. But it proved at the same time that for every bit length most of the generated primes, approximately 60%, are located between LV and RV.

3.4 Largest and Smallest Prime Distances

Looking at the distances is relevant due to the fact that they can provide information about the properties and dependencies between two consecutive generated primes. This can be very important, i.e., in relation to improve a prime factorization processes and knowing the approximate location of two primes. The largest and smallest prime distances that occurred are listed in Tables 5 and 6.

Table 5 Largest prime distance

Bit	Value	Occurrence
32	1,073,726,616	1
64	4611 ... 887,704	1
128	85,069 ... 365,142	1
256	28,947 ... 658,100	1
512	335,188 ... 410,182	1

Table 6 Smallest prime distance

Bit	Value	Occurrence
32	2	3
64	2,031,919,274	1
128	97,049 ... 555,744	1
256	24,277 ... 917,692	1
512	1982 ... 318,524	1

Table 7 Distance between two generated primes

Bit	Maximum distance
32	1,073,726,608
64	4611 ... 968,430
128	85,069 ... 094,649
256	28,947 ... 018,638
512	355,188 ... 303,022

Table 8 Mean value of all prime distances

Bit	Mean value
32	35,794,328
64	1537 ... 847,280
128	28,357 ... 762,191
256	9649 ... 455,788
512	1173 ... 562,455

3.5 *Maximum Distance*

The maximum distance was computed via the results of the largest and smallest distance. The results are shown in Table 7.

3.6 *Mean Value of Prime Distances*

The result for the mean value of the primes distances is shown in Table 8.

3.7 Standard Deviation of Prime Distances

Table 9 shows the proportional distribution of the prime distances for each sector and bit length.

Both standard deviations show similar results. However, the deviation of the distances in sector 2 is prominent. Again, sector 2 and 3 together contain most of the generated distances. In this case approximately 63%.

4 Occurrence of Primes Near the Threshold Values

Given the fact that the generated integers are becoming increasingly larger with the increasing length of bits the analyzed scope was adjusted based on the discovered pattern of the largest prime. Thus, these patterns were used in order to locate other primes within a predefined range. The most important ones are shown in Table 10.

The first digits of a prime are mainly responsible for the specific pattern with regard to the threshold value. The delta (Δ) range was chosen based on the position of the last digit of the found pattern. The idea is illustrated using the 32 bit value in Table 11 where x is representing any digit. The last digit of the pattern ends in the fourth position to the right. So, this position is set to one filled with zeros in the delta value.

The classification of the number line is depicted in Fig. 5. The delta (Δ) between each sector represents the searching area for close related numbers in relation to the threshold value.

In case of the 32 bit pattern you can create a specific amount of groups of patterns ($\Delta_{Pattern-Range}$) computed via the difference between both patterns as shown in Eq. (6). This then represents the actual search range for these specific patterns. In

Table 9 Statistical distribution of all prime distances

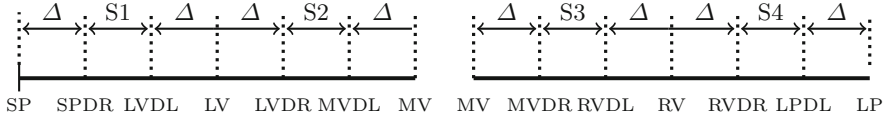
Bit	Sector 1 (S1)	Sector 2 (S2)	Sector 3 (S3)	Sector 4 (S4)
32	18.58%	36.98%	25.78%	18.57%
64	18.57%	36.98%	25.87%	18.57%
128	18.57%	36.98%	25.87%	18.57%
256	18.57%	36.98%	25.87%	18.57%
512	18.57%	36.98%	25.87%	18.57%

Table 10 Patterns for LP, SP

Bit	Pattern LP	Pattern SP
32	4,294,967 ...	3,221,225 ...
64	18,446,744 ...	1,383,505,806 ...
128	34,028,236 ...	255,211,775 ...
256	115,792,089 ...	8,684,406 ...
512	1,340,780 ...	100,558,559 ...

Table 11 Patterns for LP, SP

Pattern	Chosen Δ
4,294,967,xxx	1000
3,221,225,xxx	1000

**Fig. 5** Likelihood of occurrence near the threshold values**Fig. 6** Dissemination of patterns

this case it is computed as followed:

$$\begin{aligned}
 \Delta_{Pattern-Range} &= Pattern_{LP} - Pattern_{SP} \\
 &= 4,294,967 - 3,221,225 \\
 &= 1,073,742
 \end{aligned}
 \tag{6}$$

Figure 6 depicts the dissemination of all patterns across the entire number range. In this respect only the 32 bit values are displayed again.

As can be seen the minimum amount of patterns is distributed approximately evenly above a certain threshold value indicated with the red line. On the other hand it shows a slowly but surely decreasing distribution approaching to the threshold value which can be traced back to the fact that the integers are getting larger.

Figure 7 is depicting a much smaller range showing that there are areas of a greater concentration of patterns as indicated with the red circles. In this view these accumulations can be seen as hills in the graph.

Table 12 shows the preliminary results of the first analysis of the percentage distribution of primes within certain areas. The numbers in brackets present the percentage of the amount of found located primes. As a matter of fact the frequency of occurrence will start to drop as greater the bit size becomes. But the relative frequency is still very high in relation to the bit size.

Fig. 7 Accumulations of patterns

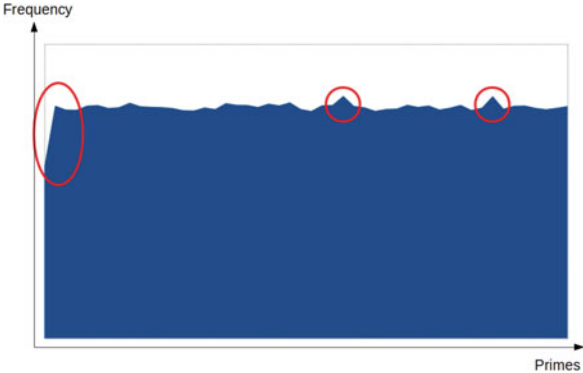


Table 12 Percentage distribution of primes

Section	Bit 32	Bit 64	Bit 128	Bit 256	Bit 512
SD:	0.09%	0.22%	0.12%	0.35%	0.56%
LV:	0.19%	0.43%	0.24%	0.69%	0.06%
MV:	0.19%	0.43%	0.24%	0.69%	0.06%
RV:	0.19%	0.22%	0.23%	0.69%	0.06%
LD:	0.09%	0.00%	0.12%	0.35%	0.03%
Sector-1:	20.95%	20.70%	20.90%	20.45%	20.96%
Sector-2:	28.76%	28.46%	28.65%	28.18%	28.66%
Sector-3:	28.60%	28.41%	28.62%	28.17%	28.65%
Sector-4:	20.95%	20.70%	20.90%	20.44%	20.96%

Table 13 Patterns for LD, SD

Bit	Pattern LP	Pattern SP
32	1073 ...	2 ...
64	46,116 ...	20,319 ...
128	8506 ...	97,049 ...
256	2894 ...	24,277 ...
512	3351 ...	1982 ...

5 Occurrence of Distances Near the Threshold Values

The exact same procedure as described with the primes in the previous section was conducted with the distances, too. Table 13 shows the patterns for the largest and smallest distances while Table 14 illustrated using the 32 bit value to find out the corresponding delta. In this respect, it should be noted that the delta for the smallest distance was set to the same value as for the largest distance due to the small factor for the smallest distance. Table 15 presents the final results.

In addition to Tables 12 and 15 a more detailed analysis is still underway but the first analysis already shows that the generated primes can be separated into groups of patterns. Then these patterns can be assigned to the different sectors

Table 14 Patterns for LD, SD

Pattern	Chosen Δ
1073 xxx xxx	1,000,000
2	1,000,000

Table 15 Percentage distribution of distances

Section	Bit 32	Bit 64	Bit 128	Bit 256	Bit 512
SD:	0.19%	0.04%	0.02%	0.07%	0.06%
LV:	0.34%	0.08%	0.04%	0.12%	0.11%
MV:	0.25%	0.06%	0.03%	0.09%	0.08%
RV:	0.16%	0.04%	0.02%	0.06%	0.05%
LD:	0.00%	0.00%	0.00%	0.00%	0.00%
Sector-1:	18.22%	18.49%	18.53%	18.44%	18.46%
Sector-2:	36.69%	36.91%	36.94%	36.87%	36.89%
Sector-3:	25.67%	25.82%	25.85%	25.80%	25.81%
Sector-4:	18.49%	18.55%	18.56%	18.55%	18.55%

and narrowed down the area of concentration where most of the primes are primarily located. This means that the actual located primes within that range is supposed to be much higher considering that some patterns only differ in one digit of the primary pattern. Currently it can be stated that both distances and primes have a noticeable behavior. While the distances seem to be more concentrated near the mean value the primes seem to have a close proximity to some threshold values with fluctuations in between. But this assumption still needs to be proven in detail.

What definitely can be said about this property is that both the smallest and largest prime numbers have some patterns to adjacent numbers that are close to the maximum and minimum values. In the case of the largest 512 bit prime the most noticeable property is the fact that all smaller numbers are closely related to each other and lap with the first six digits. The same applies to the smallest prime numbers and to all other bit sizes.

6 Patterns Within Primes

The search for patterns is very complex and difficult. However, first tests show that there are not only patterns at the beginning of a prime but also within a prime. This will be demonstrated with a small example of a 64 bit number as shown below:

17261221124532159023
17267101136670495809

The multiple patterns are marked in red. Patterns of a similar kind are very probable but, as already mentioned, are very difficult to find. The more such patterns

Table 16 Likelihood of occurrence of the last digit

Bit	1	3	7	9
32	25.50%	23.78%	26.07%	24.65%
64	25.52%	23.73%	26.10%	24.65%
128	0%	33.41%	34.06%	32.53%
256	0%	33.36%	34.02%	32.62%
512	0%	33.41%	34.06%	32.62%

are found within a prime number, the more advantageous it is to find other prime candidates. This has two decisive consequences. On the one hand, the number space is automatically restricted and on the other hand the probability of finding possible prime numbers increases due to the classification into groups of patterns.

7 Searching for Last Digits

Another special feature of primes is the fact that the last digit of a prime candidate can either end with 1, 3, 7, or 9. The numbers 2 and 5 are excluded in this consideration because they are the only exception to this rule. For this reason it was analyzed how evenly the primes are distributed with regard to their last digit. The final result of this analysis is shown in Table 16.

As can be seen every last digit appears nearly with the same likelihood within the absolute values for every bit length. The last digit of 7 appears the most followed by 1, 9, and 3. However, this applies only to integers with a bit length of 32 and 64 bit. Much more prominent is the result shown in the second column representing the last digit of 1. Integers at a length of 128 bit and above do not show any single prime number ending with the digit of 1. 7 is still the most common followed again by 3 and 9. The analysis of this result is still ongoing.

8 Benefits from the Statistics

The first analysis found several aspects including block patterns of same digits within the prime number itself as well as among several primes. Furthermore, the generated primes seem to be generated within a certain undocumented upper and lower boundary, which means that the generated primes do not exceed or fall below a certain generated limit. But the most remarkable property is the fact the generated prime numbers do not differ much from each other as one could assume. Whether or not the primes were produced at once or in several runs and on different machines. All in all this knowledge already reveals a lot about the generation process and further reduces randomness as a consequence of this. This entire knowledge can be of great importance when it comes to factorizing a large integer and to accelerate

and facilitate the factorization process. Knowing and visualizing the proportional distribution of primes as well as their related distances and their relationship of distance can be a tremendous help in cracking keys that are based on PNGs and used in RSA, for instance. This allows a faster prime factorization within a predefined range based on the properties and relationships. This knowledge can also be used to increase the forecast probability for a number being a potential prime key candidate to search for.

9 Conclusions

This paper outlines and indicates that the current implementations of LibreSSL and the one-sided approach to analyze PNGs are not sufficient enough to prove randomness. However, this still requires further analysis including sorting and analyzing the primes for these patterns as well as an intense code audit in order to find correlations between the available statistics and the generation process itself. This code audit is temporarily using LibreSSL as an example. A parallel analysis of GNU Multiple Precision Arithmetic Library is underway and being set up so that the results can be compared regarding their predictability and security. As a result, this work might demonstrate that the current status “symmetric encryption cannot be undone without the private key” might be wrong and thereby enable investigators to decipher encrypted data.

References

1. C. Eckert, *IT-Sicherheit: Konzepte - Verfahren - Protokolle*, 8th edn. (Oldenbourg, München, 2013), pp. 327–329, 431–433
2. C. Paar, J. Pelzl, *Understanding Cryptography - A Textbook for Students and Practitioners*, 2nd corrected printed (Springer, Berlin, 2010)
3. M. Blum, S. Micali, How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.* **13**(4), 850–864 (1984)
4. R.L. Rivest, A. Shamir, L. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, vol. 21(2) (Communications of the ACM, New York, 1978), pp. 120–126
5. D. Knuth, *The Art of Computer Programming*, vols. I–III, 3rd edn. (Addison-Wesley Pub Co, Boston, 1997)
6. M. Schubert, *Mathematik für Informatiker - Ausführlich erklärt mit vielen Programmbeispielen und Aufgaben* (Vieweg+Teubner Verlag, Wiesbaden, 2009)
7. Bundesamt für Sicherheit in der Informationstechnik, *Quellcode-basierte Untersuchung von kryptographisch relevanten Aspekten der OpenSSL-Bibliothek*, Projekt 154, Bonn, Version 1.2.1, 2015-11-03
8. Bundesamt für Sicherheit in der Informationstechnik, *Documentation and Analysis of the Linux Random Number Generator*, Bonn, Version 3.5, 2019-12-13
9. L. Accardi, M. Gäbler, *Statistical Analysis of Random Number Generators* (2011), pp. 117–128. https://doi.org/10.1142/9789814343763_0009
10. R. Fudjak, J. Misurec, P. Mlynek, *Analysis of Random Number Generator from Texas Instrument in MSP430 x5xx Families* (2014). <https://doi.org/10.1109/TSP.2015.7296344>

Secure Authentication Protocol for Drones in LTE Networks



Dayoung Kang, Gyuhong Lee, and Jin-Young Choi

1 Introduction

In December 2016, Amazon, the largest e-commerce company in the USA, succeeded in delivering a delivery service using a drone to a farm in Cambridge, England [15]. The time from order to delivery was only 13 min, and people began to notice the possibility of using commercial drones. Drones began to be developing for military use during World War I in the USA [16], but now they are in higher demand from the civilian sector. The drone applications are diverse, such as disaster management, search and rescue, agricultural use, and the arts, and the expected economic effect is excellent. In order to operate and control drones in vast areas, diverse methods of communication have been developed. Cellular networks have become highly attractive to assist with drone identification, authentication, and communication, and people started to get interested in drones [18].

Even though we are transitioning from LTE to 5G, more than 50% of GSM subscribers use LTE networks for communication in 2020. GSMA (GSM Association) announces that the number of 5G connections will reach 1.4 billion by 2025, 20% of total connection. Nevertheless, 4G (LTE) will continuously grow over this period, accounting for 56% of global connections by 2025 [11].

There are various vulnerabilities on LTE networks [6], but we are focusing on the security vulnerability that could reveal the unique identifier (IMSI) of mobile subscribers during mutual authentication and key exchange. If drone pilots used LTE

D. Kang (✉) · J.-Y. Choi

The Graduate School of Information Security, Korea University, Seoul, Republic of Korea
e-mail: dayokiki@korea.ac.kr; narnia@korea.ac.kr

G. Lee

The Department of Cyber-Warfare, Korea Army Academy at Yeongcheon, Yeongcheon, Gyeongsangbuk-do, Republic of Korea

© Springer Nature Switzerland AG 2021

K. Daimi et al. (eds.), *Advances in Security, Networks, and Internet of Things*,
Transactions on Computational Science and Computational Intelligence,
https://doi.org/10.1007/978-3-030-71017-0_2

for drone communication and a malicious adversary obtained IMSI of the drone, the adversary could conduct passive or active attacks related to location privacy. If a drone leaks its location, a malicious adversary can predict the frequency, destination, starting point, or mission of the drone. Therefore the leakage of IMSI needs to be addressed for the security and privacy of LTE drones.

In this paper, we first review various studies [12, 19, 21, 22] on how to improve the LTE authentication protocol for mobile phones and show that these techniques are not fit to LTE drone, which means that we need an authentication protocol suitable for LTE drones. We describe the operation concept of a drone using LTE, a network concept based on LTE, and a general user authentication process of the LTE network. Later, we perform a security analysis to derive risks about leakage of IMSI. Next, we proposed a key authentication protocol suitable for LTE drones as a countermeasure. Finally, we specify and verify the proposed protocol using Scyther [9], an automated protocol verification tool.

2 Related Works

Since there does not exist studies related to the LTE authentication protocol specialized in LTE drones, we refer to a paper that poses the possibility of IMSI leakage during the authentication process of a mobile phone on an LTE network and proposes various security protocols.

Broek et al. [21], in 2015, proposed Dynamic ID-based authentication. The author insisted that Home Subscription Server (HSS) must authenticate the subscriber and the key exchanged by receiving the Pseudo IMSI (PMSI), instead of receiving the ISMI from the SIM user at the initial stage. After shared with the HSS, the subscriber stored two PSMIs in the SIM. The SIM subscriber receives an Authenticate from the server by throwing one of the PSMIs rather than ISMI at the initial phase. At this time, HSS discard the PMSI used for authentication and make a new PMSI, and the subscriber receives the new PMSI and use it for the next communication.

Norrman et al. [19] proposed that a SIM subscriber sends the IMSI encrypted with a public key of HSS to the HSS so the IMSI would not be leaked. Since the HSS's private key is required to decrypt the IMSI encrypted with the HSS's public key, a malicious adversary without the private key cannot decrypt the ciphertext and get the IMSI.

Hsieh et al. [12] provided an authentication protocol that used One-Time Passwords (OTPs) based on the time and location information to authenticate the subscriber securely. When the mobile subscriber transmits the location information to the HSS, the HSS can predict the next moveable distance compared with the previously received location information. The author claimed that if the mobile phone subscriber was outside the predictable travel distance, the HSS could not authenticate the subscriber.

Abdrabou et al. [5] proved that the EPS-AKA protocol used for LTE authentication transmitted an unencrypted IMSI and proposed an authentication protocol modified the EPS-AKA protocol. For this, the proposed authentication algorithm should be stored in USIM on subscriber-side and in the authentication center on the home subscriber server-side.

The above-discussed authentication protocols proposed an improvement of the current protocol and modification of the existing infrastructure, but these were studies on mobile phone subscribers, not drones. In this paper, assuming that the current infrastructure is not changed, we propose an authentication protocol for LTE drones that utilizes the secure channel of the drone operating system. In particular, we introduce an authentication protocol suitable for LTE drone operation using Asymmetric cryptography and 2-Factor Authentication.

3 LTE Drone Control System

3.1 General System Architecture

The drone control system is divided into a UAV (Drone) and a ground control station (GCS) that commands and controls the drone, and for mutual communication, there exist data link to transmit and receive information between the drone and the GCS.

We can divide the drone data link into two types [22], and one is a peer-to-peer link where the drone is directly connected and controlled to the GCS, another is a network-type link where the drone connects to the GCS using a high-speed wireless network such as LTE or IEEE 802.16. There is a message protocol (e.g., MAVLink [1]) used to exchange messages between the drone and GCS, and the drone transmits the MAVLink message to the GCS over the LTE networks. In this paper, we discuss a network-type link drone control system using LTE.

The drone performs through a base station (eNodeB), SGW (serving gateway), and PGW (packet gateway) to communicate with GCS. The mutual authentication between the drone and the LTE network is required at the initial phase. For this authentication, the required information is stored in the drone's SIM and the HHS/AuC (home subscriber server/authentication center) operated by the subscribed cellular carriers (e.g., Verizon, AT&T). Also, there exists MME (mobility management entity), which is responsible for initiating authentication of the drone device.

Figure 1 presents the architecture of the drone control system using LTE networks [7].

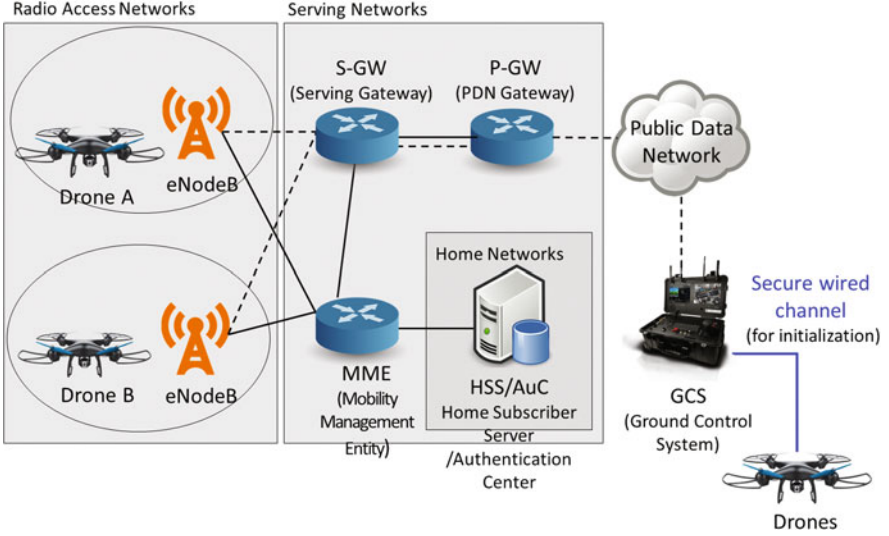


Fig. 1 Using LTE for drone control system

3.2 LTE Authentication Protocol

The drone needs to be authenticated by HSS/AuC to use LTE networks, but the drone and the HSS do not send and receive authentication information directly, but perform the authentication procedure through the MME [7, 22].

The drone sends an Attach Request with authentication information (IMSI, UE Network Capability) in plaintext to make a connection from the MME at the initial phase.

Upon receiving the Attach Request from the drone, the MME sends an Authentication Data Request (IMSI, SN id, Network Type) to the HSS to authenticate the drone.

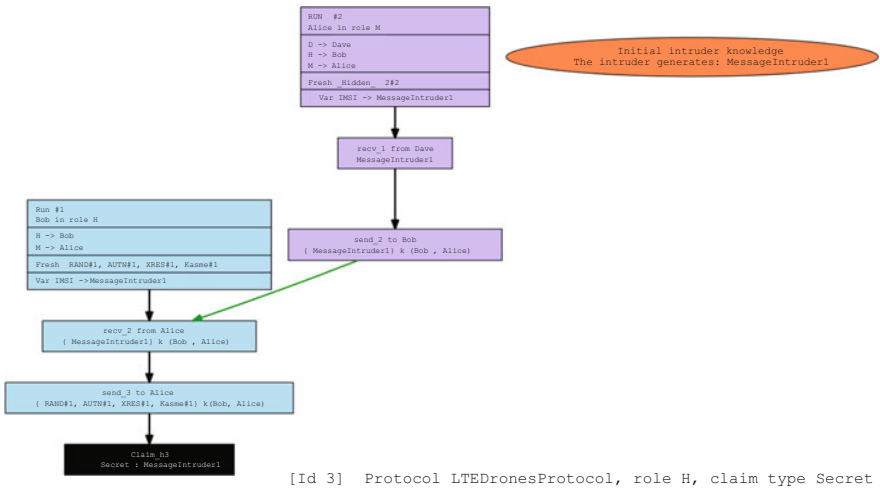
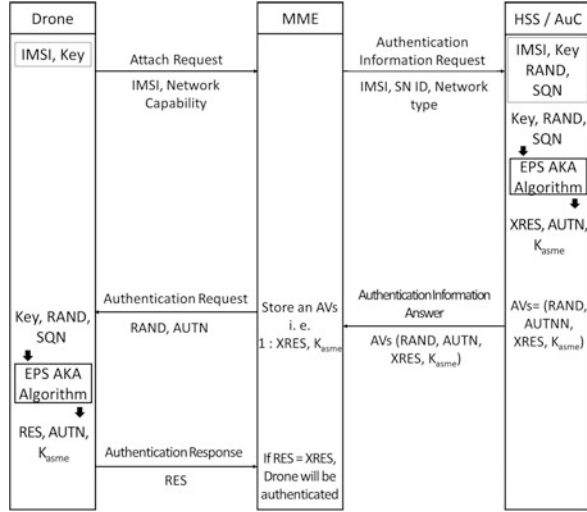
The HSS puts the received Authentication Data into the EPS-AKA algorithm, obtains Authentication Vector (RAND, XRES, AUTN, and K_{asme}) as a result, and sends the AVs to the MME. After receiving the Authentication Vectors (AVs) from HSS, the MME stores the AVs and sends an Authentication Request (RAND, AUTN) to the drone to request authentication.

The drone that received the Authentication Request puts it into the EPS-AKA algorithm in SIM, acquires AUTN, RES, K_{asme} , and sends RES to MME.

On receiving RES from the drone, the MME compares it to the XRES in possession. Finally, the MME completes the authentication procedure if it is the same as the XRES it has.

After that, MME set the encryption method, and the Drone, MME, and HSS have secure communication performed using the K_{asme} owned by both sides.

Figure 2 shows how HSS authenticates drone through MME according to the LTE standard [2–4].

Fig. 2 Authentication and key agreement protocol**Fig. 3** Attacks for claim secret IMSI

3.3 Security Analysis

We specified the ESP-AKA protocol using Scyther, which is a protocol verification tool. We verified that a malicious intruder could sniff the IMSI over the LTE networks, and pretend to be the drone. We will introduce Scyther in Sect. 5, and the result of the security analysis is shown in Fig. 3.

According to Mjølunes et al. [17], even if an adversary has not experienced high-level hacking, it is easy for him to reveal the IMSI in LTE. Once the adversary sniffs IMSI, the threats that will affect the drone missions are as follows :

Disclosure of the Drone Identity A drone user must be authenticated by HSS to access the LTE networks at the first access. During the authentication, the drone transmits his IMSI in plaintext. Thus, an adversary could intercept the IMSI through a sniffing attack. Once the adversary sniffed IMSI, the adversary could obtain subscriber information, location privacy, and conversation information. Furthermore, the attacker could hide the real drone user and commit other cyber attacks such as DoS by using the IMSI [5].

Man in the Middle (MITM) Attack at the Authentication Phase A malicious adversary can always intercept an Authentication Request sent to MME by disguised as a Rogue base station. If the adversary gets the drone's IMSI, it can hold both the IMSI and the Authentication Request, so a MITM attack is possible[5].

Denial of Service A malicious attacker attempts to access the network by sending a Fake Attach request continuously to the MME pretended to be a drone after intercepting the IMSI and Authentication Request of the Drone. As the adversary attacks, MME and HSS consume its computational powers, and normal drones cannot access the network. Therefore, the drone cannot communicate with the GCS over LTE networks, so it fails to operate the mission in the affected area [17].

Down Grade Attack Using IMSI If a malicious adversary steals IMSI of a target drone and launches a DoS attack at the stage of sending an Attach Request, the drone cannot use the LTE network. In such a situation, the drone uses 3G networks as an alternative. Then, the adversary can use the 3G network's weakness to attack the drone[7, 13].

Location Leakage If a malicious attacker launches a DoS Attack toward a drone to cause the drone to use a 3G network rather than an LTE network and has the IMSI of the drone, the adversary can reveal the location of the target drone running within a radius of 2 km² [13, 20].

4 Proposed Protocol

NIST proposed methods to mitigate various threats in LTE networks [7]. To prevent a hacker disguised as a rogue base station, wiretapping, or downgrade attack, NIST identified third party over-the-top solutions with the mitigation method.

Therefore, we propose an authentication protocol that uses the trusted 3rd party (GCS), subscriber information (signature value), and location information as authentication factors. We propose a secure authentication protocol that we protect the confidentiality of IMSI by encrypting with asymmetric keys of GCS and HSS at first attach request phase.

In this paper, we propose an authentication protocol with the goal of not leaking IMSI by using the EPS-AKA Algorithm of USIM and the existing infrastructure without modification.

4.1 *Architecture of Proposed Protocol*

When buying a SIM and a cellular plan, the drone provides IMSI, LTE K of the SIM, and user information to HSS. The HSS stores the received IMSI in the form of $h(i)$ using a hash function h and ISMS i , and stores the subscriber information in AuC as well.

The secret key for MAVLink protocol should be created on a GCS and shared with drones via secure channels (e.g., local USB cable or local wired Ethernet cable). GCS receives IMSI of the drone using this must-connect secure channel.

Once GCS gets IMSI from the drone through the wired secure channel, the GCS encrypts the $h(i)$ and the current location information (GPS) with the drone user's private key and HSS's public key to request authentication.

After receiving the $h(i)$ and GCS encrypted by the subscriber, the HSS decrypts the cyphertext with HSS's private key and subscriber's private key and checks whether the subscriber is valid or not. If the requester is a valid subscriber, the HSS creates an Authentication Vectors (RAND, AUTN, XRES, K_{asme}) using the EPS-AKA algorithm shared with the subscriber.

After the HSS select MME near drones based on the GPS information, HSS sends the AVs (XRES, K_{asme}) and the temporary IMSI (TMSI) to the MME and addresses the AVs (RAND, AUTN) and TMSI to the subscriber (GCS). The drone receives AVs (RAND, AUTN) and TMSI from the GCS via a secure channel.

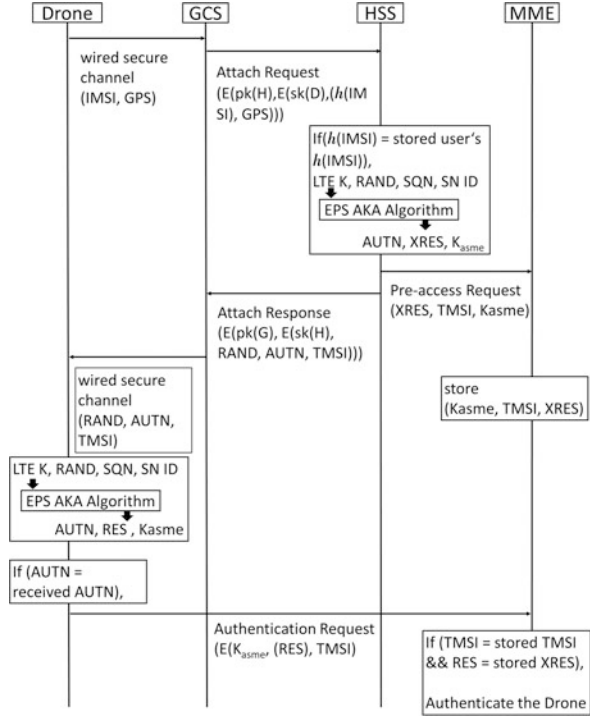
The drone puts the AVs (RAND), IMSI, Key into the EPS-AKA algorithm in SIM, acquires AUTN, RES, K_{asme} , and compares the acquired AUTN and received AUTN from HSS.

If the acquired AUTN and received AUTN are the same, the drone considers that HSS sent the TMSI, and K_{asme} is the encryption key for the LTE network.

Since the MME has been holding the AVs (XRES, K_{asme}) received from the HSS, once receiving TMSI and AVs (RES) encrypted with K_{asme} from the Drone, the MME can decrypt the cyphertext. After comparing it with the RES and XRES, the MME authenticates the drone if the values are the same.

Finally, LTE drone completes the authentication procedure, and a data link between the Drone and GCS over the LTE network is connected.

Fig. 4 LTE drone authentication phase



4.2 Phase of Proposed Protocol

The proposed protocol consists of the authentication phase between four components. We depict the authentication phases in Fig. 4, and list the used notations in Table 1.

4.3 Security Analysis of Proposed Protocol

The proposed protocol is secure against the leakage of IMSI, so it protects the drone from the adversary attacks related to the vulnerability.

Security Against Disclosure of Drone Identity In the proposed protocol, a drone transmits IMSI to its GCS over a secure wired channel, which is USB via so it is impossible to leak IMSI at this stage. The GCS encrypts IMSI with its private key, then encrypts it with HSS's public key and sends it to the HSS. Even if a malicious adversary sniffs the LTE air interface, he cannot obtain the IMSI.

Security Against MITM Attack at the Authentication Phase In our proposed protocol, a GCS sends an authentication request for the drone directly to the HSS

Table 1 Notations of the proposed protocol

Parameter	Meaning
Drone	LTE device (user)
GCS	Ground control station (command and control the drone)
HSS	Home subscriber server
MME	Mobility management entity
IMSI (TMSI)	Drone USIM number (temporarily ID)
GPS	Current location of drone
K_{asme}	Secret key for communication between drone, MME and HSS
RAND	Random number
$h(.)$	Hash function
$pk(X)$	Public key of X
$sk(X)$	Private (secret) key of X
$E(a,b)$	Encrypt message 'b' with a key 'a'
XRES	Expected response
RES	Response for authentication
AUTN	Authentication token

without going through the eNodeB and MME. Therefore, since the GCS sends IMSI as an encrypted message directly to the HSS, even if a malicious adversary masquerades as a rogue eNodeB, the adversary cannot receive the authentication request and obtain the IMSI.

Security Against Denial of Service and Down Grade Attack Using IMSI As the proposed protocol is secure against MITM attacks at the authentication request phase, the malicious attacker cannot interrupt the authentication messages between HSS and GCS. After successful drone authentication, the HSS allocates TMSI instead of IMSI and provides the TMSI to the GCS, so the malicious adversary cannot do DoS attacks using IMSI. Since TMSI is periodically changed, MITM attacks or downgrade attacks using them would be troublesome to the malevolent adversary.

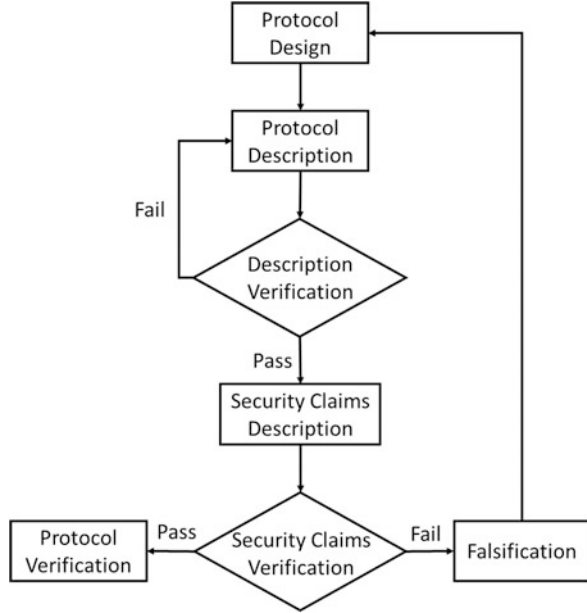
Security Against Location Leakage If a malicious adversary fails to collect IMSIs of drones, the attacker will not be able to detect the location or movement of the drone over time.

5 Formal Analysis

5.1 Protocol Verification Tool : Scyther

Scyther is an automatic tool for the verification, falsification, and analysis of security protocol under the perfect cryptography assumption. We assumed that all

Fig. 5 Protocol verification using Scyther



cryptographic functions are perfect, which means any adversary learns nothing from an encrypted message unless he knows the key. We can use Scyther to find the potential security problems from the protocol development [8, 9].

Figure 5 shows an overview of a protocol verification process using Scyther. We could check the security properties for LTE Drone Authentication and to verify that certain values are confidential (secrecy) or certain properties hold for the communication partners (authentication). Scyther allows us to verify these properties or falsify them.

5.2 Specification

The descriptions of the proposed protocols and the claims are written in Security Protocol Description Language (SPDL), which allows Scyther to verify the properties. The descriptions of the LTE authentication protocol are as follows :

```

const pk : Function;
secret sk : Function;
inversekeys (pk, sk);
hashfunction h;
usertype Message;
protocol LTEDroneAuth(D, G, H, M){
  role D{
    fresh RES : Nonce;
  }
}
  
```

```

    fresh TMSI2 : Message;
    send_4 (D, M, {RES}k(D,H,M),TMSI2); }
role G{
    fresh Imsi, GPS : Nonce;
    var rand, autn : Nonce;
    var TMSI2 : Message;
    send_1 (G, H, {{h(Imsi),GPS}sk(G)}pk(H));
    recv_3 (H, G, {{rand, autn, h(Imsi),
    TMSI2}sk(H)}pk(G)); }
role H{
    var Imsi, GPS : Nonce;
    fresh rand, autn, XRES : Nonce;
    fresh authenticateduser : Nonce;
    fresh TMSI1 : Message;
    fresh TMSI2 : Message;
    recv_1 (G, H, {{h(Imsi),GPS}sk(G)}pk(H));
    match (h(Imsi),h(authenticateduser));
    send_2 (H, M, {XRES, TMSI1,
    k(D,H,M)}k(H,M));
    send_3 (H, G, {{rand, autn, h(Imsi),
    TMSI2}sk(H)}pk(G)); }
role M{
    var XRES, RES : Nonce;
    var TMSI1 : Message;
    var TMSI2 : Message;
    recv_2 (H, M, {XRES, TMSI1, k(D,H,M)}k(H,M));
    recv_4 (D, M, {RES}k(D,H,M), TMSI2);
    match(RES, XRES);
    match(TMSI1, TMSI2); }
}

```

5.3 Analysis of the Verification Results

Cas Cremers[10] and G. Lowe [14] suggest a methodology for the formal specification and verification of abstract security protocols. The author introduces security properties related to secrecy and several forms of authentication. Based on the approach, the security properties that we used to verify that our proposed authentication protocol are as follows:

Secrecy Secrecy property explains that certain information is not revealed to adversaries, even though the message is communicated over untrusted networks. Secrecy claim is written as a claim(A, secret, rt), where A is the role executing this event, and rt is the term that should be secret, which is not known to the adversary.

Aliveness Aliveness is a form of authentication that intends to authorize that an aimed communication partner has executed some events that means the partner is alive. Aliveness claim is written as a $\text{claim}(A, \text{alive})$, where A is the role executing this event and should be alive.

Non-injective Synchronization Non-injective synchronization is that everything we intended to happen in the protocol description also occurs in the trace. Non-injective synchronization claim is written as a $\text{claim}(A, \text{Nisynch})$, where A is the role executing this event.

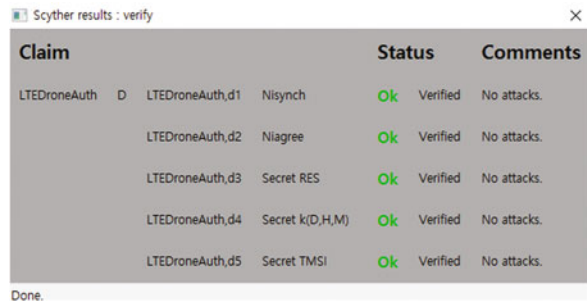
Non-injective Agreement The definition of non-injective agreement expresses that for all claims in any trace, there exist runs for the other roles in the described protocol, such that all communication events causally preceding the claim must have occurred before the claim. Non-injective agreement claim is written as a $\text{claim}(A, \text{Niagree})$, where A is the role executing this event.

Weak Agreement Weak agreement is that if an initiator completes a run of the protocol, apparently with a responder, then the responder has previously been running the protocol, apparently with the initiator. Weak agreement claim is written as a $\text{claim}(A, \text{Weakagree})$, where A is the role executing this event.

We verify that the secrets, which required for authentication, such as IMSI and RES and XRES, are not leaked by using Scyther to verify the proposed authentication protocol. Also, we proved that the proposed protocol satisfies the other security properties. The verification result is as follows :

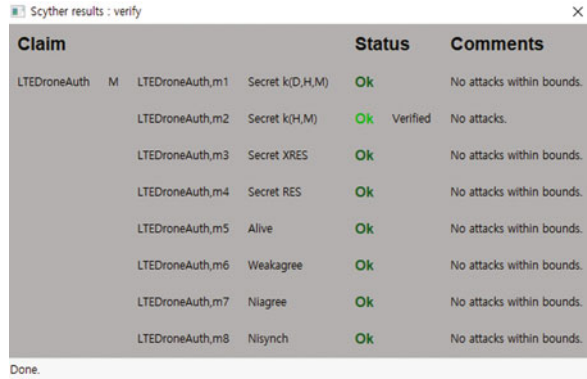
- LTE Drones : We verify that LTE drone satisfies Non-injective synchronization and Non-injective Agreement in the proposed protocol. While transmitting RES from drone to MME, they are secure against malignant adversaries. The verification result is shown in Fig. 6.
- MME : We prove that MME satisfies the security properties, which are Aliveness, Weak agreement, Non-injective synchronization, and Non-injective Agreement within its bound. MME keeps the secret, K_{asme} , and XRES from HSS and RES from LTE drone. It is proved that the proposed mechanism withstands all automatic attacks, and no attack was found within its bounds. The analysis result is shown in Fig. 7.

Fig. 6 Verification result of LTE drone



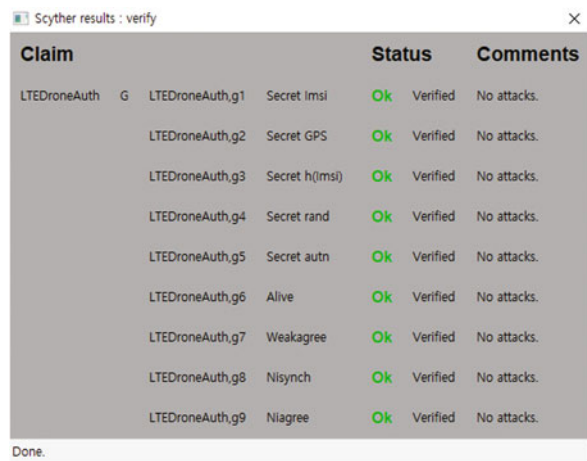
Claim	Status	Comments
LTEDroneAuth, d1 Nisynch	Ok	Verified No attacks.
LTEDroneAuth, d2 Niagree	Ok	Verified No attacks.
LTEDroneAuth, d3 Secret RES	Ok	Verified No attacks.
LTEDroneAuth, d4 Secret $k(D,H,M)$	Ok	Verified No attacks.
LTEDroneAuth, d5 Secret TMSI	Ok	Verified No attacks.

Done.

Fig. 7 Verification result of MME


Claim	Status	Comments
LTEDroneAuth M LTEDroneAuth,m1 Secret $k(D,H,M)$	Ok	No attacks within bounds.
LTEDroneAuth,m2 Secret $k(H,M)$	Ok	Verified No attacks.
LTEDroneAuth,m3 Secret XRES	Ok	No attacks within bounds.
LTEDroneAuth,m4 Secret RES	Ok	No attacks within bounds.
LTEDroneAuth,m5 Alive	Ok	No attacks within bounds.
LTEDroneAuth,m6 Weakagree	Ok	No attacks within bounds.
LTEDroneAuth,m7 Niagree	Ok	No attacks within bounds.
LTEDroneAuth,m8 Nisynch	Ok	No attacks within bounds.

Done.

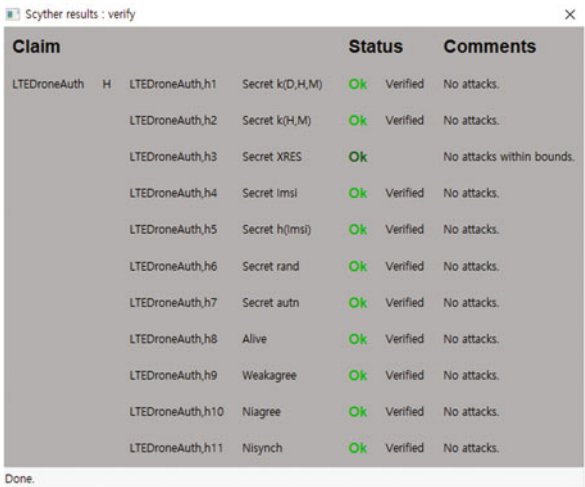
Fig. 8 Verification result of GCS


Claim	Status	Comments
LTEDroneAuth G LTEDroneAuth,g1 Secret lmsi	Ok	Verified No attacks.
LTEDroneAuth,g2 Secret GPS	Ok	Verified No attacks.
LTEDroneAuth,g3 Secret $h(lmsi)$	Ok	Verified No attacks.
LTEDroneAuth,g4 Secret rand	Ok	Verified No attacks.
LTEDroneAuth,g5 Secret autn	Ok	Verified No attacks.
LTEDroneAuth,g6 Alive	Ok	Verified No attacks.
LTEDroneAuth,g7 Weakagree	Ok	Verified No attacks.
LTEDroneAuth,g8 Nisynch	Ok	Verified No attacks.
LTEDroneAuth,g9 Niagree	Ok	Verified No attacks.

Done.

- GCS: We verify that GCS satisfies the security properties, which are Aliveness, Weak agreement, Non-injective synchronization, and Non-injective Agreement. While sending a message to HSS, GCS keeps the secrets, the location information of the drone, and IMSI by hiding with a hash function. While receiving a message from HSS, a malicious adversary cannot capture the secrets, RAND, AUTN, and K_{asme} . The verification result is shown in Fig. 8.
- HSS : We verify that HSS satisfies the security properties, which are Aliveness, Weak agreement, Non-injective synchronization, and Non-injective Agreement. While sending a message to MME, HSS keeps the secret, XRES and K_{asme} . While receiving a message from GCS, a malicious adversary cannot capture the secrets, IMSI, and the location information of the drone. While replying a message to GCS, HSS keeps the secrets, RAND and AUTN. The analysis result is shown in Fig. 9.

Fig. 9 Verification result of HSS



The image shows a screenshot of a Scyther verification window titled "Scyther results : verify". It contains a table with three columns: "Claim", "Status", and "Comments". The table lists 11 claims related to LTE drone authentication, all of which are marked as "Verified" with a green "Ok" status and "No attacks" in the comments. The claims involve various secrets like keys, IMSI, and random numbers, as well as properties like "Alive", "Weakagree", "Niagree", and "Nisynch".

Claim	Status	Comments
LTEDroneAuth, h1 Secret $k(D,H,M)$	Ok Verified	No attacks.
LTEDroneAuth, h2 Secret $k(H,M)$	Ok Verified	No attacks.
LTEDroneAuth, h3 Secret XRES	Ok	No attacks within bounds.
LTEDroneAuth, h4 Secret imsi	Ok Verified	No attacks.
LTEDroneAuth, h5 Secret $h(imsi)$	Ok Verified	No attacks.
LTEDroneAuth, h6 Secret rand	Ok Verified	No attacks.
LTEDroneAuth, h7 Secret autn	Ok Verified	No attacks.
LTEDroneAuth, h8 Alive	Ok Verified	No attacks.
LTEDroneAuth, h9 Weakagree	Ok Verified	No attacks.
LTEDroneAuth, h10 Niagree	Ok Verified	No attacks.
LTEDroneAuth, h11 Nisynch	Ok Verified	No attacks.

Done.

As shown in results, the proposed protocol proves a mutual authentication between LTE drone and HSS by using a trusted third party (GCS). We can claim that our proposed protocol is secure against the leakage of IMSI.

6 Conclusion

In this paper, we introduced LTE drones communication architectures, and a general authentication protocol, EPS-AKA protocol in LTE. Since there exist the security flaws in the LTE authentication protocol, it could be vulnerable to the location privacy and security of LTE drones. Once HSS authenticates LTE drones by using general LTE authentication protocol, adversaries could detect IMSI since it was not encrypted.

We proposed a secure authentication protocol that is appropriate for the LTE drone environment consisting of command and control systems such as GCS. The proposed authentication protocol for LTE drones uses signed and encrypted messages by a public key of GCS and a private key of HSS in order to hide the IMSI of drones from the wireless section. As a result, we could address the security and privacy risk of LTE drones by hiding IMSI.

Besides, we proved the formal analysis of the new proposed protocol against the frequent attacks and automated adversaries. The proposed authentication protocol has been proved secure by using Scyther, and we verified the protocol all the security property requirements specified in the abstract model.

We believe that this work represents a significant step toward secure LTE drones. In our future work, we will evaluate and discuss the performance of the proposed protocol.

Acknowledgments This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00532, Development of High-Assurance (\geq EAL6) Secure Microkernel).

References

1. Overview · mavlink developer guide (nd). <https://mavlink.io/en/protocol/overview.html>
2. 3GPP TS 23.003: Technical Specification Group Core Network and Terminals; Numbering, addressing and identification (Release 16) (2020). <http://www.3gpp.org/dynareport/23003.htm>
3. 3GPP TS 23.401: Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 16) (2020). <http://www.3gpp.org/dynareport/23401.htm>
4. 3GPP TS 23.402: Technical Specification Group Services and System Aspects; Architecture enhancements for non-3GPP accesses (Release 16) (2019). <http://www.3gpp.org/dynareport/23402.htm>
5. M.A. Abdrabou, A.D.E. Elbayoumy, E.A. El-Wanis, LTE authentication protocol (EPS-AKA) weaknesses solution, in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)* (IEEE, New York, 2015), pp. 434–441
6. J. Cao, M. Ma, H. Li, Y. Zhang, Z. Luo, A survey on security aspects for LTE and LTE-A networks. *IEEE Commun Surv. Tutor.* **16**(1), 283–302 (2013)
7. J. Cichonski, J. Franklin, M. Bartock, Guide to LTE security. Tech. rep., National Institute of Standards and Technology (2016)
8. C.J. Cremers, The scyther tool: verification, falsification, and analysis of security protocols, in *International Conference on Computer Aided Verification* (Springer, New York, 2008), pp. 414–418
9. C. Cremers, Scyther user manual. Department of Computer Science, University of Oxford, Oxford (2014)
10. C. Cremers, S. Mauw, Operational semantics, in *Operational Semantics and Verification of Security Protocols* (Springer, New York, 2012), pp. 13–35
11. GSMA: The mobile economy (2020). <https://www.gsma.com/mobileeconomy/>
12. W.B. Hsieh, J.S. Leu, Design of a time and location based one-time password authentication scheme, in *2011 7th International Wireless Communications and Mobile Computing Conference* (IEEE, New York, 2011), pp. 201–206
13. D.F. Kune, J. Koelndorfer, N. Hopper, Y. Kim, Location leaks on the GSM air interface. ISOC NDSS (Feb 2012) (2012)
14. G. Lowe, A hierarchy of authentication specifications, in *Proceedings 10th Computer Security Foundations Workshop* (IEEE, New York, 1997), pp. 31–43
15. M. McFarland, Amazon makes its first drone delivery in the U.K (2016). <https://money.cnn.com/2016/12/14/technology/amazon-drone-delivery/index.html>
16. S.L. McFarland, T.D. Biddle, ISIS-international review devoted to the history of science and its cultural influence, in *America's Pursuit of Precision Bombing, 1910–1945*, vol. 87(2) (Smithsonian Institution Press, Washington, 1996), pp. 389–389
17. S.F. Mjølunes, R.F. Olimid, Easy 4G/LTE IMSI catchers for non-programmers, in *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security* (Springer, New York, 2017), pp. 235–246
18. H.C. Nguyen, R. Amorim, J. Wigard, I.Z. Kovacs, P. Mogensen, Using LTE networks for UAV command and control link: a rural-area coverage analysis, in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)* (IEEE, New York, 2017), pp. 1–6
19. K. Norrman, M. Näslund, E. Dubrova, Protecting IMSI and user privacy in 5G networks, in *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications* (2016), pp. 159–166

20. A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, J.P. Seifert, Practical attacks against privacy and availability in 4G/LTE mobile communication systems (2015). Preprint. arXiv:1510.07563
21. F. Van Den Broek, R. Verdult, J. de Ruiter, Defeating IMSI catchers, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 340–351
22. G. Wang, B. Lee, K. Lim, J. Ahn, Technical trends on security of control and non-payload communications network for unmanned aircraft systems. *Electron. Telecommun. Trends* **32**(1), 82–92 (2017)

Memorable Password Generation with AES in ECB Mode



Timothy Hoang and Pablo Rivas

1 Introduction

Password security is a constant bother in the modern world. Today, it is possible to save passwords into Google's autofill and other password manager programs. These programs can generate passwords on the fly and store them on a computer or online account so that one would only have to remember a single password to access every other password they have. This is useful for people to have a multitude of different secure passwords for all of their accounts to make it more difficult for malicious entities to access them all. However, the downside of this technology is that all of the accounts are linked to one crucial location. Another downside is needing to first log into the master account if signing onto one of the accounts from another location, making access to the desired account difficult. The last downside is that many of these programs charge a subscription fee to securely store passwords. With ALP program, it is possible to create multiple randomly generated passwords that should be easy to remember given the user's parameters. This solves the one location and access from other computers issues since one can more easily remember each individual password since they are readable. Since the password will be readable, and therefore more memorable, one will have an easier time accessing their accounts from other locations without access to the password manager.

T. Hoang (✉) · P. Rivas
Marist College, Poughkeepsie, NY, USA
e-mail: timothy.hoang1@marist.edu

2 Concerns

There is a concern for dictionary attacks with normal random word combinations for password generation [1]. That is why the lexicon was made completely customizable, as it will negate the effect of this type of brute force attack. With the customizable lexicon, the user is not limited to traditional words found in the dictionary. The user is able to add whatever string they deem memorable. This includes slang, names, made-up words, non-English words, sentimental numbers, and any bit of information. This is useful as it allows one to create passwords that are equally complex as a completely random string of the same length, while also being secure from traditional dictionary attacks since a stylistic touch is added to the lexicon. Therefore, unless the attackers know exactly what “words” were in the dictionary at the time of creating the password, they cannot perform a dictionary attack. This makes the customizable lexicon approach protected against brute force dictionary attacks.

In the lexicon, for example, the user puts in these strings, “4Plakilt3, WaR75atel8, M1zule, Laz4Apt, M0der0ck.” To the user, these made up words are memorable for some reason (the numbers, capitalization, and segments of the words hold some significance to that specific user and no one else). Now, say the user requests that the password be 16 characters in length. From those words, the program can combine them to generate a password such as M1zuleWaR75atel8 or 4Plakilt3Laz4Apt. As long as the user has enough words and variety in word lengths within the dictionary, it is possible to create many completely unique passwords at any length. Since the “words” in the dictionary are deemed memorable according to the user, what may look like a bunch of random characters to someone can be easily remembered by the user. Unless the attacker knows every word within the lexicon that the user utilized at the time of generating their password, the only way to brute force it would be to brute force every single character. This gives the same time complexity as if they were to brute force a password where every character is random.

3 Methodology/Experimental Setup

The lexicon reading and password generation portion of the ALP application is complete. The function is for the program to read a lexicon that the user creates. From the lexicon of words, a random “readable” password will be generated. Each word in the .txt file is separated by line. Although any length word can be added to the lexicon, the program, as it stands, will only choose words that are 16 characters or less since that is what is required for the AES 128. This character count can be expanded easily in the code, but for the purpose of demonstrating its use in an AES 128 cipher, it was limited to 16. If the smallest word in the lexicon is too big to fill in the remaining characters needed, randomly generated characters will be chosen for the remainder of the password. In addition to this, the user is able to toggle

between a “readable” and a completely random password. In the completely random passwords, each character is randomly generated and not read from the lexicon at all. These programs are only intended for English letters and numbers, so it may produce problems when introduced to other characters.

For the decryption portion, inverse programs were created for the ShiftRow(), NibbleSub(), and MixColumn() functions that are present in the AESencryption.java file which encrypts messages with AES 128. The changes to these functions include the change present in InvMixColumn(), where there is Galois multiplication in different fields. In addition to these inverse functions, the order of key operations has been reversed.

Figure 1 contains the order of operations to be done for encryption and decryption in AES. Since the encryption part was completed previously, here is an explanation of the right side of the picture from the bottom up. The program starts by adding the round key. Then, for the next nine keys, the following will take place: invShiftRow(), invNibbleSub(), add the round key, then InvMixColumn(). For the last key, the program will do the same process except it will leave out the InvMixColumn()

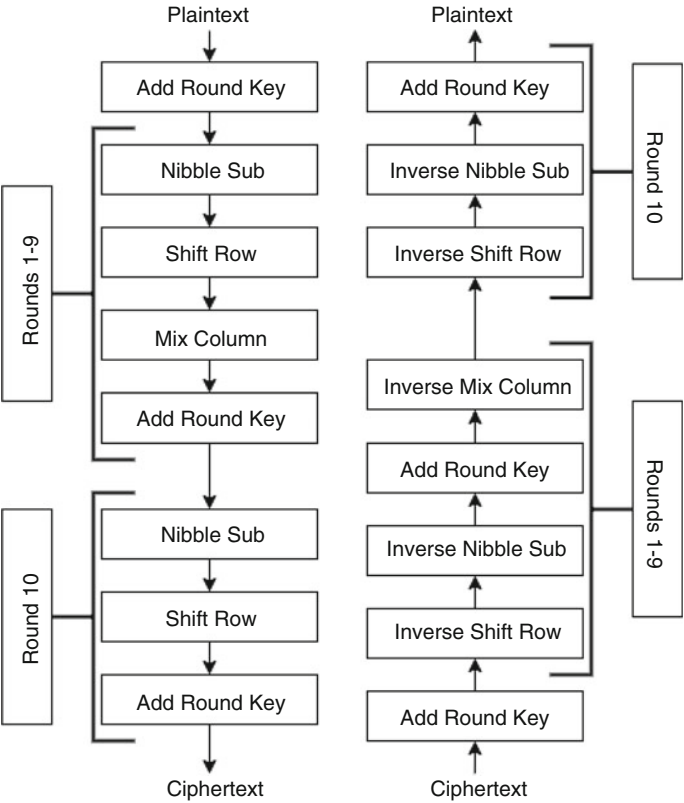


Fig. 1 Order of operations for AES encryption and decryption

function. The order of operations is achieved in the AESdecrypt() function, which is located in the AESdecipher.java file [2].

In InvShiftRow(), all the program does is change the rows opposite to how it was shifted in ShiftRow(). That is, instead of taking the last one, two, and three subjects of the bottom three rows in the matrix and bringing them to the front, the program takes the first three, two, and one subjects from those rows and moves them to the back. This is displayed visually in Fig. 2 [3].

For InvNibbleSub(), the program does the exact same operation as AESNibbleSub() but uses the substitution values contained in the table shown in Fig. 3 [4].

For InvMixColumn(), the program will perform operations as shown in Fig. 4, where the numbers in the second matrix are the Galois field numbers [5].

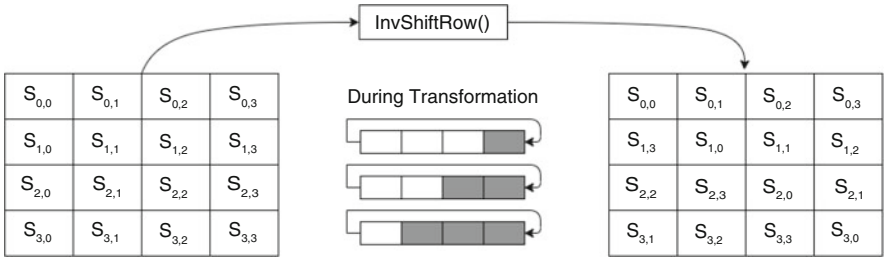


Fig. 2 Inverse Shift Row function

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	9e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f5	64	85	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	5b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	5e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Fig. 3 Inverse Nibble Substitution function substitution values

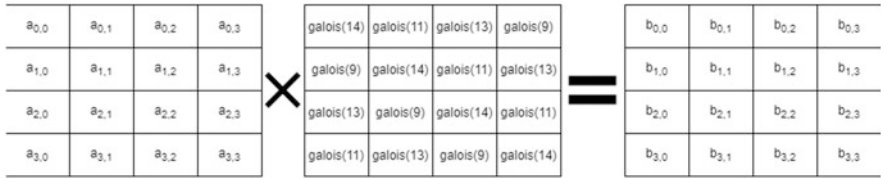


Fig. 4 Inverse Mix Column function

4 Experiment Results

The program will produce a memorable password using words provided in the lexicon of choice. By running the main program, it has produced passwords such as hic3cco7dinh14mb and contr2buto9hedrd. These passwords are concatenations of the strings [“hic”, “3cco7dinh”, “14mb”] and [“contr2buto9”, “hedrd”], respectively. The test cases for each of the inverse functions can be performed by running test.java program. test.java will display the matrices passed into each of the functions as well as their resulting matrices. The output for these test cases can be found in the testcases.txt file located within the data folder of this project. All of the inverse functions perform as intended. Attempting to run the full decryption process does not decrypt the message properly. Despite the fact that several experiments were conducted to address this, the source of this issue has not been found.

5 Conclusion

In conclusion, the ALP program is able to create secure, memorable passwords for use with the lexicon of choice. However, the decryption process of the program is still being investigated. Work has been made on the individual processes and order methodology functioning properly as far as the test cases and results show, but there is something holding back the full decryption. The password generation portion of ALP has many use cases for individuals of all kinds, as everyone needs proper security for their many accounts in the modern world. For further improvement, figuring out and fixing the error concerning the order of operations in the AES decryption process is the most crucial. In addition to this, adding the ability to have special characters in the lexicon for password generation and being able to set requirements for the resulting password, such as requiring a certain number of numbers, special characters, and capitalization can be done.

The code to reproduce the experiments can be accessed under the MIT license in this repository: github.com/timhoangt/ALP

Acknowledgments This work was supported in part by the New York State Cloud Computing and Analytics Center, and by the VPAA’s office at Marist College.

References

1. L. Bošnjak, J. Sres, B. Brumen, Brute-force and dictionary attack on hashed real-world passwords, 1161–1166 (2018). <https://doi.org/10.23919/MIPRO.2018.8400211>
2. A. Wadday, S. Wadi, H. Mohammed, A. Abdullah, Study of WiMAX based communication channel effects on the ciphered image using MAES algorithm. *Int. J. Appl. Eng. Res.* **13**, 6009–6018 (2018)
3. B. Bhattarai, N.K. Giri, FPGA Prototyping of the secured biometric based Identification system. (2015). <https://doi.org/10.13140/RG.2.1.4067.2729>
4. G. Selimis, A. Kakarountas, A. Fournaris, A. Milidonis, O. Koufopavlou, A low power design for Sbox cryptographic primitive of advanced encryption standard for mobile end-users. *J. Low Power Electron.* **3**, 327–336 (2007). <https://doi.org/10.1166/jolpe.2007.139>
5. L.M. Raju, S. Manickam, Secured high throughput of 128-Bit AES algorithm based on interleaving technique. *Int. J. Appl. Eng. Res.* **10**, 11047–11058 (2015)

A Comprehensive Survey on Fingerprint Liveness Detection Algorithms by Database and Scanner Model



Riley Kiefer and Ashokkumar Patel

1 Introduction

Fingerprint biometrics are one of the most popular forms of biometric data for authentication. Many modern cell phones have some form of fingerprint recognition, and fingerprints are commonly used by the police and investigation services. While other biometric data like face recognition is gaining traction, it will take time for it to reach mainstream use. With a growing number of people using biometric authentication, it falls into the hands of researchers and companies to ensure the security of these systems. One of the biggest threats to biometric authentication is the ability for an attacker to spoof the biometric data. For a fingerprint, this is possible by making an artificial fingerprint from the residue on a surface. Some of the most common types of spoofing materials include gelatin, latex, and glue. While researchers have developed algorithms to counter the most common spoofing materials, novel or unseen materials can easily fool an algorithm. Alternatively, cadaver fingers also pose a threat to security. There is a need for robust algorithms to detect fingerprint spoofing of all types. The goal of this research is to compile the latest performance data on software-based anti-spoofing schemes for fingerprints to assist researchers in developing next-generation anti-spoofing measures.

This work is inspired by the E. Marasco and A. Ross' 2014 survey on Anti-spoofing Schemes for Fingerprint Recognition [1]. Their survey details almost all facets of fingerprint biometrics, and it provides a comparison of algorithms published on or before 2014. This survey focuses on a comparison of algorithms published in late 2014 to 2019. This work is an extension and continuation of our original brief survey on Spoofing Detection Systems for Fake Fingerprint

R. Kiefer · A. Patel (✉)
Florida Polytechnic University, Lakeland, FL, USA
e-mail: rkiefer@floridapoly.edu; apatel@floridapoly.edu

Presentation Attacks [2]. This chapter expands upon our original work by including a summary of the LivDet competition results over time, adding algorithm variants, and including algorithms that were tested on non-LivDet datasets.

2 A Brief Review of the LivDet Competition Series

The LivDet competition is held once every 2 years. Competitions have been held in 2009 [3], 2011 [4], 2013 [5], 2015 [6], 2017 [7], and 2019 [8]. At the competition, biometric spoofing data of all types are used to test algorithms submitted by researchers. The fingerprint dataset is divided by images from three or four different scanner models. The primary metric of performance is the Average Classification Error (ACE), which is the average of the Type I and Type II statistical errors. Table 1 includes the various fingerprints scanners used in the LivDet datasets. Most scanners are optical; however, there is a recent shift to thermal images (from Orcanthus) to test algorithm performance on the latest hardware technologies.

Figure 1 shows the improvement of the average overall ACE for LivDet competitors over all competition years. A projection of the trendline shows the possible performance of LivDet-2021; however, this is highly dependent on the data itself. As seen in LivDet-2011, all submitted algorithms had over 20% ACE, alluding to the relative difficulty of the dataset. Figure 2 provides the best ACE metric of all algorithms submitted by competitors, for all scanner types across all competition years. This data is often used as a baseline performance for researchers testing their algorithm.

Table 1 A complete summary of the fingerprint scanner specifications used in the LivDet fingerprint competition series

Company	Years Used	Model	Resolution	Image Size	Format	Scanner Type
Biometrika	09, 11, 13	Fx2000	569	312x372	RAW	Optical
CrossMatch	09	Verfier 300 LC	500	480x640	RAW	Optical
Identix	09	DFR2100	686	720x720	RAW	Optical
Digital Persona	11	4000B	500	355x391	RAW	Optical
Italdata	11, 13	ET10	500	640x480	RAW	Optical
Sagem	11	MSO300	500	352x384	RAW	Optical
CrossMatch	13, 15	L Scan Guardian	500	800x750	-	-
Swipe	13	-	96	208x1500	-	-
Biometrika	15	HiScan-PRO	1000	1000x1000	BMP	Optical
GreenBit	15	DactyScan26	500	500x500	PNG	Optical
Green Bit	17, 19	DactyScan84C	500	500x500	PNG	Optical
Orcanthus	17, 19	Certis2 Image	500	300xn	PNG	Thermal Swipe
Digital Persona	15, 17, 19	U.are.U 5160	500	252x324	PNG	Optical

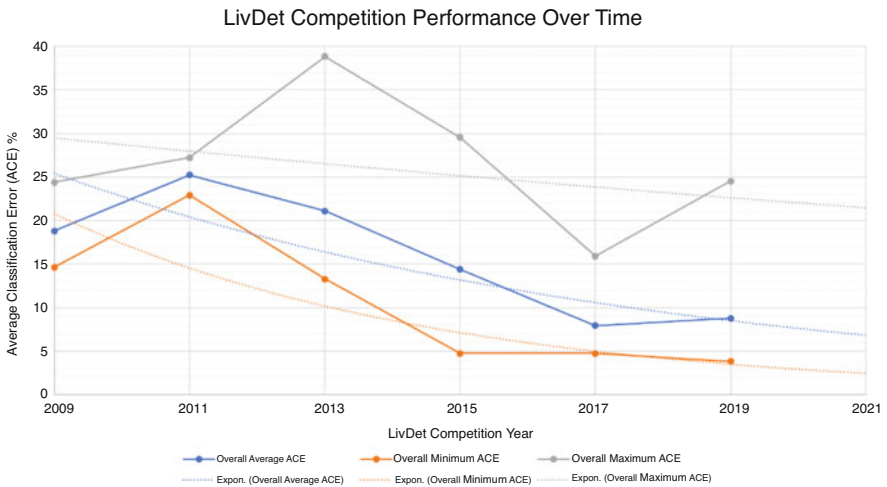


Fig. 1 A graph of the performance of LivDet competitions over time in terms of average, minimum, and maximum overall ACE on all scanners and projections for LivDet-2021

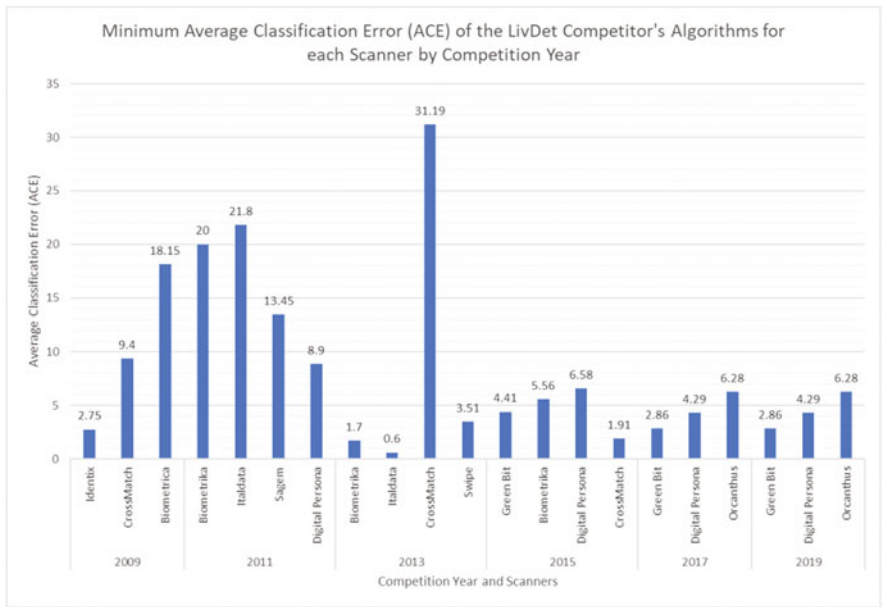


Fig. 2 A graph of the best competitor's algorithms by scanner type for each competition year

3 The LivDet-2009 Competition Dataset

The LivDet-2009 competition dataset [3] includes three scanner models: Biometrika, CrossMatch, and Identix. All fingerprint images were collected in a consensual manner, and there were three spoof materials used: gelatin, silicone, and Play-Doh. Table 2 ranks the algorithms that were submitted between 2014 and 2019 and tested on the LivDet-2009 dataset in terms of the ACE metric.

4 The LivDet-2011 Competition Dataset

The LivDet-2011 competition dataset [4] includes four scanner models: Biometrika, Digital Persona, Italdata, and Sagem. All fingerprint images were collected in a consensual manner and there were six spoof materials used: latex, gelatin, silicone, Play-Doh, wood glue, and Eco-flex. Table 3 ranks the algorithms that were submitted between 2014 and 2019 and tested on the LivDet-2011 dataset in terms of the ACE metric, except for [9], which used the ER metric.

5 The LivDet-2013 Competition Dataset

The LivDet-2013 competition dataset [5] includes four scanner models: Biometrika, CrossMatch, Italdata, and Swipe. Half of the fingerprint images were collected in a consensual manner (CrossMatch and Swipe) and there were six spoof materials used: Body double, latex, Play-Doh, wood glue, gelatin, Eco-Flex, and Mudsill. Table 4 ranks the algorithms that were submitted between 2014 and 2019 and tested on the LivDet-2013 dataset in terms of the ACE metric, except for [9], which used the ER metric.

6 The LivDet-2015 Competition Dataset

The LivDet-2015 competition dataset [6] includes four scanner models: Biometrika, CrossMatch, Digital Persona, and Green Bit. All fingerprint images were collected in a consensual manner and there were ten spoof materials used: Eco-Flex, gelatin, latex, wood glue, liquid Eco-Flex, RTV, Play-Doh, Body Double, OOMOO, and a special gelatin. Table 5 ranks the algorithms that were submitted between 2014 and 2019 and tested on the LivDet-2015 dataset in terms of the ACE metric.

Table 2 A complete summary of all algorithms and their variants ranked by the average ACE of all scanners of the LivDet-2009 dataset, with green representing the best for that scanner

Reference	Algorithm Name and/or Brief Description	Average	Biometrika	Cross Match	Identix
16	DCNN and SVM Trained with RBF Kernel	0	0	-	-
16	DCNN and SVM Trained with Polynomial Kernel Order 2	0.3837	0.3837	-	-
16	DCNN and SVM Trained with Polynomial Kernel Order 3	0.5756	0.5756	-	-
18	MvDA: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	0.73	1.3	0.9	0
18	MvDA: G5- SID RICLBP LCPD DSIFT RILPQ-3, M=4168	0.73	1.2	1	0
22	Deep Triplet Embedding (TNet)	0.77	0.71	1.57	0.044
18	MvDA: G6- SID RICLBP LCPD DSIFT, M=3912	0.8	1.1	1.3	0
18	MvDA: G4- SID RICLBP LCPD DSIFT WLD, M=6793	0.9	1.6	1.1	0
18	MvDA: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	0.93	1.7	1.1	0
18	MvDA: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M= 7826	1	1.9	1.1	0
18	Spidernet: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	1.23	1.6	1.8	0.3
18	Spidernet: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M= 7826	1.23	1.6	1.8	0.3
18	Spidernet: G5- SID RICLBP LCPD DSIFT RILPQ-3, M=4168	1.27	1.4	2	0.4
18	Spidernet: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	1.3	1.5	2.1	0.3
18	Spidernet: G6- SID RICLBP LCPD DSIFT, M=3912	1.3	1.6	2	0.3
18	Spidernet: G4- SID RICLBP LCPD DSIFT WLD, M=6793	1.33	1.9	1.8	0.3
18	AdaBoost: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	1.37	1.7	2.4	0
18	AdaBoost: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	1.37	2.1	2	0
18	Linear SVM: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	1.4	1.6	1.6	1
18	Linear SVM: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M= 7826	1.4	1.7	1.5	1
18	AdaBoost: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M= 7826	1.43	2.2	2.1	0
18	AdaBoost: G4- SID RICLBP LCPD DSIFT WLD, M=6793	1.47	2.3	2.1	0
18	Linear SVM: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	1.47	1.7	1.8	0.9
18	Linear SVM: G4- SID RICLBP LCPD DSIFT WLD, M=6793	1.47	1.8	1.6	1
18	Linear SVM: G5- SID RICLBP LCPD DSIFT RILPQ-3, M=4168	1.5	1.9	1.7	0.9
18	AdaBoost: G6- SID RICLBP LCPD DSIFT, M=3912	1.53	2.3	2.3	0
18	AdaBoost: G5- SID RICLBP LCPD DSIFT RILPQ-3, M=4168	1.57	2.3	2.4	0
18	Linear SVM: G6- SID RICLBP LCPD DSIFT, M=3912	1.57	2.1	1.7	0.9
13	Model 1- CNN-VGG- 227x227	1.63	4.1	0.6	0.2
13	Model 2- CNN-Alexnet- 224x224	2.73	5.6	1.1	0.4
43	Wavelet-Markov	2.83	5.4	2.8	0.3
41	Local Uniform Comparison Image Descriptor (LUCID)	2.86	0.14	7.94	0.49
20	Binarised Statistical Image Features (BSIF)	3.03	3.48	4.6	1.02
11	CNN, Random Sample Patch	3.42	-	-	3.42
16	DCNN and SVM Trained with linear Kernel	3.84	3.84	-	-
13	Model 3- CNN-Random	3.9	9.2	1.7	0.8
38	Augmented Convolutional Network PCA SVM	3.94	9.23	1.78	0.8
29	Local Quality Features (LQF)	5.3	8.5	5.6	2
38	Convolutional Network PCA SVM	5.36	9.49	3.76	2.82
13	Model 4- Local Binary Patterns (LBP)	5.53	10.4	3.6	2.6
38	Augmented Local Binary Patterns (LBP) PCA SVM	5.58	10.44	3.65	2.64
16	DCNN and LR Classifier	8.06	8.06	-	-
42	Image Quality Assessment (IQA)-based	8.23	12.8	10.7	1.2
48	Local Accumulated Smoothing Pattern (LASP)	11.51	9.99	12.28	12.24
19	Local Coherence Patterns and SVM	13.17	13.21	15.58	10.71
38	Local Binary Patterns (LBP) PCA SVM	19.25	50	6.81	0.95

7 The LivDet-2017 Competition Dataset

The LivDet-2017 competition dataset [7] includes three scanner models: Green Bit, Digital Persona, and Orcanthus. All fingerprint images were collected in a consensual manner and there were six spoof materials used: wood glue, Eco-flex, Body Double, gelatin, latex, and liquid Eco-Flex. Due to the LivDet-2017 dataset

Table 3 A complete summary of all algorithms and their variants ranked by the average ACE of all scanners of the LivDet-2011 dataset, with green representing the best for that scanner

Reference	Algorithm Name and/or Brief Description	Average	Biometrika	Digital Persona	Italdata	Sagem
10	LBP and Discrete Shearlet Transform	0	-	0	-	-
26	CNN-MobileNet-v1 and Muniatie-based Local Patches	1.67	1.24	1.61	2.45	1.39
18	Spidernet: G4- SID RICLBP LCDP DSIFT WLD, M=6793	1.68	2	0.7	2.8	1.2
34	Spidernet: G5- SID RICLBP LCDP DSIFT RILPQ-3, M=4168	1.9	1.7	1.3	2.9	1.7
34	Fully Convolutional Network 64 x 64	1.95	1.55	0.8	4.1	1.34
18	Spidernet: G1- SID RICLBP LCDP DSIFT LPQ-3 + LPQ-5, M=4424	1.95	2.3	1.4	2.1	2
18	Spidernet: G2- SID RICLBP LCDP DSIFT WLD LPQ-5 + LPQ-7, M=7304	2	2.4	1	3.1	1.5
18	Spidernet: G3- SID RICLBP LCDP DSIFT WLD BSIF-5, M= 7826	2	2.3	1	2.6	2.1
18	MvDA: G1- SID RICLBP LCDP DSIFT LPQ-3 + LPQ-5, M=4424	2.08	0.7	0.7	4.6	2.3
34	Fully Convolutional Network 48 x 48	2.13	1.1	1.1	4.75	1.56
18	MvDA: G5- SID RICLBP LCDP DSIFT RILPQ-3, M=4168	2.13	0.7	0.8	4.7	2.3
49	Fisher Vector	2.13	3.45	0.2	3.1	1.75
18	Spidernet: G6- SID RICLBP LCDP DSIFT, M=3912	2.15	2.2	1.1	3.1	2.2
18	Linear SVM: G4- SID RICLBP LCDP DSIFT WLD, M=6793	2.25	2	1.3	3.5	2.2
18	MvDA: G2- SID RICLBP LCDP DSIFT WLD LPQ-5 + LPQ-7, M=7304	2.28	0.9	0.7	5.3	2.2
18	MvDA: G6- SID RICLBP LCDP DSIFT, M=3912	2.28	0.9	0.8	4.9	2.5
18	Linear SVM: G2- SID RICLBP LCDP DSIFT WLD LPQ-5 + LPQ-7, M=7304	2.33	2	1.4	3.8	2.1
18	Linear SVM: G3- SID RICLBP LCDP DSIFT WLD BSIF-5, M= 7826	2.33	1.8	1.2	4	2.3
18	Linear SVM: G5- SID RICLBP LCDP DSIFT RILPQ-3, M=4168	2.35	1.5	1.4	4	2.5
18	Linear SVM: G6- SID RICLBP LCDP DSIFT, M=3912	2.38	1.8	1.5	3.8	2.4
18	Linear SVM: G1- SID RICLBP LCDP DSIFT LPQ-3 + LPQ-5, M=4424	2.43	1.8	1.4	4.3	2.2
34	Fully Convolutional Network 32 x 32	2.44	2.35	0.9	5.4	1.09
28	Gram-128 Model	2.45	2.75	0.55	5	1.5
18	MvDA: G4- SID RICLBP LCDP DSIFT WLD, M=6793	2.45	0.5	0.8	5.9	2.6
21	DCNN-Inception v3 + Minutiae-based local patches	2.59	2.6	2.7	3.25	1.8
18	MvDA: G3- SID RICLBP LCDP DSIFT WLD BSIF-5, M= 7826	2.7	1.2	0.8	6.7	2.1
49	Vector Locally Aggregated Descriptors	2.88	3.9	0.1	6.5	1
17	CNN-VGG 227x227 With Dataset Augmentation	2.9	-	-	-	-
22	Deep Triplet Embedding (TNet)	3.33	5.15	1.85	5.1	1.23
18	AdaBoost: G3- SID RICLBP LCDP DSIFT WLD BSIF-5, M= 7826	3.55	3.3	2.9	5.9	2.1
28	Gram-128 Model with Augmentation	3.58	4.95	2	4.8	2.56
18	AdaBoost: G2- SID RICLBP LCDP DSIFT WLD LPQ-5 + LPQ-7, M=7304	3.58	3	3.3	5.8	2.2
18	AdaBoost: G4- SID RICLBP LCDP DSIFT WLD, M=6793	3.58	3.2	3.1	5.8	2.2
18	AdaBoost: G1- SID RICLBP LCDP DSIFT LPQ-3 + LPQ-5, M=4424	3.68	3.1	3.2	5.8	2.6
17	CNN-Alexnet With Dataset Augmentation	3.7	-	-	-	-
18	AdaBoost: G5- SID RICLBP LCDP DSIFT RILPQ-3, M=4168	3.98	3.5	3.5	6	2.9
18	AdaBoost: G6- SID RICLBP LCDP DSIFT, M=3912	4.03	3.7	3.5	6.1	2.8
17	CNN-VGG 227x227 Without Dataset Augmentation	4.2	-	-	-	-
13	Model 1- CNN-VGG- 227x227	4.53	5.2	3.2	8	1.7
17	CNN- Random With Dataset Augmentation	4.7	-	-	-	-
17	CNN-Alexnet Without Dataset Augmentation	5	-	-	-	-
13	Model 2- CNN-Alexnet- 224x224	5.6	5.6	4.6	9.1	3.1
31	Deep Residual Network- ROI	5.65	7.6	2.1	11	2.5
23	Weber Local Binary Descriptor (WLBP)	5.96	5.65	4.1	11.85	2.25
38	Convolutional Network- PCA SVM	6.19	9.9	1.9	5.09	7.86
13	Model 3- CNN-Random	6.4	8.2	3.6	9.2	4.6
31	DCNN with image scale equalization	6.45	9.2	1.35	12.35	2.9
38	Augmented Convolutional Network- PCA SVM	6.45	8.25	3.65	9.27	4.64
31	Deep Residual Network- ROI+LGP	6.68	9.6	1.9	13.5	1.72
49	Bag of Words	6.7	8.15	3.15	11.15	4.35
12	Low Level Features and Shape Analysis: SURF+PHOG+Gabor	6.9	7.89	6.25	8.1	5.36
20	Binarised Statistical Image Features (BSIF)	7.17	6.8	3.55	13.65	4.68
12	Low Level Features and Shape Analysis: SURF+PHOG	7.32	8.76	6.9	7.4	6.23
12	Low Level Features and Shape Analysis: SURF	8.04	9.12	7.95	8.35	6.77
13	Model 4- Local Binary Patterns	8.18	8.8	4.1	12.3	7.5
18	Augmented Local Binary Patterns (LBP) PCA SVM	8.22	8.85	4.15	12.34	7.54
41	Local Uniform Comparison Image Descriptor (LUCID)	8.54	-	-	-	8.54
46	Local Binary Patterns and Principle Component Analysis	8.625	7.1	9.7	10.5	7.2
50	Bayesian Belief Network-MLQC	8.89	9.45	7.1	12.6	6.4
50	Bayesian Belief Network-MLQ	9.09	9.45	7.1	13.4	6.4
17	CNN- Random Without Dataset Augmentation	9.4	-	-	-	-
12	Low Level Features and Shape Analysis: Gabor	9.46	11.21	7.85	12.5	6.28
50	Direct Modelling-Gaussian Mixture Model	9.75	9.5	8.35	13.95	7.2
40	Local Binary Patterns (LBP) with image denoise	10.2	10.2	-	-	-
38	Local Binary Patterns (LBP) PCA SVM	10.32	8.2	3.85	23.68	5.56
39	Pore Characteristics: Fusion	12	18.4	7.8	15.2	6.7
40	LBP without image denoise	12.17	12.17	-	-	-
39	Pore Characteristics: Baseline	12.9	20.6	8.4	14	8.4
50	Bayesian Belief Network-ML	13.5	14.85	9.3	21.1	8.75
50	Spoof Detector	14.08	15	11	21.55	8.75
12	Low Level Features and Shape Analysis: PHOG	17.92	22.45	13.07	20.05	16.1
48	Local Accumulated Smoothing Pattern (LASP)	21.22	22.6	27.1	17.6	17.58
39	Pore Characteristics: Pore Analysis	25.9	26.6	23.4	31.4	22
19	Local Coherence Patterns and SVM	33.21	-	-	-	-
45	Pyramid Histogram of Gradients (PHOG)- QDA	ER=5.0	ER=4.5	ER=5.7	ER=5.4	ER=4.3
45	Pyramid Histogram of Gradients (PHOG)-GMM	ER=5.1	ER=4.8	ER=7.5	ER=4.7	ER=3.4
45	Pyramid Histogram of Gradients (PHOG)-GC	ER=5.1	ER=5.1	ER=6.1	ER=4.6	ER=4.4
45	Local Binary Patterns (LBP)-GMM	ER=5.4	ER=4.4	ER=8.7	ER=4.8	ER=3.6
45	Local Phase Quantization (LPQ)-QDA	ER=6.1	ER=3.8	ER=10.7	ER=3.7	ER=6.1
45	Local Binary Patterns (LBP)-GC	ER=6.2	ER=5.1	ER=9.1	ER=7.8	ER=2.9
45	Local Phase Quantization (LPQ)-GMM	ER=7.0	ER=3.9	ER=11.7	ER=4.6	ER=7.9
45	Local Binary Patterns (LBP)-QDA	ER=7.1	ER=4.2	ER=8.3	ER=12.3	ER=3.4
45	Local Phase Quantization (LPQ)-GC	ER=7.7	ER=3.7	ER=14.9	ER=5.3	ER=6.7

Table 4 A complete summary of all algorithms and their variants ranked by the average ACE of all scanners of the LivDet-2013 dataset, with green representing the best for that scanner

Reference	Algorithm Name and/or Brief Description	Average	Biometrika	CrossMatch	Italdata	Swipe
26	CNN-MobileNet-v1 and Munittae-based Local Patches	0.25	0.2	-	0.3	-
34	Fully Convolutional Network- 32 x 32	0.28	0.15	-	0.4	-
34	Fully Convolutional Network- 48 x 48	0.38	0.35	-	0.4	-
34	Fully Convolutional Network- 64 x 64	0.43	0.2	-	0.65	-
21	DCNN-Inception v3 + Minutiae-based local patches	0.5	0.6	-	0.4	-
22	Deep Triplet Embedding (TNet)	0.57	0.55	-	0.5	0.66
44	Filter: Optimized spoofnet	0.72	0.15	1.77	0.05	0.92
28	Gram-128 Model with Augmentation	0.8	0.7	-	0.9	-
44	Filter: Optimized Architecture Optimization (AO)	1.03	0.7	1.96	0.55	0.92
28	Gram-128 Model	1.05	0.85	-	1.25	-
18	MvDA: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	1.58	0.4	3.9	0.4	1.6
18	MvDA: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M=7826	1.63	0.4	4.7	0.4	1
18	Spidernet: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	1.65	0.7	3.4	0.9	1.6
18	Spidernet: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M=7826	1.68	0.9	3.3	0.7	1.8
18	MvDA: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	1.68	0.5	4.4	0.5	1.3
35	Slim-ResNet- Convolutional Neural Network (thres)	1.74	0.47	-	3.01	-
18	MvDA: G5- SID RICLBP LCPD DSIFT RI LPQ-3, M=4168	1.8	0.5	4.3	0.6	1.8
18	Spidernet: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	1.83	1	3.7	1	1.6
18	Spidernet: G4- SID RICLBP LCPD DSIFT WLD, M=6793	1.83	1.1	3.3	1.1	1.8
18	Spidernet: G5- SID RICLBP LCPD DSIFT RI LPQ-3, M=4168	1.85	1	3.6	0.9	1.9
18	Spidernet: G6- SID RICLBP LCPD DSIFT, M=3912	1.85	1.1	3.5	0.7	2.1
49	Fisher Vector	1.88	1.3	3.7	0.6	1.9
23	Weber Local Binary Descriptor (WLBPD)	1.89	0.4	-	0.95	4.31
20	Binarised Statistical Image Features (BSIF)	1.9	0.55	-	0.55	4.61
18	MvDA: G4- SID RICLBP LCPD DSIFT WLD, M=6793	2	0.5	4.6	0.5	2.4
18	MvDA: G6- SID RICLBP LCPD DSIFT, M=3912	2.03	0.5	5.2	0.4	2
44	Filter: Optimized cf10-11	2.03	1.5	2.67	2.65	1.3
18	Linear SVM: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	2.15	0.7	3.9	1.3	2.7
18	Linear SVM: G5- SID RICLBP LCPD DSIFT RI LPQ-3, M=4168	2.25	0.7	4.4	1.3	2.6
18	Linear SVM: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	2.3	0.7	3.9	1.7	2.9
13	Model 1- Convolutional Neural Network-VGG- 227x227	2.33	1.8	3.4	0.4	3.7
18	Linear SVM: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M=7826	2.33	0.6	4.3	1.7	2.7
18	Linear SVM: G6- SID RICLBP LCPD DSIFT, M=3912	2.35	0.7	4.8	1.3	2.6
18	Linear SVM: G4- SID RICLBP LCPD DSIFT WLD, M=6793	2.5	0.7	4.7	1.8	2.8
12	Low Level Features and Shape Analysis: SURF+PHOG+Gabor	2.61	2.27	2.5	2.17	3.5
49	Vector Locally Aggregated Descriptors	2.68	1.7	4.3	0.7	4
35	Slim-ResNet- Convolutional Neural Network	2.84	0.47	-	5.21	-
13	Model 2- Convolutional Neural Network-Alexnet- 224x224	2.85	1.9	4.7	0.5	4.3
31	Deep Residual Network- ROI+LGP	2.96	-	-	-	-
31	Deep Residual Network- ROI	2.99	-	-	-	-
12	Low Level Features and Shape Analysis: SURF+PHOG	3.27	3.42	2.96	2.85	3.85
18	AdaBoost: G4- SID RICLBP LCPD DSIFT WLD, M=6793	3.3	1	5.6	1.3	5.3
18	AdaBoost: G3- SID RICLBP LCPD DSIFT WLD BSIF-5, M=7826	3.4	1.1	5.6	1.6	5.3
47	GoogLeNet	3.4	3.4	-	-	-
18	AdaBoost: G6- SID RICLBP LCPD DSIFT, M=3912	3.45	1.2	6.3	1.3	5
13	Model 3- Convolutional Neural Network-Random	3.5	0.8	3.2	2.4	7.6
18	AdaBoost: G5- SID RICLBP LCPD DSIFT RI LPQ-3, M=4168	3.53	1.3	6.3	1.4	5.1
38	Augmented Convolutional Network- PCA SVM	3.55	0.8	3.29	2.45	7.67
47	CaffeNet	3.55	3.55	-	-	-
18	AdaBoost: G1- SID RICLBP LCPD DSIFT LPQ-3 + LPQ-5, M=4424	3.6	1	6.4	1.5	5.5
18	AdaBoost: G2- SID RICLBP LCPD DSIFT WLD LPQ-5 + LPQ-7, M=7304	3.65	0.9	5.8	1.5	6.4
12	DCNN with image scale equalization	3.7	4.35	7	1.4	2.05
33	Low Level Features and Shape Analysis: Gabor	3.72	2.7	4.67	4.02	3.5
12	Low Level Features and Shape Analysis: SURF	5.26	5.75	6.08	4.6	4.6
44	Filter: Random Architecture Optimization (AO)	6.26	3.5	7.91	2.55	11.06
12	Siamese	6.95	6.95	-	-	-
17	Low Level Features and Shape Analysis: PHOG	7.24	3.87	9.32	6.7	9.05
49	Bag of Words	7.26	4.95	5.5	12.25	6.35
44	Filter: Random spoofnet	7.42	5.3	12.18	8.95	3.25
35	ResNet- Convolutional Neural Network	10.63	4.09	-	17.16	-
37	Histogram of Invariant Gradients (HIG): Minutiae Circle Combined	12.2	3.9	28.76	1.7	14.44
39	Local Uniform Comparison Image Descriptor (LUCID)	12.24	-	-	-	12.24
39	Pore Characteristics: Pore Analysis	12.7	2.2	34.9	1	-
13	Model 4- Local Binary Patterns	14.18	1.7	49.4	2.3	3.3
38	Augmented Local Binary Patterns- PCA SVM	14.2	1.7	49.45	2.3	3.34
12	Low Level Features and Shape Analysis: WLD+LPQ	14.31	1.4	45.95	3.4	6.51
39	Pore Characteristics: Baseline	14.4	2.4	38.4	2.5	-
39	Pore Characteristics: Fusion	14.4	2	39.6	1.6	-
37	Histogram of Invariant Gradients (HIG): Dense Block Packing Extended	14.87	10.9	28.76	1.7	18.11
37	Histogram of Invariant Gradients (HIG): Dense Block Packing	15.19	3.9	34.13	8.3	14.44
38	Convolutional Network PCA SVM	15.84	4.55	5.2	47.65	5.97
44	Filter: Random cf10-11	18.85	22.55	16.89	23.55	12.4
37	Histogram of Invariant Gradients (HIG): Minutiae Circle	21.82	4.3	39.96	10.6	32.41
38	Local Binary Patterns- PCA SVM	33.75	25.65	49.87	55.45	4.02
45	Local Binary Patterns-GC	ER = 13.9	ER = 2.4	ER = 1.2	ER = 48.8	ER = 3
45	Pyramid History of Invariant Gradients (PHOG)-GMM	ER = 6	ER = 3.2	ER = 4.6	ER = 11.8	ER = 4.5
45	Pyramid History of Invariant Gradients (PHOG)-GC	ER = 7.2	ER = 3.9	ER = 6.3	ER = 12.1	ER = 6.6
45	Pyramid History of Invariant Gradients (PHOG)- QDA	ER = 7.5	ER = 3.9	ER = 6.4	ER = 12.9	ER = 6.7
45	Local Phase Quantization (LPQ)-GMM	ER = 7.6	ER = 6.8	ER = 5.5	ER = 14.3	ER = 2.6
45	Local Binary Patterns-QDA	ER = 7.8	ER = 2	ER = 1.9	ER = 22.3	ER = 4.9
45	Local Phase Quantization (LPQ)-QDA	ER = 8.2	ER = 7.9	ER = 6.3	ER = 15.1	ER = 3.3
45	Local Phase Quantization (LPQ)-GC	ER = 8.4	ER = 7.7	ER = 7.2	ER = 14.7	ER = 3.8
45	Local Binary Patterns-GMM	ER = 8.5	ER = 1.7	ER = 3.2	ER = 24	ER = 5.1

Table 5 A complete summary of all algorithms and their variants ranked by the average ACE of all scanners of the LivDet-2015 dataset, with green representing the best for that scanner

Reference	Algorithm Name and/or Brief Description	Average	Biometrika	CrossMatch	Digital Persona	GreenBit
26	CNN-MobieNet-v1 and Munitae-based Local Patches	0.97	1.12	0.64	1.48	0.68
34	Fully Convolutional Network- 48 x 48	1.26	0.35	1.09	3.4	0.2
34	Fully Convolutional Network- 32 x 32	1.34	1.25	0.82	3	0.3
21	DCNN-Inception v3 + Minutiae-based local patches	1.39	1.76	0.81	1.08	2
34	Fully Convolutional Network- 64 x 64	2.01	0.6	1.44	5.45	0.55
52	Electrocardiogram Fingerprint (ECGFP)-30	2.6	3.5	-	2.6	1.8
35	Slim-ResNetCNN: Model 2- 7 improved residual block_bs	3.07	2.77	2.82	4.53	2.17
35	Slim-ResNetCNN: Structure 1- padding channel layer stride =1	3.08	2.78	2.82	4.53	2.17
35	Slim-ResNet Convolutional Nueral Network	3.11	2.78	3.03	4.48	2.14
35	Slim-ResNet Convolutional Nueral Network (thres)	3.11	3.1	4.32	2.37	2.64
49	Fisher Vector	3.2	3.2	3.56	4.75	1.3
32	Template Probe-CNN	3.21	2.08	0.44	5.88	3.64
52	Electrocardiogram Fingerprint (ECGFP)-10	3.3	4.3	-	3.4	2.1
28	Gram-128 Model	3.56	4.1	0.27	8.5	1.35
35	Slim-ResNetCNN: Model 1- 4 improved residual block_bs	3.58	4.51	3.12	4.88	1.8
52	Electrocardiogram Fingerprint (ECGFP)-7.5	3.7	4.7	-	4.2	2.2
35	Slim-ResNetCNN: Structure 3- 1x1 convolution layer stride =1	3.78	4.23	3.65	4.81	2.41
35	Slim-ResNetCNN: Model 3- 10 improved residual block_bs	3.79	3.8	4.1	4.53	2.71
32	Liveness Map-CNN	4.13	3.28	0.85	8.08	3.04
28	Gram-128 Model with Augmentation	4.15	3.75	3.4	7	2.46
49	Vector Locally Aggregated Descriptors	4.16	4.2	4.85	5.2	2.4
52	Electrocardiogram Fingerprint (ECGFP)-5	4.2	4.9	-	5	2.6
29	Local Quality Features (LQF)	4.22	4.78	1.93	5.84	4.33
35	Slim-ResNetCNN: Structure 2- padding channel layer stride =2	4.49	3.79	3.51	6.4	4.27
25	Liveness Map-CNNp	4.72	4.2	1.4	9.5	3.8
31	Deep Residual Network- ROI	5.32	-	-	-	-
31	Deep Residual Network- ROI+LGP	5.86	-	-	-	-
23	Weber Local Binary Descriptor (WLBPD)	9.6775	9.64	10.82	13.72	4.53
49	Bag of Words	10.67	11.15	10.38	14.1	7.05
35	ResNet Convolutional Nueral Network	24.25	32.06	9.59	40.61	14.74

Table 6 A complete summary of all algorithms and their variants ranked by the average ACE of all scanners of the LivDet-2017 dataset, with green representing the best for that scanner

Reference	Algorithm	Average	Green Bit	Digital Persona	Orcanthus
35	Slim-ResNetCNN	1.01	0.48	0.97	1.57
57	Spoof Buster with UMG Wrapper	4.12	2.58	4.8	4.99
56	Spoof Buster	4.56	3.32	4.88	5.49

recently opening to the public, there are several publications that have tested on this dataset as shown in Table 6.

8 Traditional Machine Learning Algorithms

Several researchers used traditional machine learning algorithms with modified parameters. Research on Fractional Energy of Cosine Transformed Fingerprint Images [10] showed incredible success on the FVC2000 and ATVS dataset with correctly tuned hyperparameters of size and age. Other research on Directional Ridge Frequency [11] showed the importance of using a combination of horizontal, vertical, and diagonal ridge orientations, compared to using them separately. Table 7 ranks the best traditional machine learning algorithms variations on the ATVS and

Table 7 A summary of traditional machine learning algorithms and their variants ranked by average Accuracy Rate on the FVC2000 and ATVS dataset, with green representing the best for that dataset

Reference	Algorithm/Classifier	Average	ATVS AR (%)	FVC2000 AR (%)
53	MLP-- CTFC- Size-8, Age 0.097%	98.41	97.12	99.69
53	Random Forest- CTFC- Size-8, Age 0.097%	98.28	97.49	99.06
53	MLP-- CTFC- Size-4, Age 0.024%	97.76	96.51	99
53	Random Forest- CTFC- Size-4, Age 0.024%	97.64	95.89	99.38
53	MLP-- CTFC- Size-16, Age 0.39%	96.18	92.72	99.64
53	Random Forest- CTFC- Size-16, Age 0.39%	96.1	94.35	97.84
53	Random Forest- CTFC- Size-2, Age 0.006%	95.77	94.35	97.19
53	J48- CTFC- Size-4, Age 0.024%	95.63	93.44	97.81
53	Random Forest- CTFC- Size-32, Age 1.56%	95.1	92.85	97.34
53	SVM- CTFC- Size-8, Age 0.097%	95.01	90.32	99.69
53	J48- CTFC- Size-8, Age 0.097%	94.25	91.3	97.19
53	Naïve Bayes- CTFC- Size-8, Age 0.097%	93.43	88.11	98.75
53	SVM- CTFC- Size-16, Age 0.39%	93.29	87.58	99
24	Random Forest- All Fusion/Combination	93.1	94.86	91.33
53	MLP-- CTFC- Size-2, Age 0.006%	93.07	86.58	99.56
24	SVM-All Fusion/Combination	92.54	93.87	91.21
53	J48- CTFC- Size-16, Age 0.39%	91.41	85.7	97.12
53	J48- CTFC- Size-32, Age 1.56%	91.35	85.57	97.12
53	J48- CTFC- Size-2, Age 0.006%	91.13	85.7	96.56
53	SVM- CTFC- Size-32, Age 1.56%	91.11	82.94	99.28
53	SVM- CTFC- Size-4, Age 0.024%	90.9	82.11	99.69
53	Naïve Bayes- CTFC- Size-4, Age 0.024%	90.84	82.29	99.38
53	MLP-- CTFC- Size-32, Age 1.56%	90.78	82.55	99
24	J48- All Fusion/Combination	90.12	89.13	91.1
53	Naïve Bayes- CTFC- Size-2, Age 0.006%	89.73	88.83	90.63
53	Naïve Bayes- CTFC- Size-16, Age 0.39%	88.66	88.83	88.49
53	Naïve Bayes- CTFC- Size-32, Age 1.56%	88.65	88.58	88.71
24	Random Forest- Diagonal Ridges	88.59	88.43	88.75
24	MLP- All Fusion/Combination	87.56	89.42	85.7
53	Naïve Bayes- CTFC- Size-64, Age 6.25%	87.32	87.82	86.82
53	Random Forest- CTFC- Size-64, Age 6.25%	87.2	87.2	87.2
53	Random Forest- CTFC- Size-128, Age 25%	87.03	87.08	86.98
53	Random Forest- CTFC- Size-256, Age 100%	86.28	86.28	86.28
24	Random Forest- Horivertical Ridges	86.19	78.52	93.86
53	Naïve Bayes- CTFC- Size-128, Age 25%	86.13	87.13	85.13
53	SVM- CTFC- Size-2, Age 0.006%	85.04	87.58	82.5
53	J48- CTFC- Size-64, Age 6.25%	84.94	84.94	84.94
53	J48- CTFC- Size-128, Age 25%	84.15	83.65	84.65
53	Naïve Bayes- CTFC- Size-256, Age 100%	83.82	83.82	83.82
24	SVM- Horivertical Ridges	82.48	79.49	85.46
24	MLP- Diagonal Ridges	81.47	81.65	81.29
24	J48- Horivertical Ridges	80.5	76.76	84.24
53	J48- CTFC- Size-256, Age 100%	80.48	79.48	81.48
53	MLP-- CTFC- Size-64, Age 6.25%	78.54	78.54	78.54
53	SVM- CTFC- Size-64, Age 6.25%	76.58	76.78	76.38
24	J48- Diagonal Ridges	75.99	78.21	73.76
53	SVM- CTFC- Size-128, Age 25%	75.09	74.49	75.69
24	MLP- Horivertical Ridges	74.99	78.52	71.46
53	SVM- CTFC- Size-256, Age 100%	74.04	73.89	74.19
24	SVM- Diagonal Ridges	72.94	82.27	63.6
53	MLP-- CTFC- Size-128, Age 25%	71.61	70.61	72.61
24	Naïve Bayes-All Fusion/Combination	71.1	56.42	85.78
53	MLP- CTFC- Size-256, Age 100%	70.52	70.02	71.02
24	Naïve Bayes- Diagonal Ridges	67.68	60.27	75.09
24	Naïve Bayes- Horivertical Ridges	53.1	53.71	52.48

FVC2000 dataset using the average of the Accuracy Rate (AR) performance metric for each scanner.

9 Performance on Other Datasets

While the LivDet dataset is the most popular dataset to test a liveness detection algorithm, other datasets still play an important role in testing an algorithm's robustness. The datasets tested in this section include MSU-FPAD, ATVS, PBSKD, and a custom dataset.

9.1 Performance on ATVS Data by Scanner Type (Capacitive, Optical, and Thermal)

The ATVS dataset is unique because of the variety of scanner types. With a capacitive, optical, and thermal sensor, algorithm cross-sensor robustness is easily identifiable if the performance metrics are relatively similar. Table 8 summarizes several publications that tested all three types of scanners using the Error Rate metric. As illustrated by [12], the method of a Gaussian Filter with pore extraction has a strong optical sensor performance, but a relatively low capacitive performance, which shows improvements must be made on cross-sensor algorithm robustness.

9.2 Performance on Miscellaneous Datasets Using ACE, FAR, and FRR

Table 9 shows the performance of various algorithms on the various datasets by the ACE metric. The false acceptance rate (FAR) and false rejection rate (FRR) are included if available. It is important to note that some of the datasets in this table are private, such as the MSU-FPAD and the custom dataset of [13]. While performance tests on private datasets provide a quantifiable metric of how the well the algorithm

Table 8 A ranked summary of algorithm performance on the ATVS dataset by scanner type in terms of Error Rate

Reference	Algorithm	Database	All Average	Capacitive Sensor Error Rate (%)	Optical Sensor Error Rate (%)	Thermal Sensor Error Rate (%)
9	Multi-Scale Center Symmetric Local Binary Patterns 1 (MSLBP-1)	ATVS	4.13	4.1	5.2	3.1
9	Multi-Scale Center Symmetric Local Binary Patterns 2 (MSLBP-2)	ATVS	5.13	5.2	6.1	4.1
9	First Proposed Method (MS-CS-LBP)	ATVS	3.43	3.1	4.1	3.1
9	Second Proposed Method (CS_LBP and MSLBP-2)	ATVS	4.77	4.1	6.1	4.1
19	Local Coherence Patterns and SVM	ATVS	6.51	9.05	3.58	6.9
51	Gaussian filter and extracted pores	ATVS-Ffp	7.62	13.03	2.05	7.79

Table 9 A summary of algorithm performance ranked by ACE on various datasets

Reference	Brief Algorithm Description or Name	Database and or Sensor	ACE (%)	FAR (%)	FRR (%)
26	CNN-MobieNet-v1 and Munitae-based Local Patches	MSU-FPAD- CrossMatch Guardian 200	0.11	0.11	0.1
54	Security Level=High	ATVS- FFp- All Sesnors	0.25	-	-
26	CNN-MobieNet-v1 and Munitae-based Local Patches	MSU-FPAD- CrossMatch Guardian 200	0.5	0	1
26	CNN-MobieNet-v1 and Munitae-based Local Patches	PBSKD- CrossMatch Guardian 200	0.67	0.33	1
55	ANN with texture descriptors- 4 Principal Components	Custom Dataset	0.74	0	1.47
26	CNN-MobieNet-v1 and Munitae-based Local Patches	PBSKD- CrossMatch Guardian 200	0.83	0.65	1
54	Security Level=Medium	ATVS- FFp- All Sesnors	0.98	-	-
26	CNN-MobieNet-v1 and Munitae-based Local Patches	MSU-FPAD- Lumidigm Venus 302	1.15	1.3	1
26	CNN-MobieNet-v1 and Munitae-based Local Patches	PBSKD- Lumidigm Venus 302	1.97	3.84	0.1
54	Pattern of Oreitned Edge Magnitudes (POEM)	ATVS- FFp- All Sesnors	2.2	-	-
55	ANN with texture descriptors- 2 Principal Components	Custom Dataset	2.21	2.21	2.21
26	CNN-MobieNet-v1 and Munitae-based Local Patches	PBSKD- CrossMatch Guardian 200	2.71	5.32	0.1
9	Method 1- (MS-CS-LBP)	ATVS- All Sensors	3.43	-	-
54	Census Transform Histogram (CENTRIST)	ATVS- FFp- All Sesnors	3.84	-	-
9	Multi-Scale Center Symmetric Local Binary Patterns 1 (MSLBP-1)	ATVS- All Sensors	4.13	-	-
9	Method 2- (CS_LBP and MSLBP-2)	ATVS- All Sensors	4.77	-	-
26	CNN-MobieNet-v1 and Munitae-based Local Patches	MSU-FPAD- Lumidigm Venus 302	5.07	10.03	0.1
9	Multi-Scale Center Symmetric Local Binary Patterns 2 (MSLBP-2)	ATVS- All Sensors	5.13	-	-
19	Local Coherence Patterns and SVM	ATVS- All Sensors	6.51	-	-
54	Local Uniform Comparison Image Descriptor (LUCID) (SL=Low)	ATVS- FFp- All Sesnors	7.17	-	-
51	Gaussian filter and extracted pores	ATVS-FFp- All Sensors	7.62	-	-

performed, it is difficult to compare the performance of two algorithms on separate datasets.

10 Conclusion

In summary, our survey reviews the Fingerprint LivDet competition’s growth over the years, the many algorithms published between 2014 and 2019 and their performance on the LivDet competition datasets, the performance of traditional machine learning algorithms with varying hyperparameters and inputs, and the performance of published algorithms on non-LivDet competition sets. As expected, the performances observed at the LivDet competitions continue to improve in terms of ACE with more powerful and robust algorithms. However, with more sophisticated datasets and different scanner hardware, it may pose a significant challenge to the algorithm performance. With the numerous published algorithms and their tests on the popular LivDet datasets, it is easy to compare algorithm performance.

With all the data collected in this survey, the state-of-the-art algorithm performance for each scanner type is easily identifiable and accessible for other researchers to study to improve their own model. The LivDet performance data of Tables 2, 3, 4, and 5 also gave some useful insights when plotted on a graph, which could not be included in this chapter due to page constraints. While the data from these graphs need further analysis to gain additional insight, the graph’s scanner-based trendlines show the relative difficulty of each dataset. A scatter plot of the data presented in Table 2 (LivDet-2009) shows that the Identix dataset is typically easier for most algorithms to classify compared to CrossMatch and Biometrika. A

scatter plot of the data presented in Table 3 (LivDet-2011) shows that that algorithm performance on the Digital Persona images typically had a lower ACE compared to Italdataset images, which typically had a much higher ACE compared to the other model's images. A scatter plot of the data presented in Table 4 (LivDet-2013) shows that images from Biometrika and Italdataset typically had the lowest ACE metrics, while CrossMatch images typically had the highest ACE metric. Finally, a scatter plot of the data presented in Table 5 (LivDet-2015) shows that images from GreenBit and CrossMatch typically had a lower ACE metric, while the Digital Persona dataset had a much higher ACE. This data pinpoints the scanner datasets that researchers have mastered, like the LivDet-2009 Identix dataset, while also highlighting the scanner datasets that need additional research, like the LivDet-2013 CrossMatch dataset.

The traditional machine learning algorithms with modified parameters and inputs also provided powerful solutions to anti-spoofing with the correct tuning. The research from [10] reveals that a size parameter of 64–128 and an age parameter of 6.25–25% yields the best results on the FVC2000 dataset across all tested traditional machine learning algorithms. On the ATVS dataset, with a size parameter of 64 and an age parameter of 6.25%, the Random Forest and Naïve Bayes performed slightly worse compared to the FVC2000 dataset, but significantly better than the other tested machine learning algorithms on the ATVS dataset. With the right parameters, the research from [10] shows impressive results on the FVC2000 dataset but will need some tweaking to have an equal performance on the ATVS dataset. Research from [11] also pinpoints which traditional machine learning algorithm works the best with varying ridge orientations. The data provided also highlights the importance of using a combination of ridge angles with the significant increase in performance for most algorithms on both datasets. The miscellaneous datasets discussed at the end of this chapter, while the algorithm comparability is low, still provides a means for additional data to train for model robustness, and it reveals promising models [14, 15] that should also be tested on the LivDet datasets for comparability. Some datasets like the ATVS can provide a powerful measure of cross-scanner-type robustness. By using optical, thermal, and capacitive scanners, performance results can show how dependent an algorithm is on a certain type of input.

With the introduction of more powerful and unique hardware, algorithms need to adapt. While this survey reviews most of the data that compares a model's performance by scanner type, there is still a need to research, survey, and analyze the cross-spoof material data. Attempts have been made to learn the characteristics of varying spoof materials, but with novel spoofing materials research seems to be outpaced. Perhaps, the further development on the One-Class Classifier [16] will combat the rapid number of novel spoof materials or research on Optical Coherence Tomography (OCT) [17] will inhibit the effectiveness of many spoof materials by altering the approach to liveness detection solutions. In conclusion, as the data in this survey suggests, there is a continual need for more advanced generalization in terms of cross-sensor and cross-material robustness to ensure the security of fingerprint biometrics.

References

1. E. Marasco, A. Ross, A survey on anti-spoofing schemes for fingerprint recognition systems. *ACM Comput. Surv.* **47**(2) (2014)., Article 28, 36 Pages
2. R. Kiefer, J. Stevens, A. Patel, M. Patel, A survey on spoofing detection systems for fake fingerprint presentation attacks, in *Fourth International Conference on ICT for Intelligent Systems (ICTIS – 2020)*. Accepted and pending publication
3. G. Marcialis et al., First international fingerprint liveness detection competition—LivDet 2009, Clarkson.edu, 2009
4. D. Yambay, L. Ghiani, P. Denti, G. Marcialis, F. Roli, S. Schuckers, LivDet 2011 – fingerprint liveness detection competition 2011, Clarkson.edu, 2011
5. L. Ghiani et al., LivDet 2013 fingerprint liveness detection competition 2013, in *2013 International Conference on Biometrics (ICB)*, (Madrid, 2013), pp. 1–6
6. V. Mura, L. Ghiani, G. Marcialis, F. Roli, LivDet 2015 fingerprint liveness detection competition 2015, Clarkson.edu, 2015
7. V. Mura et al., arXiv:1803.05210v1 [cs.CV] 14 Mar 2018 LivDet 2017 fingerprint liveness detection competition 2017, Arxiv.org, 2019
8. G. Orru et al., LIVDET inaction- fingerprint liveness detection competition 2019, Arxiv.org, 2019
9. Z. Akhtar, C. Micheloni, G.L. Foresti, Correlation based fingerprint liveness detection, in *2015 International Conference on Biometrics (ICB)*, (Phuket, 2015), pp. 305–310
10. S. Khade, S.D. Thepade, Novel fingerprint liveness detection with fractional energy of cosine transformed fingerprint images and machine learning classifiers, in *2018 IEEE Punecon*, (Pune, India, 2018), pp. 1–7
11. S. Khade, S.D. Thepade, A. Ambedkar, Fingerprint liveness detection using directional ridge frequency with machine learning classifiers, in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, (Pune, India, 2018), pp. 1–5
12. M. Lu, Z. Chen, W. Sheng, A pore-based method for fingerprint liveness detection, in *2015 International Conference on Computer Science and Applications (CSA)*, (Wuhan, 2015), pp. 77–81
13. C. Zaghetto, M. Mendelson, A. Zaghetto, F.D.B. Vidal, Liveness detection on touch-less fingerprint devices using texture descriptors and artificial neural networks, in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, (Denver, CO, 2017), pp. 406–412
14. F. Pala, B. Bhanu, On the accuracy and robustness of deep triplet embedding for fingerprint liveness detection, in *2017 IEEE International Conference on Image Processing (ICIP)*, (Beijing, 2017), pp. 116–120
15. T. Chugh, A.K. Jain, Fingerprint presentation attack detection: Generalization and efficiency, in *ICB*, (2019)
16. J.J. Engelsma, A.K. Jain, Generalizing fingerprint spoof detector: Learning a one-class classifier. arXiv:1901.03918 (2019)
17. T. Chugh, A.K. Jain, OCT fingerprints: Resilience to presentation attacks. arXiv:1908.00102 (2019)

Suitability of Voice Recognition Within the IoT Environment



Salahaldeen Duraibi, Fahad Alqahtani, Frederick Sheldon,
and Wasim Alhamdani

1 Introduction

User authentication refers to the process in which a user submits his/her identity credential (often represented by paired username and password) to an information system in order to validate the person who he/she claims to be. In general, within the context of IoT, three factors of authentication are usually employed: (i) something a user knows (e.g., a password); (ii) something a user has (e.g., a secure token); and (iii) something a user is (e.g., biometric characteristics). Passwords are the most common authentication mechanism (i.e., single factor). However, password- and token-based authentications have many security issues [1, 2] and are not suitable for smart devices because of the unattended nature of IoT smart devices. A biometric

S. Duraibi (✉)

Computer Science Department, University of Idaho, Moscow, ID, USA

Computer Science Department, Jazan University, Jazan, Saudi Arabia

e-mail: dura6540@vandals.uidaho.edu

F. Alqahtani

Computer Science Department, University of Idaho, Moscow, ID, USA

Computer Science Department, Jazan University, Jazan, Saudi Arabia

Computer Science Department, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

e-mail: Alqa0199@vandals.uidaho.edu

F. Sheldon

Computer Science Department, University of Idaho, Moscow, ID, USA

e-mail: sheldon@uidaho.edu

W. Alhamdani

Computer Science Department, University of the Cumberland, Williamsburg, KY, USA

e-mail: wasim.alhamdani@ucmberrlands.edu

© Springer Nature Switzerland AG 2021

K. Daimi et al. (eds.), *Advances in Security, Networks, and Internet of Things*,

Transactions on Computational Science and Computational Intelligence,

https://doi.org/10.1007/978-3-030-71017-0_5

authentication system verifies the identity of a person based on either their unique physiological traits [3, 4] or their unique behavioral biometrics [5, 6]. Biometric authentication is more user friendly in nature than the approaches that rely on passwords and secure tokens. While physiological traits can achieve high accuracy in user authentication, they are subject to a variety of attacks [7, 8] and also raise privacy concerns [9]. Moreover, the accuracy of physiology-based mechanisms may be substantially degraded by environmental factors, such as the viewing angle, illumination, and background noise [10, 11]. In contrast, behavioral biometrics (i.e., key stroke, voice, or gait analysis) appear less sensitive to ambient light or noise [12, 13].

There have been a few studies on the security and usability of behavioral biometric authentication for the IoT ecosystem. To the best of our knowledge, there are only two studies that adopted voice biometrics as an authentication mechanism for the IoT ecosystem. Shin and Jun [14] implemented voice recognition technology to verify authorized users for controlling and monitoring a smart home environment. Shin et al. proposed a voice recognition system that is divided into server and device parts. The role of the server part of the system is for user preregistration, user recognition, and command control analysis. The role of the device is command reception, device control, and then response. The type of models and techniques employed in their research is not discussed. Likewise, the implementation of their [1] model is this chapter.

The rest of the chapter is laid out as follows: Sect. 2 is the background, Sect. 3 is the related work, Sect. 4 discusses the motivation, Sect. 5 our proposed model is presented, Sect. 6 is the implementation, and Sect. 7 concludes the chapter.

2 Background

Automatic speaker verification [ASV] (Fig. 1) is a pattern recognition problem that predominantly works on speech signals. ASV systems intend to acquire different information from voice data and combine them for each speaker. For example, *idiolectal* and *prosody* identify [6] high-level attributes of a speaker's voice, while *short-term spectral* identifies low-level attributes of the speech signals. The latter is the main source of individuality in speech [5]. Low-level attributes are easy to extract and are most common when applied to ASV systems.

As can be seen in Fig. 1, there are three processes in any ASV system. These are *voice feature extraction*, *speaker modeling*, and *decision making*. The *feature extraction* process is the same in the enrollment and verification stages where the voice signals are converted into a sequence of frames. Each frame is a short window of the waveform with overlapping adjacent windows [15]. Considering the unique resource-constrained characteristics of IoT devices, our discussion will focus only on *short-term spectral qualities* [16] as it requires less computational resources. Additionally, the selection of appropriate feature extraction methods is crucial in this process because they influence the performance of the system. The two most popular

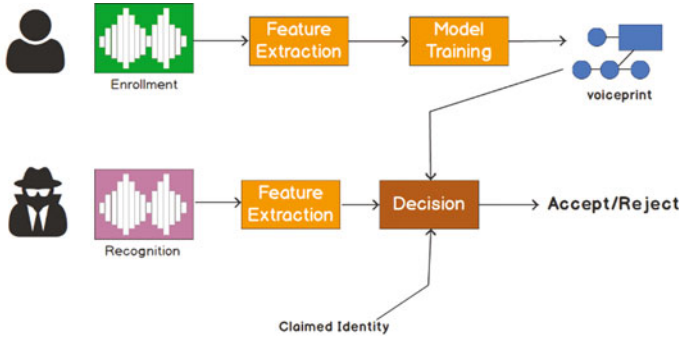


Fig. 1 ASV

spectral feature extraction methods, *filter bank analysis* and *linear predictive coding*, are discussed in the following sections [9, 17, 18].

2.1 Filter Bank Analysis

In filter bank analysis, the voice signals are expected to pass through a bank of band-pass filters that cover a range of frequencies consistent with the transmission characteristics of the signal. The spacing of the filters can be either be uniform or nonuniform, based on the perceptual criteria such as linear frequency cepstral coefficient [LFCC] or mel-frequency cepstral coefficient [MFCC]; the latter provides a linear spacing [9, 19].

2.2 Linear Predictive Coding [LPC]

In the LPC, the speech signal can be modeled by a linear process prediction. Signals at each time step use unique and specific periods of preceding samples that capture the temporal evolution of the features from one speech segment to another [9, 19].

1. In the ASV systems, 2. The speaker modeling step comes, 3. After features of the voice are extracted. ASV has the ability to construct a model λ_s for each user where “s” is user and λ is the specific model. Such a modeling depends on, for example, whether it is used for applications that use fixed words (text-dependent), or applications that use phonemes not seen in the enrollment data (text-independent).

Speaker modeling methods can be of two categories: nonparametric or parametric. The nonparametric approaches include templates which are suitable for a text-dependent verification system [9].

The parametric speaker modeling includes *vector quantization (VQ)*, *Gaussian mixture models (GMM)*, and *hidden Markov models (HMM)*. In VQ, a set of

representative samples of the user's enrollment voice is constructed by clustering the feature vectors. *GMM* has been proven to be very effective in the cases of a text-independent speaker recognition [15], also referred to as a refinement of vector quantization. HMM is suitable for text-dependent ASVs and is used for access control of personal information or bank accounts. Among the three techniques, GMM has been proven very effective for phones, and may be the best candidate for IoT devices [20]. There are also some other nonparametric and parametric approaches in the literature. However, those presented in this chapter are the most common techniques implemented in ASVs. The final process is decision making. In this process, an "accept or "reject" decision is delivered based on the verification models discussed above.

3 Related Work

This section reviews different biometric authentication mechanisms employed in the IoT ecosystem. Generally, there are two main types of IoT authentication methods including centralized and distributed architectures as shown in Figs. 2 and 3, respectively. The centralized architecture uses a centralized server to manage the credentials used in the authentication, whereas in the distributed architecture the authentication is accomplished point-to-point between the communicating parties [21]. Biometric authentication is achieved based on these two architectures. There are four basic biometric authentication performance measuring strategies" they include accuracy, scale, security, and privacy. Elements such as enrollment, biometric reference, comparison, networking, and personal biometric criteria are common in biometric authentication systems. Biometric-based authentication systems use two other factors: physical and behavioral [21]. The physical factors include fingerprint, face, iris, hand geometry, and palm print recognition, while the behavioral factors may include, but are not limited to, voice, signature, and gait recognitions [11]. Biometric systems have some advantages over conventional identity-based methods (password and ID); they cannot be transferred, stolen, lost, broken, or easily guessed [12, 22]. The acceptance and performance of the biometric

Fig. 2 Centralized authentication architecture

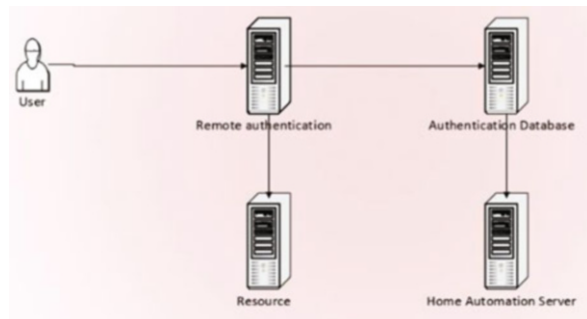
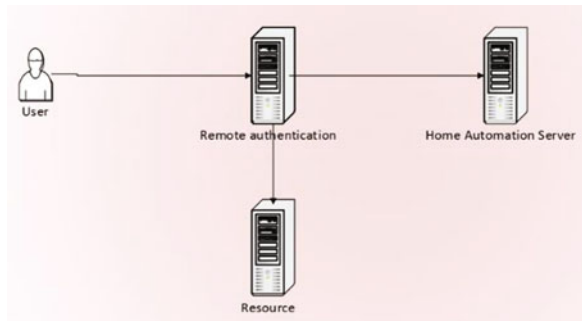


Fig. 3 Distributed authentication architecture



systems are presented in Table 1. From Table 1, the voice biometric systems are more suitable compared to other biometric systems. The voice biometric systems use voice rather than complicated input methods, like fingerprints which need special hardware for input. Hence, voice is appropriate for IoT where convenience is important. Some researchers have investigated and adopted voice based-biometric systems into the IoT ecosystem.

For example, Kim and Hong [23] used MFCC and pitch as voice features and the GMM in the voice authentication process for speaker recognition. Likewise, Chen et al. [24] proposed an authentication and authorization scheme that uses voice rhythmic pattern [11] for mobile IoT devices.

4 Motivation

Entity authentication refers to a process by which an agent in a distributed system gains confidence in the identity of a communication partner. In other words [25], defines authentication as a provision for ensuring the correctness of the claimed identity of an entity. Most of the time authentication is mistaken with authorization, which is concerned with the level of access or privilege an entity may possess.

In this light, security throughout the process of authentication has to be maintained. For example, if an attacker steals the credentials of a user and gains access to a smart-door lock of a house or the health-monitoring smart device of a patient, this could be life-threatening. Hence, security before, during, and after the authentication of a smart device is of the highest importance. However, biometric authentication suffers from the public nature of some biometrics, including the facial-feature method that uses the face as a biometric, which is easy to be replicate. Fingerprint biometrics are commonly left everywhere and can be reproduced. Likewise, voice biometrics also suffer from the issue of recording and replaying for authentication. Storing biometric data on servers also raises concern. For example, if a perpetrator gains access to the server where the biometrics are stored, the attacker may take those biometrics and access anything the biometric is used to protect. This poses a major problem.

Table 1 Comparison of biometric system

Factor type	Biometric systems	Weakness	Strength
Behavioral	Voice recognition	Has a relatively low accuracy, inefficiencies in certain circumstances	Needs no hardware, ease of use, widespread usage, can be used for remote authentication
	Signature recognition	Has a relatively low accuracy	Wide acceptance, non-rigging
	Detect behavior	Shows nonperformance in certain conditions	Continuous authentication
	Tough dynamics	Inconsistent accuracy, lack of efficiency under certain conditions	Continuous authentication, does not require specific hardware
	Keystroke dynamics	Inconsistent accuracy, lack of efficiency under certain conditions	Continuous authentication, does not require specific hardware
Physical	Fingerprint recognition	The need for additional hardware, the difficulty of obtaining high-quality images, the lack of efficiency in certain circumstances	Use and wide acceptance, low cost, good accuracy
	Face recognition	The need for additional hardware, lack of efficiency in certain circumstances	Use and wide acceptance, good accuracy and less fraud
	Iris recognition	The need for additional hardware, high cost, time-taking authentication	High precision, non-rigging
	Hand geometry recognition	The need for additional hardware, precision	Easy to use, less fraud
	Palm detection	The need for additional hardware, high cost	Public acceptance, high precision

There are also issues that are concerned with remote authentication. Normally, personal non-attended smart devices ask a user to remotely authenticate himself to his devices. However, there is an issue of trust with remote verification. Because the user sends his biometrics remotely for authentication, he cannot ensure that his biometrics data will not be hijacked and potentially be misused or mishandled. This raises an issue of trust or privacy. Because of this, our research focuses on a secure voice biometric-based authentication. This chapter specifically focuses on steps taken towards the primary testing of an IoT user-authentication model that uses voice biometrics. The security and resilience aspects of the model are ongoing.

5 Our Proposed Model

A voice biometric-based IoT user authentication model was proposed in one of our previous papers as presented in Fig. 4 [26]. The model has two phases, the enrollment phase when the user of the smart device speaks his voice for registration. The other phase is verification when the system checks whether the identity claimer is the real user by comparing the previously enrolled voice with the voice of the identity claimer. Subsequently, if the similarity of the two voices reaches a certain predefined threshold, then access is granted to the claimer; otherwise the claimer is rejected. The design criteria of the model is given in Table 2.

Fig. 4 The proposed IoT voice biometric model

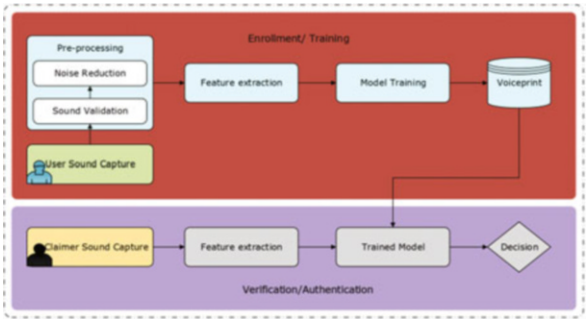


Table 2 Design criteria of the model

Design criteria	Description
Universality	A very high percentage of the population should have the characteristic. For example, virtually everyone has recognizable fingerprints, but there are rare exceptions.
Distinctiveness	No two people should have identical characteristics. For some otherwise acceptable characteristics, identical twins share virtually the same patterns, such as facial features and DNA, but not other features, such as fingerprints and iris patterns.
Permanence	The characteristic should not change with time. For otherwise acceptable characteristics, such as facial features and signatures, periodic re-enrollment of the individual may be required.
Collectability	Obtaining and measuring the biometric feature(s) should be easy, nonintrusive, reliable, and robust, as well as cost-effective for the application.
Performance	The system must meet a required level of accuracy, perform properly in the required range of environments, and be cost-effective.
Circumvention	The difficulty of circumventing the system must meet a required threshold. This is particularly important in an unattended environment, where it would be easier to use such countermeasures and a fingerprint prosthetic or a photograph of a face.
Acceptability	The system must have high acceptance among all classes of users. Systems that are uncomfortable to the user, appear threatening, require contact that raises hygienic issues, or are nonintuitive are unlikely to be acceptable to the general population

6 Implementation

6.1 Description of the Implementation

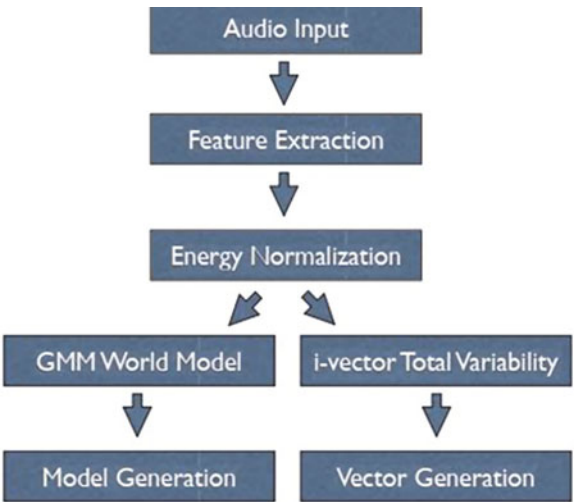
The normal process of voice biometric implementation usually includes creation of models from audio data, generation of tables, and self-authentication to the IoT manager. We, therefore, used a client-server model for the initial implementation. A user mobile device simulates running [1] a client, where the server simulates the IoT manager and handles the verification requests. By contrast, the IoT devices receive the command from the IoT manager and automatically respond to the command.

First, the user inputs his own voice command using the smartphone. Subsequently, the system on the server side determines whether the connection is for enrollment or verification and accordingly performs the process in each phase. New connections are considered first for enrollment, while the returning connections are considered by the server for verification by prompting the identity claimer with challenging words.

6.2 Open Source Software

To accomplish the initial test, we used a software package called Mistral/Alize. Mistral is tested by the National Institute of Standards and Technology [NIST] and with 0.5 error rate. Figure 5 shows the process of Mistral package.

Fig. 5 The processing of the Mistral software package



6.3 Dataset

There are a number of datasets for audio dataset selection, but we selected the MIT dataset, which appears to be recent and is considered to be good for initial results. MIT is designed for mobile environments. The dataset consists of 48 speakers including 22 female and 26 male. The dataset is used for the universal background model (UBM) training.

6.4 Preprocessing

In this step, after the user inputs his voice using his own smartphone, the voice is validated, and noise is removed from the voice signal. For example, Fig. 6 shows the raw voice signal without noise reduction, while Fig. 7 illustrates that the noise (silent part) is removed before it is submitted for feature extraction. In this process, using SPro [27], which is supported in the Mistral program, frame selection is performed by excluding silent frames longer than 100 ms. In addition, if a sample is different from 16 KHz, SPro performs resampling by default. Mistral requires the voice to be more compact. The most important aspect of this process is that the voice is converted into the binary format.

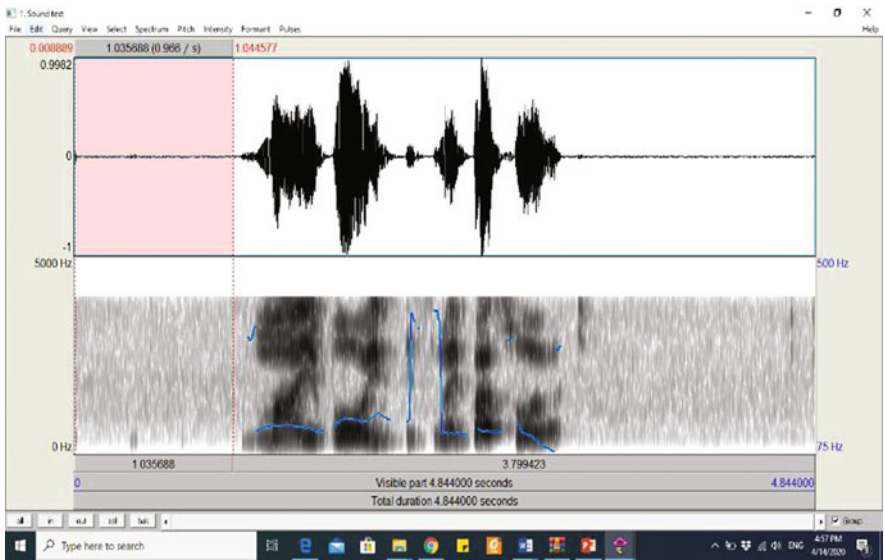


Fig. 6 Voice waves before noise reduction

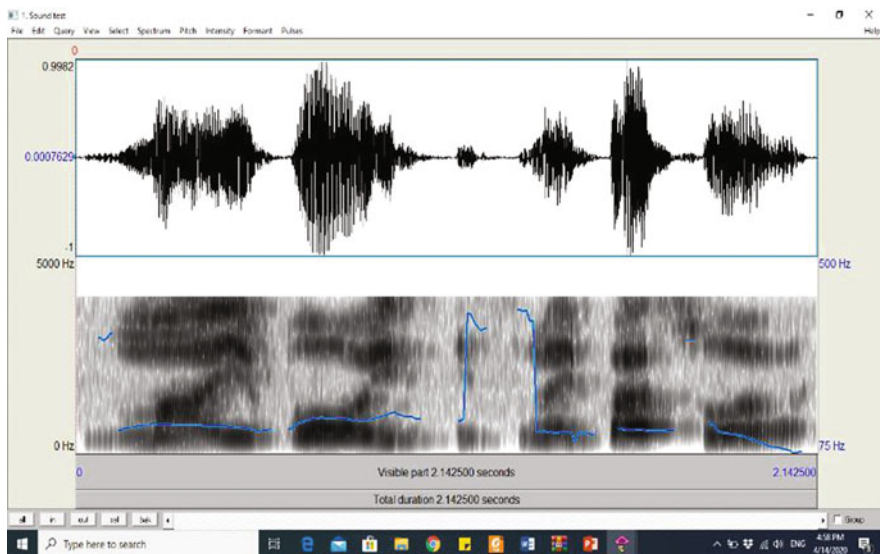


Fig. 7 Voice waves after noise reduction

6.5 Feature Extraction

In this step, using SPro, the feature extraction process is conducted. The voice data, preprocessed in the preceding step, is subjected to feature extraction. Feature extraction is conducted through 12 MFCCs, as shown in Fig. 8, and is subsequently stored in a parameter file. The front end of the system actually stops at this step and, according to our work, is accomplished at the user's smartphone. At this point of the testing, we used a virtualized Android OS to represent the mobile part of the implementation.

6.6 Voice Model Training/Server Side

In the Mistral toolkit, the next step is training the model. This step controls the list of users of the smart IoT device. Using the Train World process, we used the Gauss mixture model [GMM] to conduct this part of the training with the MIT dataset for the UBM training. This procedure requires an already trained UBM. The remaining 12 were kept for testing. For this part of the implementation, we used a virtualized Linux server to host the Android Things operating system, which acts as the manager of the IoT devices.

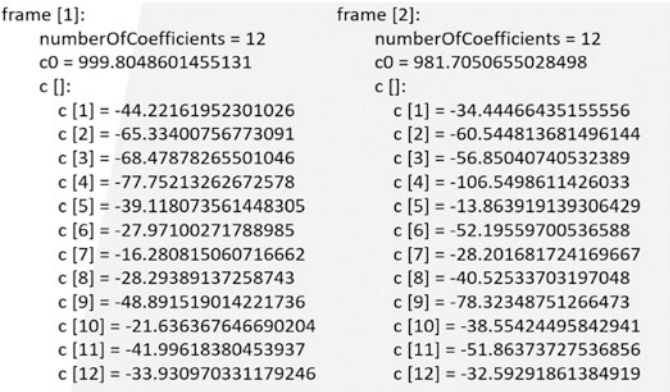


Fig. 8 12 MFCCs features

6.7 Verification

After both parts of the model have been trained, the data is to against the world UBM model to create verification. The tests were run twice; the first test included using enrolled speakers. The second test did not include enrolled speakers. After the testing phase started, scores were calculated for each test speech segment verification based on the Mistral toolkit. By using a decision threshold speaker model, verification could either be accepted or rejected. In commercial verification systems, users are required to test and decide on the threshold. However, in our implementation, we only speculated on the possible use of the threshold.

6.8 Result and Discussion

We had no problem with the enrollment of the speakers in the dataset. In the first test, only the intended target speakers were used to train at the UBM before they were enrolled. Figure 9 shows the detection error trade-off curve of the MIT voice dataset.

In the future, we will collect enrollment segments from many users anonymously and use different devices in regular daily life settings for a significantly improved result; this will build a more reliable UBM to be distributed for application. In addition, a secure remote authentication mechanism will be investigated. Voiceprint security, once stored in the voice database, will be included in the future research.