ICIMP 2021

The Sixteenth International Conference on Internet Monitoring and Protection

ISBN: 978-1-61208-862-4

May 30th – June 3rd, 2021

ICIMP 2021 Editors

Dragana Krstic, University of Niš, Serbia

ICIMP 2021

Foreword

The Sixteenth International Conference on Internet Monitoring and Protection (ICIMP 2021), held between May 30 – June 3rd, 2021, continued a series of special events targeting security, performance, vulnerabilities in Internet, as well as disaster prevention and recovery.

The design, implementation and deployment of large distributed systems are subject to conflicting or missing requirements leading to visible and/or hidden vulnerabilities. Vulnerability specification patterns and vulnerability assessment tools are used for discovering, predicting and/or bypassing known vulnerabilities.

Vulnerability self-assessment software tools have been developed to capture and report critical vulnerabilities. Some of vulnerabilities are fixed via patches, other are simply reported, while others are self-fixed by the system itself. Despite the advances in the last years, protocol vulnerabilities, domain-specific vulnerabilities and detection of critical vulnerabilities rely on the art and experience of the operators; sometimes this is fruit of hazard discovery and difficult to be reproduced and repaired.

System diagnosis represent a series of pre-deployment or post-deployment activities to identify feature interactions, service interactions, behavior that is not captured by the specifications, or abnormal behavior with respect to system specification. As systems grow in complexity, the need for reliable testing and diagnosis grows accordingly. The design of complex systems has been facilitated by CAD/CAE tools. Unfortunately, test engineering tools have not kept pace with design tools, and test engineers are having difficulty developing reliable procedures to satisfy the test requirements of modern systems. Therefore, rather than maintaining a single candidate system diagnosis, or a small set of possible diagnoses, anticipative and proactive mechanisms have been developed and experimented. In dealing with system diagnosis data overload is a generic and tremendously difficult problem that has only grown. Cognitive system diagnosis methods have been proposed to cope with volume and complexity.

Attacks against private and public networks have had a significant spreading in the last years. With simple or sophisticated behavior, the attacks tend to damage user confidence, cause huge privacy violations and enormous economic losses.

The CYBER-FRAUD track focuses on specific aspects related to attacks and counterattacks, public information, privacy and safety on cyber-attacks information. It also targets secure mechanisms to record, retrieve, share, interpret, prevent and post-analyze of cyber-crime attacks.

Current practice for engineering carrier grade IP networks suggests n-redundancy schema. From the operational perspective, complications are involved with multiple n-box PoP. It is not guaranteed that this n-redundancy provides the desired 99.999% uptime. Two complementary solutions promote (i) high availability, which enables network-wide protection by providing fast recovery from faults that may occur in any part of the network, and (ii) non-stop routing. Theory on robustness stays behind the attempts for improving system reliability with regard to emergency services and containing the damage through disaster prevention, diagnosis and recovery.

We take here the opportunity to warmly thank all the members of the ICIMP 2021 Technical Program Committee, as well as all of the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICIMP 2021. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICIMP 2021 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICIMP 2021 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Internet monitoring and protection.

ICIMP 2021 Chairs:

ICIMP 2021 Publicity Chair

Marta Botella-Campos, Universitat Politecnica de Valencia, Spain
Daniel Basterretxea, Universitat Politecnica de Valencia, Spain

ICIMP 2021

COMMITTEE

ICIMP 2021 Publicity Chairs

Marta Botella-Campos, Universitat Politecnica de Valencia, Spain
Daniel Basterretxea, Universitat Politecnica de Valencia, Spain

ICIMP 2021 Technical Program Committee

Vivek Adarsh, University of California, Santa Barbara, USA
Prashant Anantharaman, Dartmouth College, USA
Muhammad Ajmal Azad, University of Derby, UK
Lasse Berntzen, University of South-Eastern Norway, Norway
Francesco Buccafurri, Mediterranean University of Reggio Calabria, Italy
Paolina Centonze, Iona College, New York, USA
Paolo D'Arco, University of Salerno, Italy
Lorenzo De Carli, Worcester Polytechnic Institute, USA
Raffaele Della Corte, "Federico II" University of Naples, Italy
Parvez Faruki, Malaviya National Institute of Technology, India
Mathias Fischer, Universität Hamburg, Germany
Oliver Gasser, Max Planck Institute for Informatics in Saarbruecken, Germany
Kambiz Ghazinour. State University of New York in Canton, USA
Rita Girao-Silva, University of Coimbra &INESC Coimbra, Portugal
Ghaleb Hoblos, Normandy University, Caen, France
Zhen Huang, DePaul University, USA
Imane Idrissi, Normandy University/UNIRouen, France / USMBA University, Fez, Morocco
Mikel Iturbe, Mondragon University, Spain
Hamid Jahankhani, Northumbria University London, UK
Terje Jensen, Telenor, Norway
Basel Katt, Norwegian University of Science and Technology (NTNU), Norway
Irfan Khan Tanoli, University of Beira Interior (UBI), Portugal
Vitaly Klyuev, University of Aizu, Japan
Pushpendra Kumar, Manipal University Jaipur, India
Aditya Kuppa, Tenable Inc. / University College Dublin, Ireland
Yuping Li, Pinterest, USA
Pooria Madani, York University, Toronto, Canada
Pradip Mainali, OneSpan, Belgium
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Jims Marchang, Sheffield Hallam University, UK
Michael J. May, Kinneret Academic College, Israel
Anze Mihelic, University of Maribor, Slovenia
Aleksandra Mileva, University Goce Delcev in Stip, Republic of North Macedonia
Mahyar Tourchi Moghaddam, INRIA Grenoble-Rhône-Alpes, France
Lorenzo Musarella, University Mediterranea of Reggio Calabria, Italy
Sebastião Pais, NOVA LINCS | University of Beira Interior, Portugal

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

A Secure Healthcare System for Privacy-Preserving based on Blockchain Technology

Mohammed Adnan Mohammed
Computer & Embedded Systems. ENIS
University of Sfax
Sfax. Tunisia
Email: mohammed.adnan@enis.tn

Manel Boujelben
Computer & Embedded Systems. ENIS
University of Sfax
Sfax. Tunisia
Email: manel.boujelben@enetcom.usf.tn

Mohamed Abid
Computer & Embedded Systems. ENIS
University of Sfax
Sfax. Tunisia
Email: mohamed.abid_ces@yahoo.fr

Abstract- In recent years, Internet of Things (IoT) technology is recognized as a technological revolution used in different sectors, especially those with the emerged automation concept. It has many applications in various fields, for instance, smart cities, retail, healthcare, etc. However, many issues are yet to be fully addressed, such as scalability, connectivity, privacy and security. Blockchain technology has emerged as a promising solution for privacy and security challenges. It uses a decentralized distributed ledger, which records digital assets' information and keeps these records immutable and then reduces fraud risk. This paper proposes a Blockchain-based medical data protection system that enables users to control over their sensitive data collected from wearable sensors. Patients can upload medical records and healthcare providers can retrieve data while preserving sensitive health information against potential threats. We present a prototype implementation based on Quorum Blockchain and evaluate its memory and CPU time overheads using quorum profiling tool. The empirical results show that the integration of Blockchain technology with existing IoT systems is feasible and provides effective performance and security.

Keywords: Blockchain; Healthcare; IoT; Quorum

I. INTRODUCTION

Nowadays, the Internet of Things (IoT) technologies can provide solutions to sense, actuate with, and communicate over the Internet. The IoT plays a central role in turning current cities into smart cities, electrical grids into smart grids, etc. Furthermore, it visualizes a connected world, where things can communicate measured data and interact, making a digital representation of the real world through many smart applications [1]. Blockchain technology was utilized initially for protecting financial transactions, but when proving its effectiveness, it was used in other fields like transportation, supply chain, healthcare and energy [2]. Blockchain is identified as the key to solve scalability, privacy, and reliability problems related to the IoT paradigm. It can enrich the IoT by providing a trusted sharing service, where information is reliable and can be traceable. Data sources can be identified at any time and data remains immutable over time, increasing its security.

Electronic Health Records (EHRs) have been used as an effective method to store and manage medical data. Currently, EHRs are stored using the client/server architecture by which each hospital retains the stewardship of the patients' data.

Smart healthcare is a part of IoT systems. It facilitates the diagnosis of the diseases and remote monitoring for the patients' vital activities. As a result, these systems deliver faster responses and active treatment to save patients' lives with less effort and cost. As the data of these systems are very sensitive this leads us to many questions such as what are the challenges that face the fonctionning of smart healthcare systems, are these systems safe, how can they protect data from security threats, what are the effects of lack of security and privacy on the work of these systems and what are the most effective ways or technologies to protect data and overcome on these challenges.

Smart healthcare applications have many challenges, such as integration, data overload, accuracy, and cost. Briefly, the most important challenge which is discussed in this paper is data security and privacy. Data of these systems consists of information of patients and hospitals, and other stakeholders that participate in these systems. therefore it is sensitive and vulnerable to various security risks such as eavesdropping, hijacking, denial of service, and tampering. Thereby, these systems cannot be used safely by the health organizations and insurance companies.

Here the need emerged for methods or techniques to solve security challenges and protect data. Recently, there is an increasing trend in deploying Blockchain in the healthcare sector (e.g., public healthcare management, counterfeit drug prevention, and clinical trial). Therefore, this paper proposes to integrate Blockchain technology with EHR systems to protect data and make these systems safer and effective.

The remainder of this paper is structured as follows. Section II describes several related work to security challenges in IoT networks and Blockchain Integration into IoT systems. Section III presents the background information about Blockchain technology and its architecture. In Section IV, we further detail the Medical IoT application. Section V focuses on the proposed system steps. In section VI, the performance of our proposal is evaluated with regards to memory and CPU overhead. Section VII represents the challenges related to Blockchain-IoT integration. Finally, in section VIII, conclusions and future works are addressed.

 ISBN: 978-1-61208-862-4

II. RELATED WORK

This section illustrates the related work as below :

A. IoT Networks and their Security Challenges

The IoT is an emerging technology connecting sensors, vehicles, hospitals, industries, and consumers through internet connectivity. However, IoT applications suffer from many challenges. One of these challenges, or maybe the most important one, is security. Many researchers tried to solve the security issues of the IoT systems. The authors in [1] presented a comparative study of various existing architectures in IoT networks for malware detection and prevention. The work highlights different security requirements of IoT communication environment and provides various details of the malware programs. Nevertheless, it has only focused on one layer of IoT architecture and it does not present clear solutions for privacy and security problems.

The authors in [3] analyzed the IoT system's security issues, which helps to understand and improve IoT security architecture. To overcome security problems, the authors propose that smarter security systems should be implemented, including managed threat detection, anomaly detection, and predictive analysis.

The work [4] has conducted a comprehensive security risk assessment using the OCTAVE Allegro method, which stands for the Operationally Critical Threat, Asset, and Vulnerability Evaluation. Then, the authors have identified ten critical cyber and physical assets. As an outcome, approximately fifteen security risks originating from both inside and outside smart homes have been identified. The consequences or impacts of these risks have been described, assuming that the threats are realized. The suitable countermeasures for mitigating the risks to an acceptable level have been produced. This research focuses solely on identifying security threats, impacts or risks, and proper countermeasures for IoT-based smart homes. According to the impacts of attacks on the internet of things, the authors in [5] discussed the procedures to mitigate attacks as DDoS or Mirai attacks on the IoT systems. Their recommendations were that security community must respond more quickly to security needs and establish novel defenses or techniques to avoid disrupting the IoT networks or perhaps the Internet infrastructure itself.

Regarding the security threats of IoT applications and frameworks, the work [6] has explained various security threats at different layers of IoT applications. Also, they discussed the existing and upcoming solutions to IoT security threats, including Blockchain, fog computing, edge computing, and machine learning. They then illustrated the state-of-the-art IoT security with future research directions to enhance upcoming IoT applications security levels.

In the literature, the security of the main IoT frameworks is surveyed in [7]. The authors reviewed the proposed architecture, the essentials of developing third-party smart apps, the compatible hardware, and each framework's security features. The comparison of security architectures revealed that the standards used for securing communications and verifying the various security features and immunity against attacks are one of the most critical contemporary issues facing the IoT. Regarding the layers of the Internet of things systems, it is often necessary to characterize the different threats related to each specific layer of the IoT system model. The authors in [8] analyzed the IoT systems layers or their architectures to detect , which layer is most vulnerable to provide suitable security solutions. The result is that the most vulnerable level of the IoT system model is the perception layer (physical layer). This is due to many reasons, such as technological heterogeneity and constrained resources. Authors demonstrated that it is crucial to work on this level's issues by implementing lightweight security solutions that suit the heterogeneous environments with resource-constrained devices.

B. The integration of Blockchain Technology into IoT Networks

A Blockchain is an immutable distributed database to , which new time-stamped transactions can be appended and grouped into a hash-chain of blocks. The Blockchain protocol structures the information in a chain of blocks , which are linked together by a reference to the previous block. One of the most critical challenges of IoT systems is the lack of confidence. According to the literature, the integration of promising technologies like IoT and Blockchain will become a revolution in IoT systems.

Blockchain technology usage in an IoT context has been introduced in [9]. This work explains that Blockchain features, such as immutability, transparency, and data encryption allow tackling IoT challenges. Furthermore, IoT systems have a lack of intrinsic security measures. The authors introduced two usage patterns: Device manipulation and data management. At last, they discussed the main challenges faced by the integration of IoT and Blockchain. A secure Blockchain-based smart home framework has been proposed in [10]. The authors thoroughly analyzed the security concerning the fundamental security goals (confidentiality, integrity, and availability).

The authors [11] discussed implementing e-government in Smart Cities and the available technologies and challenges that face it from a security and privacy perspective. They illustrated how sensitive information goes online and the procedure to protect it while transmitted, stored, and processed. Concerning securing the IoT system layers, this work [12] presents a model of multi-layer secure IoT network model based on Blockchain technology. This model divides the IoT into a multi-level de-centric network and adopts Blockchain technology to ensure high security and credibility. This model provides a solution for the wide-area networking of the IoT. In the smart healthcare field, the authors in [13] proposed a Blockchain leveraged decentralized eHealth architecture , which comprises three layers:

(1) The Sensing layer: Body Area Sensor Networks, (2) The NEAR processing layer: Edge Networks , which consist of devices at one hop from data sensing IoT devices and (3) The FAR processing layer: Core Networks that comprises Cloud or other high computing servers.

A Patient Agent (PA) software executes a lightweight Blockchain consensus mechanism and utilizes a Blockchain leveraged task-offloading algorithm to ensure patient's privacy. The PA processes medical data to ensure reliable, secure, and private communication. Furthermore, concerning the Personal Health Record (PHR) and

 ISBN: 978-1-61208-862-4

Electronic Health Record (EHR), the authors [14] presented the prototype implementation and evaluation of the OmniPHR architecture model that integrates distributed health records using Blockchain technology and the openEHR interoperability standard. The system can maintain distributed data via a Blockchain that could be recovered with low average response time and high availability. Large eHealth systems should have a mechanism to detect unauthorized changes in patients' medical documentation and enable access permissions (transactional transparency).

In the context of transactional transparency, the work in [15] proposed a model of eHealth integrity based on Blockchain to ensure information integrity in the eHealth system. In contrast to existing solutions, the proposed model allows information removal, which is a legal requirement in many countries' eHealth systems. A Blockchain is mainly used to implement a data-integrity service. This service can be implemented using other mechanisms, however, a Blockchain provides a solution that does not require trusted third parties and works in a distributed eHealth environment.

III. BLOCKCHAIN TECHNOLOGY

This section describes the concept of Blockchain technology as next :

A. Blockchain Presentation

A Blockchain is an immutable distributed database to which new time-stamped transactions can be appended and grouped into a hash-chain of blocks. The Blockchain protocol structures information in a chain of blocks where each block links by a reference to a previous block; consequently, forming a chain [16]. Blockchain has many features or benefits. Firstly, it is the best way to secure recording the data on the network. Yet, it is considered as a mechanism for transparent storage; thereby, anyone can verify the information's authenticity on the network. Additionally, the network's data cannot be changed or tampered without incurring huge overheads, making it secure and efficient. Secondly, Blockchain is leading a fundamental shift different from the traditional Internet of information and communications to the Internet of Value, providing trust, achieved through implementing Blockchain technology among strangers. Consequently, data can be exchanged instantly and efficiently without the need for intermediaries or third parties. From the above, we can summarize the features of the Blockchain as follows:

- Trust: adding information (Transaction) to the Blockchain ledger is performed only after the network participants' approval. When satisfaction is received to prove that the information is trustful, an authentication of information is performed in short intervals, and records are updated in the participant's ledgers.
- Immutability and Transparency: The term **"**immutability**"** refer to information that can only be appended to previous data, Briefly, it means that each block is related to the previous block. Once the block enters, it cannot be changed or lost. Transparency is ensured while all changes are reflected in the ledger of all participants. It is worth mentioning that any part of the network can audit these changes.
- Substantial Improvements: Blockchain can reduce the cost and greater the speed when transferring money or other assets due to the facts that it works 24/7, it does not need intermediary working during "regular" business hours, nor require a commission to verify the truthfulness of the records [17].
- Disintermediation: One of the Blockchain's important features is the capability of removing the central model. The reason for this feature is it depends on the peer to peer model without the need for any central intermediary to authenticate transactions. Furthermore, Blockchain ledger (database) cannot be maintained by anyone but by all participating network computers distributed worldwide.

B. Taxonomy of Blockchain Systems

As listed in Table 1, Blockchain networks have three different types based on network nodes permissions:

- Public Blockchain (permission-less). A public Blockchain network allows anyone to join it, and all the users have equal rights.
- Private Blockchain (permissioned), unlike the previous type, it is a closed network where privacy is important. This network includes the participating nodes that only are pre-selected and vetted. They are permissioned and the users in this type do not have equal rights in the network.
- Consortium Blockchain: This type is considered as a partially private and permissioned Blockchain. It is a set of pre-determined nodes that are responsible for consensus and block validation. Therefore, it is a partially centralized system, owing to some selected validator nodes' control, unlike the private Blockchain (which is entirely centralized) and the public Blockchain (which is entirely decentralized). This type combines the previous two types, as user requirements, whether read or write permissions would be public or limited to the network participants [18].

TABLE 1 : TYPES OF BLOCKCHAIN NETWORKS

		Blockchain systems		
		Public Blockchain	Private Blockchain	Consortium Blockchain
Features	Access	- Anyone	-Single organization	-Multiple selected organizations
	Participants	-Permissionless - Anonymous	-Permissioned -Known identities	- Permissioned -Known identities
	Security	-Consensus mechanism -Proof of Work / Proof of Stake	- Pre-approved Participants - Voting/multi-party consensus	- Pre-approved Participants - Voting/multi-party consensus
	Transaction Speed	- Slow	-Lighter and faster	-Lighter and faster

C. Blockchain system Components:

The Blockchain system consists of many technical components that enable it to provide services, such as security, distributed ledger system, transactions, consensus protocols, cryptographic techniques, and smart contracts.

- Transactions: Blockchain network nodes perform this procedure to exchange information between them based on peer to peer. The source node generates then broadcasts it to the whole network for validation. Lastly, transactions are assembled to form the block.
- The Distributed Ledger: is an append chain of cryptographically-linked blocks of data, maintained and updated by a decentralized network, which means all network nodes share a copy of the information (records). The distributed ledger contains all the transactions on the Blockchain. The network nodes are encouraged by economic incentives to maintain and secure the system so that the data has robust protection from adversarial interference, double-spend, counterfeit, collusion, tampering, or other types of malicious actions. [19].
- The Consensus Mechanism: is how all accounting nodes reach consensus to determine a Blockchain transaction's effectiveness. In the Blockchain network, many different processes need to coordinate their actions and define the total order of the information that is stored on each block to put this into the context of a Blockchain-based system. These processes' challenge lies in reaching a consensus on the block that should be appended to the chain at each particular index. Blocks are time-stamped and thus are ordered chronologically. Therefore, each Blockchain system embeds a consensus protocol that aims to prove that all correct processes agree on the same block, and the chosen block is considered valid and proposed by one process [17]. According to that many consensuses, algorithms are proposed:

a) Proof of Work (PoW): This algorithm relies on the node to carry out mathematical operations to find a random number and obtain the accounting right. Bitcoin, Dogecoin, and Litecoin are among the digital currencies based on the PoW consensus mechanism. However, its resources consumption is high, as the whole network needs to participate in the operation, which has low performance and efficiency.
b) Proof of Stake (PoS): consensus mechanism is that the difficulty of obtaining a node's accounting right is inversely proportional to the stake held by the node. According to the proportion and time of coins taken by each node, the difficulty of mining coins can be reduced in the same proportion to increase the speed of finding random numbers.
c) Proof of Authority (PoA): The transaction and the block are validated by an approved node (called a validator) without a huge computational overhead of a mining process. The validator must authenticate on the Blockchain. The PoA Blockchain becomes safer and cheaper [19].
d) Practical Byzantine Fault Tolerance (PBFT): In this approach, a primary and a secondary replica are utilized in the consensus process. The secondary is continuously evaluating the primary decisions in the Blockchain and make any necessary actions if the primary is compromised.

- The Smart Contract: It is a predefined code that is automatically executed by a Blockchain miner. As a result, it updates the ledger status on the Blockchain network. These changes cannot be falsified or tampered with once a specific consensus mechanism confirm them. The smart contract refers to the code that realizes the functions of receiving, storing, and transferring information. The smart contract will be triggered automatically without the outside parties' participation once the conditions are met. Due to the decentralized nature and the cryptographic algorithms of the Blockchain, the participating parties do not have the authority to change the clauses individually, which makes them trustful [20].
- The Asymmetric Encryption and Authorization Technology: The account identification information is highly encrypted and can only be accessed under the data owner's authorization. To use the Blockchain, every node will get a pair of keys. The first key is called the public key , which is used as a unique address and shared with all nodes in the network. It encrypts the message (Transaction) and verifies the received signatures. The Second key is called the private key, which must be kept secret. It is used for signing Blockchain transactions and decrypting the received messages.

D. How the Blockchain Technology Works

The first block is created and called the "Genesis Block"; then, the second one is formed and connected to the first block in chronological order. Similarly, the following blocks are performed. The Blockchain users search the numerical solution that corresponds to the specific hash value, which is called "digging mine". Any user (node) who finds the solution broadcasts it to the whole network and it will get the reward. The rest of network users will stop looking for the solution and start verifying the numerical solution. When the numerical solution is verified, the newly built blocks are added to the existing Blockchain. After that, the complete Blockchain is generated [21].

To clarifies the work of Blockchain, we use a Bitcoin Blockchain as an example. If the source node wants to send bitcoins to another node (destination node) it will create the transaction and broadcasts it to the entire network. Then, all transactions are queued in the transaction pool. Miners create blocks (sets of transactions) to be added to the chain. Miners are required to check each transaction's validity, and the current block connects and refers to the correct hash of the previous block. By this way, it is easy to detect whether data from a block is tampered with or not. In this case, the proposed block is added to the chain, and all nodes update the distributed ledger. Finally, the send bitcoin

 ISBN: 978-1-61208-862-4

process (Transaction) from the source node to the destination node is complete.

IV. MEDICAL IoT APPLICATION

This section describes the concept of medical IoT applications as next :

A. EHR Systems

A key feature of an EHR is that health information can be created and managed by authorized users in a digital format that can be shared across the entire healthcare ecosystem. This includes patient information from wearable devices owned and controlled by patients to be sent to healthcare providers, physicians, specialists, pharmacies, laboratories, and emergency facilities. Electronic healthcare records will consist of health information from all providers involved in a patient's care [22]. EHR systems can improve many major areas in the healthcare industry as follows:

a) Physician productivity can speed up physician diagnoses and digitize administrative tasks.

b) Patient satisfaction: provide them the ability to quickly obtain their data and see , which areas of their health history require improvement.

c) Ensuring the confidentiality, integrity, and availability of the stored data because data sharing will be only among authorized users.

The EHR system uses encryption techniques and cryptographic signatures to achieve confidentiality and ensure electronic health data integrity and authenticity. EHR system also uses access authorization to health data records to avoid data breach risks. Nevertheless, when integrated with Blockchain technology, EHR can use Blockchain mechanisms to manage the health data. In EHR systems, the patient uploads encrypted data to the system. The authorized Healthcare Provider retrieves these data and decrypts them to provide diagnosis and encrypt them again to be sent to another unit, such as a laboratory or pharmacy, to complete the task.

B. Medical IoT –Blockchain Applications

Smart healthcare applications have many challenges, such as integration, data overload, accuracy, and cost. Briefly, data security and privacy is also a major concern. Data is sensitive and vulnerable to various security risks such as eavesdropping, hijacking, denial of service, and tampering. Therefore, the need emerged for methods or techniques to solve security challenges and protect data. Recently, there is an increasing trend in deploying Blockchain in the healthcare sector (e.g., public healthcare management, counterfeit drug prevention, and clinical trial) [22].

In the medical IoT applications, private Blockchain is used which mean just known persons (Known identities) can access to the network. As shown in Figure 1, the patient from house send symptoms over the network to the doctor. The doctor will then send it to another unit such as a laboratory, consultant or sometimes emergency unit . As a result, will get a diagnosis and then sent it to pharmacy that sends a medicine to the patient. Any person (node) in the network has a medical ledger. It contains a copy of the same medical records for all transactions in the network and automatically updates it when any transaction is sent across the network. The transactions (blocks) are immutable. Therefore, the Blockchain is considered as the best way to protect medical records or personal information.

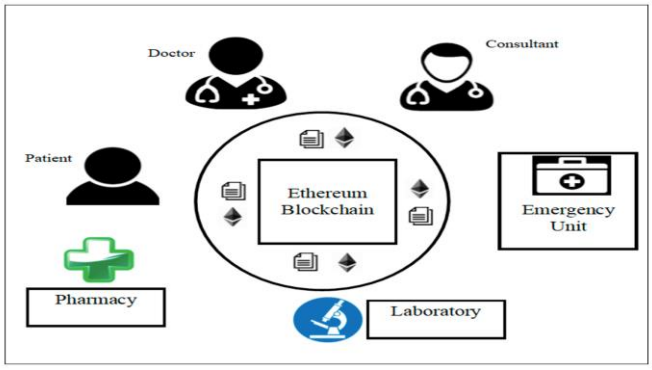


Figure 1: Blockchain-based healthcare system.

The advantages of applying Blockchain in the healthcare field (Medical IoT (healthcare) are as follows:

- Easier access to medical data: Because healthcare information's sensitivity is crucial, the costly overhaul of information technology systems and the overall regulatory environment and privacy concerns block its development. Blockchain technology may offer a solution by helping patients to get easy access to their data. Instead of navigating through multiple laws and processes of medical service providers to retrieve the information, this can be easy by utilizing the distributed ledger and the ability to maintain privacy through the public and private key. Furthermore, easiness for identifying the user and granting access to the appropriate medical records while keeping the overall data is ensured. Moreover, Blockchain thereby eliminates the centralized aspect where information is stored with one provider, as information is shared and accessible across all stakeholders upon the request.
- Facilitated sharing of medical records: The medical profession's problem is that medical data are extremely valuable for research purposes, and the improvement of overall medical conditions and operations is crucial. However, at the same time, this information is highly sensitive and faces massive legal hurdles with regards to sharing and aggregating the information from the various sources. Blockchain can solve this issue by allowing the patient's medical data to anonymize while keeping intact all pertinent medical information and rendering it serviceable in the aggregate. By using Blockchain, the patient would remain anonymous by keeping his/her private key secure and only sharing their information via their public key. In the meantime, the information remains publicly available for research purposes without the risk of revealing the patient's identity.
- Unification of medical records: In the case of using Blockchain, the medical information would be

 ISBN: 978-1-61208-862-4

decentralized thereby rendering it available directly to the patient, who can leverage the asymmetric encryption of the Blockchain to share their medical data with their physician while maintaining anonymity. Furthermore, the Blockchain system would allow for a standardized data format that would make it easier to share and communicate with different physicians. Lastly, users can choose to participate anonymously in medical research by offering their data to studies without the risk of personal identification [23].

V. PROPOSED SOLUTION

As mentioned earlier, the goal of this paper is to ensure healthcare data privacy and decentralized storage by using Blockchain technology. Due to the limited block size, privacy leakage and the increase of computational overhead, the EHR systems cannot upload the medical records and store them directly in the Blockchain. Therefore, to tackle these issues, few solutions have been proposed. Many applications use a cloud server as a third party. However, this solution has the risk of a single point of failure that means if any node is down the user cannot retrieve data of this node. Also, some curious cloud servers may collect sensitive patient data without consent. Therefore, in this paper, a decentralized peer-to-peer file system named InterPlanetary File System (IPFS) is used to avoid the risk of a single point of failure. IPFS is a decentralized file-sharing platform that identifies files through their content. It relies on a Distributed Hash Table (DHT) to retrieve file locations and node connectivity information. In a P2P network such as IPFS, if one node is down, other nodes in the network can serve needed files. According to [22], this approach is considered as the best solution to prevent a single point of failure in addition to other advantages, such as high storage throughput and faster data retrieval.

To describe our proposal illustrated in Figure 2 , we use a case scenario where a patient sends medical data to the EHR system and a health provider, such as a doctor or pharmacist, to request or retrieve these data. Our proposal uses a private blockchain network (permissioned network). In this type of network, the identities of participants are known and users are authenticated previously. Let us suppose that all the nodes such Local Healthcare Managers (LHM) and Electronic Healthcare Managers (EHM) have received a pair of private and public keys.

The patient is wearing some sensors and has a smartphone (or a PDA) to receive medical data from the sensors. The following steps show how patient's healthcare data will be registered and then accessed by a medical staff (healthcare provider):

1) Wearable sensors in the patient's body send data to a mobile phone.
2) The mobile sends these data to a LHM (e.g., pc device) which collects these data. This device works as a medical wallet.
3) The LHM gets the hash value H1 of the data , which will be stored in the decentralized peer-to-peer file system IPFS and sends a transaction to store this hash value in the Blockchain. As mentioned above, data is not stored in the Blockchain but only its hash value.
4) The Blockchain provides EHM with the value of H1 and this hash value will be considered as an index of the data to be stored in IPFS.
5) When H1 reaches the EHM, LHM encrypts its medical records whit the public key of EHM and signs this data using its private key. Then, LHM sends the data to EHM. Confidentiality of Data is ensured through encryption process and authenticity and integrity are provided by the signature. The EHM verify the signature with LHM public key and then decrypts the received data with its private key. From this data, it computes the hash value named H2. Smart contracts are triggered to verify if H1 is equal to H2. If it is the case, the received data is considered as valid data to be stored in IPFS. If not, data will not be stored and step 6 will not be executed.
6) EHM encrypts the medical data using its public key and sent it to IPFS to be stored.
7) Next, a new transaction will be sent to the Blockchain network, and then the ledgers of all the nodes are updated.

To retrieve specific medical Data from the system, a health provider must do the following tasks:

- It sends a transaction to the Blockchain network to fetch the required data index. Then, the index is sent to EHM. At this point, the EHM will request the data from IPFS system and compare between the index and hash value of the requested data. Smart contract is executed to ensure the validity of the data by comparing the two hash values. If they are equal, the EHM will retrieve data. Otherwise, the system will discard the request.
- Lastly, the EHM decrypts the medical data with its private key and then encrypts it again using the health provider's public key and finally, it sends it to the health provider. Health provider receives the required data and decrypts it using its private key. After that, the EHM will update the Blockchain. Additionally, LHMs and
- health provider's records will also be updated because each node has a copy of the Blockchain.

VI PERFORMANCE EVALUATION

This section describes the implementation process of the proposed system to evaluate its performance . In our scenario, we use a Blockchain network based on Quorum. It consists of seven nodes representing actors in the proposed system, such as patients, doctors and pharmacies, etc.

Quorum is an Ethereum-based distributed ledger protocol that has been developed to add the ability to create private Blockchain between selected participants and adds transaction privacy on normal Ethereum transactions [24]. It uses Raft consensus algorithm , which supposes that the consotium members are known and provisioned into the system. A leader is responsible for generating new blocks. RAFT need 2f+1 nodes to be setup in the network to have the capability to tolerate f faulty nodes [25].

 ISBN: 978-1-61208-862-4

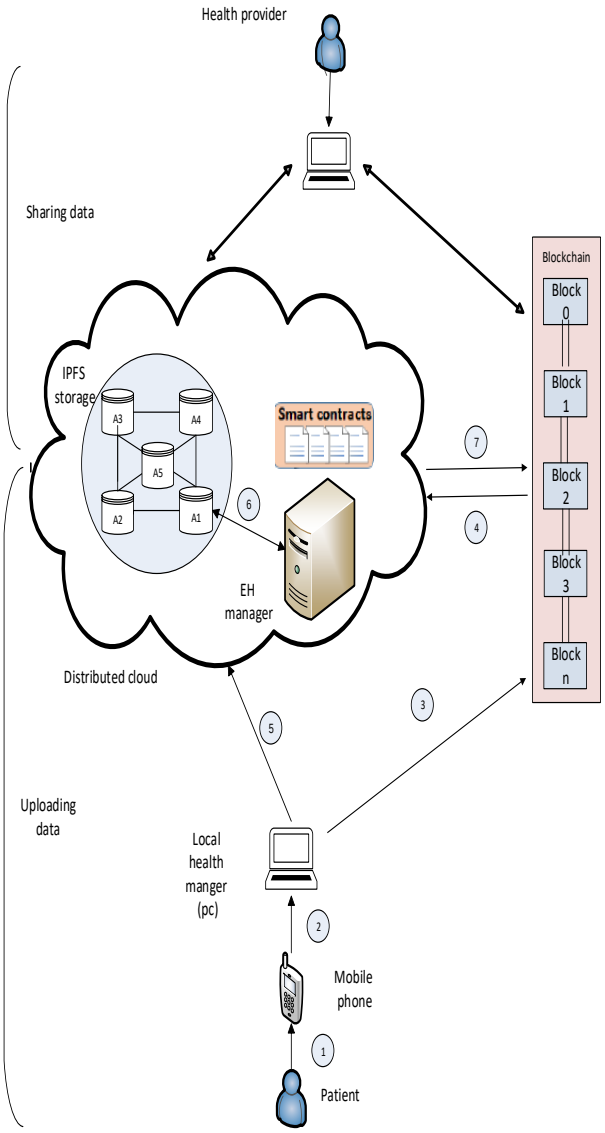


Figure 2: The proposed system architecture

We led the simulation for fifteen minutes, from minute 45 to 60, and during this period thirteen transactions are sent. Each transaction contains a medical data hash. The quorum profiling tool is used to benchmark network statistics on our quorum network. We focus on memory usage and CPU time as performance metrics. First, we calculate the average amount of memory usage in all the nodes. From Figure 3, it can be seen that the memory usage increases with the creation of new transactions and reaches the top when the time reaches 55 minutes. Specifically, at the beginning of this experiment, there weren't any transactions. Then, more transactions are exchanged and new blocks are added to the Blockchain. Therefore, more data is cached in the memory , which explain the memory overhead. We stop sending transaction at 55 minutes and accordingly, the memory usage becomes almost constant.

The results of the second experiment are illustrated by Figure 4. We evaluated the average amount of CPU time when the 13 transaction are processed. In the beginning, the CPU usage increases as the nodes create transactions and encrypt/decrypt data. Then. after the minute number 55 of the experiments, no more transactions are created and consequently, the average amount of CPU time decreases at the end of the experiment.

VII. CHALLENGES IN BLOCKCHAIN –IoT INTEGRATION

This section studies the main challenges that can be addressed when the Blockchain is applied technology to the IoT domain. The Blockchain is technically designed for an Internet scenario with a powerful computer; however, this characteristic is far from IoT's nature. Briefly, the exiting challenges are as follows:

- Storage capacity and scalability: In IoT healthcare applications, devices can generate gigabytes (GBs) of data in real-time, representing a significant barrier to its integration with Blockchain. It is known that some current Blockchain implementations can only process a few transactions per second. Furthermore, Blockchain is not designed to store large amounts of data like those produced in the IoT.
- Legal issues: The IoT implementation in the medical domain is affected by countries' laws or regulations regarding data privacy and protection. Laws that deal with information privacy and handling are a big challenge to be tackled in IoT and will be an even more significant challenge if used in combination with Blockchain.
- Security: One of the main challenges in the integration of the medical IoT with Blockchain is IoT data's reliability. Blockchain can ensure that data in the chain are immutable and can identify their transformations. Nevertheless, when data arrives already corrupted in the Blockchain, they stay corrupt. Corrupt medical IoT data can arise from many situations apart from malicious ones [26].
- Smart contracts: Providing a secure and reliable processing engine for IoT applications, filtering, and group mechanisms should be complemented smart contracts. Consequently, enabling applications to address the IoT depending on the context and requirements. Mining is still a key challenge in IoT applications due to its limitations. IoT is mainly composed of resource-constrained devices; however, globally the IoT has potentially huge processing power. The consensus algorithms of Blockchain technology, such as Proof of Work (PoW), consumes a lot of node energy, which is an additional challenge [27].

VIII. CONCLUSION AND FUTURE WORK

In this paper, the most important aspects of IoT and Blockchain technologies have been investigated. For a concise presentation, we first introduced Blockchain definition, types and fundamental characteristics. Next, we clarified the process of Blockchain work. This paper demonstrated next many issues related to the EHR systems. These systems cannot protect medical data from theft tampering, and other malicious activities. Therefore, the use

 ISBN: 978-1-61208-862-4

of a distributed storage system (IPFS) with Blockchain could protect the sensitive medical data from malicious attacks and security threats.

To achieve that, we proposed a system that consists of two-part: uploading medical data of patients and sharing or retrieving data by healthcare providers (doctors, hospitals, etc.). Finally, performance evaluation in terms of memory and CPU overhead is conducted. As presented by the implementation results, the proposal system allows users to share medical data in a reliable and quick manner. To achieve the desired level of patient privacy and network security, it uses different keys for encryption and decryption of medical data and prevents unauthorized access to the e-health system. Additionally, the proposal system decrease consumption of network resources and computational overhead by storing actual medical data in a distributed storage system (IPFS). We believe that our solution is a step towards effective management of e-health records , which is promising and important in most applications of healthcare.As future work, we will expand our system and implement it on more complex senarios.

REFERENCES

[1] M. Wazid, A. K. Das, Joel J. P. C. Rodrigues, S. Shetty, and Y. Parky, "IoMT Malware Detection Approaches: Analysis and Research Challenges", IEEE Access, Vol. 7, Nº 1, December 2019.

[2] H. Rathore, A. Mohamed, and M. Guizani, "A Survey of Blockchain Enabled Cyber-Physical Systems", Sensors, Vol. 20, Issue. 1, January 2020.

[3] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT) A Vision, Architectural Elements, and Security Issues" International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam. India, 10-11 Feb. 2017, ISBN:978-1-5090-3244-0.

[4] B. Ali, A. I. Awad, "Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes", Sensors, Vol. 18, Issue. 3, March 2018.

[5] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DdoS in the IoT: Mirai and Other Botnets", Computer, Vol. 50, Issue. 7, pp .80 – 84, July 2017.

[6] V. Hassija, et al, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures", IEEE Access, Vol. 7, pp. 82721 – 82743, June 2019.

[7] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks", Journal of Information Security and Applications, Vol. 38, pp.8-27, February 2018.

[8] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating Critical Security Issues of the IoT World: Present and Future Challenges", IEEE Internet of Things Journal, Vol. 5, Issue. 4, pp. 2483 – 2495, Aug. 2018.

[9] A. Panarello, N. Tapas,G.Merlino, F. Longo, and A. Puliafito, "Blockchain and IoT Integration: A Systematic Survey", Sensors, Vol. 18, Issue. 8, August 2018.

[10] A. Dorri, S.S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT Security and Privacy: The Case Study of a Smart Home", IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13-17 March 2017, ISBN:978-1-5090-4339-2.

[11] L.Yang, N. Elisa, and N.Eliot, "Privacy and Security Aspects of E-Government in Smart Cities", Smart Cities Cybersecurity and Privacy, pp. 89-102 , 2019.

[12] C. Li, L. J. Zhang, "A Blockchain Based New Secure Multi-Layer Network Model for Internet of Things", IEEE International Congress on Internet of Things (ICIOT), Honolulu, HI, 25-30 June 2017, ISBN. 978-1-5386-2011-3.

[13] Md. A. Uddin, A. Stranieri, I. Gondal, and V.Balasubramanian, "Blockchain leveraged decentralized IoT eHealth framework", Internet of Things, Vol. 9, March 2020.

[14] A. Roehrs, et al, "Analyzing the performance of a blockchain-based personal health record implementation", Journal of Biomedical Informatics, Vol. 92, April 2019.

[15] T. Hyla, J. Pejaś, "eHealth Integrity Model Based on Permissioned Blockchain", 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, VIC, Australia ,8-9 May 2019, ISBN. 978-1-7281-2600-5.

[16] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities", Future Generation Computer Systems, Vol. 88, Pages 173-190, November 2018.

[17] S. Makridakis, K.Christodoulou, "Blockchain: Current Challenges and Future Prospects Applications", Future Internet, Special Issues, October 2019 .

[18] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, and G. Das, "Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems", IEEE Consumer Electronics Magazine, Vol. 7, Issue. 4, pp. 6 – 14, July 2018.

[19] T. M. Fernández-Caramés, P. F.Lamas, "A Review on the Use of Blockchain for the Internet of Things" IEEE Access, Vol. 6, pp. 32979 – 33001, May2018.

[20] J. Yang, S. He, Y. Xu, L. Chen, and Ju Ren "A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks", Sensors, Vol. 19, Issue. 4, February 2019.

[21] Z. Zeng, et al, "Blockchain Technology for Information Security of the Energy Internet: Fundamentals, Features, Strategy and Application", Energies, Vol. 13, Issue 4, February 2020.

[22] S. Shi, D. He, Li Li, N. Kumar, M. K. Khan, and K.K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey", Computers & Security, Vol. 97, October 2020.

[23] J. A. Jaoude, R. G. Saade, "Blockchain Applications Usage in Different Domains", IEEE Access, Vol. 7, pp. 45360 – 45381, March 2019.

[24] J. P. Morgan Chase,"A Permissioned Implementation of Ethereium", GitHub repository https://github.com/jpmorganchase/quorum, 2021.

[25] D. Ongaro, J. Ousterhout, "In Search of an Understandable Consensus Algorithm", USENIX Annual Technical Conferences, Philadelphia. PA, 17-20 June 2014.

[26] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, and A. Peacockd, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities", Renewable and Sustainable Energy Reviews, Vol. 100, pp.143-174, February 2019.

[27] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey", International Journal of Web and Grid Services (IJWGS), Vol.14, No.4, pp.352 - 375, October 2018.

 ISBN: 978-1-61208-862-4

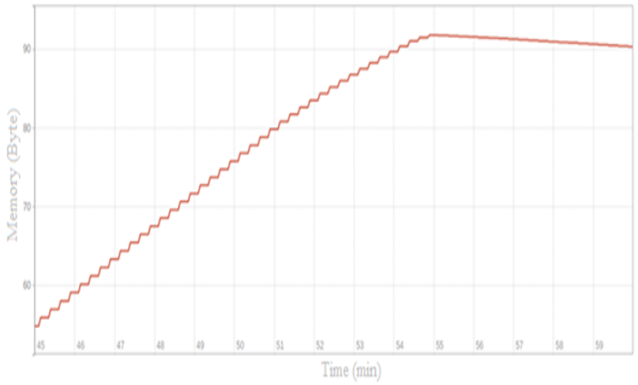


Figure 3: The average memory usage

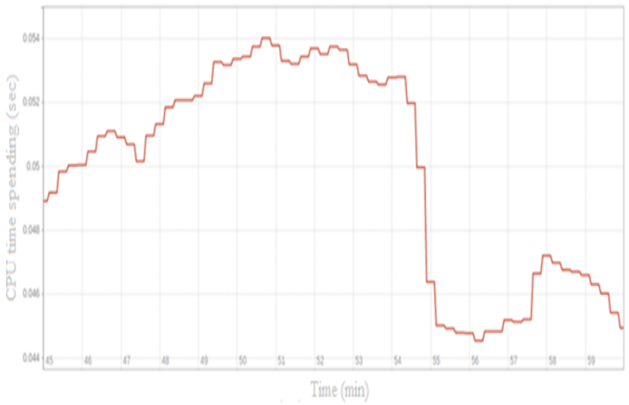


Figure 4: The average amount of CPU time.

 ISBN: 978-1-61208-862-4

SIGMA: Strengthening IDS with GAN and Metaheuristics Attacks

Simon Msika, Alejandro Quintero, Foutse Khomh
Polytechnique Montreal
Montreal, Canada
email: simon.msika@polymtl.ca, alejandro.quintero@polymtl.ca, foutse.khomh@polymtl.ca

***Abstract*—An Intrusion Detection System (IDS) is a key cybersecurity tool for network administrators as it identifies malicious traffic and cyberattacks. With the recent successes of machine learning techniques like deep learning, more and more IDS are now using machine learning algorithms to detect attacks faster. However, these systems lack robustness when facing previously unseen types of attacks. This work explores the possibility of leveraging generative adversarial models to improve the robustness of machine learning based IDS. More specifically, we generate adversarial examples, iteratively, and uses it to retrain a machine learning-based IDS, until a convergence of the detection rate. A round of improvement consists of a generative phase, in which we use GANs and metaheuristics to generate instances; an evaluation phase in which we calculate the detection rate of those newly generated attacks; and a training phase, in which we train the IDS with those attacks. We have evaluated the SIGMA method for four standard machine learning classification algorithms acting as IDS, with a combination of GAN and a hybrid local-search and genetic algorithm, to generate new datasets of attacks. Our results show that SIGMA can successfully generate adversarial attacks against different machine learning based IDS. Also, using SIGMA, we can improve the performance of an IDS to up to 100% after as little as two rounds of improvement.**

Keywords - Cybersecurity; IDS; Deep Learning; Machine Learning; GAN; Metaheuristics.

I. INTRODUCTION

In the last few years, the emergence of the Internet of Things (IoT) has led to new cybersecurity challenges. As connected objects now interact with the real world, privacy and security threats mitigation increasingly become a major issue [1]. With these new entities come the need to protect them from cyberattacks and similar intrusions. For instance, in 2016, the Mirai botnet [4] infected more than 600,000 Internet of Things devices from which were conducted massive Distributed Denial of Service (DDOS) attacks against several network companies all over the world.

Intrusion Detection Systems (IDS) are an essential tool for IoT system administrators: detecting a cyberattack is the first step to guarantee the privacy and security of users. But IoT also means a huge increase of internet traffic to analyze, and therefore the need to develop efficient, fast and robust algorithms to detect cyberattacks in this sensitive environment. Recently, machine learning models have shown astonishing performances in retrieving patterns from large volumes of data, in a very short amount of time. This success lead to their wide adoption in IDS [5]. However, as recent works on adversarial models have shown [2], machine learning algorithms, in particular deep learning tend to be fragile to adversarial examples. Using Generative Adversarial Networks (GAN) [2] an attacker can generate adversarial requests (i.e., attacks) that share the characteristics of requests that are considered to be genuine by the IDS. Although these GANs represent formidable weapons for attackers, as they can deceive most IDS into classifying attacks as benign traffic, they also provide an opportunity to preemptively strengthen intrusion detection systems against new attacks. By exposing an IDS to generated attacks as a preventive measure, we can prepare for new malicious behaviors.

In this paper, we propose a method to strengthen IDS against generated adversarial attacks, called **SIGMA**, which stands for **S**trengthening **I**DS with **G**AN and **M**etaheuristics **A**ttacks. The method consists in the iterative generation of attack datasets using adversarial learning and metaheuristics algorithms. The generated datasets are then used to retrain IDSs, i.e., teaching them to cope with patterns of attacks similar to those contained in our generated datasets. We repeat the retraining process until the detection rate of the IDS on generated attacks converges, meaning the detection rate is not improving anymore. We stop the algorithm after 3 runs without a detection rate improvement.

We evaluated SIGMA on IDSs based on four different classification algorithms: Neural Network, Random Forest, Support Vector Machine, and Naive Bayes Classifier. Each IDS was composed of two parts: a discriminator, trained to detect generated attacks, and an attack-classifier trained on the original dataset, to classify benign traffic and attacks. We trained a GAN and ran a local-search and genetic algorithm hybrid [21] to generate attacks against our IDSs. We compare the results of our model trained with both GAN and metaheuristics generated instances, with a model trained only with GAN generated instances over time and another model trained only with metaheuristics generated instances.

Results show that for IDS consisting of a Neural Network or a Random Forest algorithm, the SIGMA method allowed for a detection rate of 100% of generated attacks two to three times faster than the model strengthened only with the GAN generated attacks. However, models trained only with the instances created using metaheuristics search were almost completely unable to detect GAN generated attacks, suggesting that metaheuristics alone are not sufficient to increase the robustness of the studied IDS.

The remainder of this paper is organized as follows. Section II provides an overview of the technologies used in our model. We discuss the related literature in Section III. Section IV presents our strengthening method to increase the robustness of IDS against generated adversarial attacks (i.e., SIGMA). Section V describes the approach followed to

 ISBN: 978-1-61208-862-4

evaluate SIGMA, while Section V-E discusses the obtained results. Section VI discusses threats to the validity of our study and Section VII presents some implications of our work. Finally, Section VIII concludes the paper, summarising our findings along with some recommendations for future work.

II. Background

This section provides background information about Generative Adversarial Networks and metaheuristics used in this paper.

A. Generative Adversarial Networks

Generative Adversarial Networks are a class of unsupervised machine learning algorithm. They are composed of two neural networks: a generator G and a discriminator D.

Considering a dataset, the generator generates new data instances similar to the ones in the dataset. The discriminator, on the other hand, evaluates the data authenticity, i.e., whether or not the data it reviewed belongs to the actual dataset. The goal of the generator is to generate data labeled as genuine by the discriminator. The generator takes random numbers as input (referred to as random noise), and returns a data instance. With x as input of the discriminator D, we represent the probability that x is an attack generated by G as D(x). Therefore, D(x) is equal to zero when x is considered as an authentic data from the dataset, and equal to one when x is labeled as generated data, or fake. With z as random noise, we represent the instance generated by the generator network G as G(z). The generator G is trained to maximize the function $1 - D(G(\mathrm{z}))$.

As shown on Figure 1, the Generator and the Discriminator are trained simultaneously, therefore being on a constant battle throughout the training process.

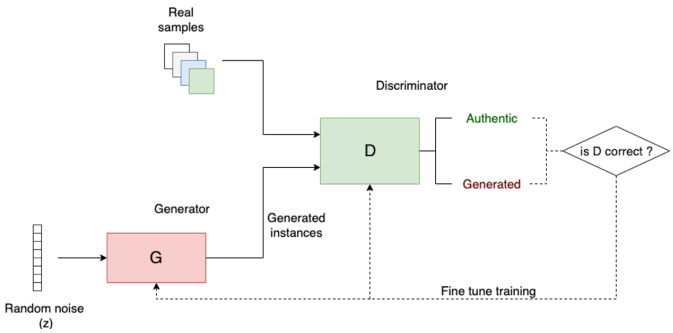


Figure 1. Diagram of a Generative Adversarial Network (GAN).

This algorithm has rapidly grown in popularity thanks to its performance in image generation [15]. It can generate realistic examples, and has a better performance than Deep Belief Networks or Boltzmann Machines [2].

GAN are also notably used to disrupt trained classifiers [9]: slight controlled modifications to the original input leads to misclassification. This has been extensively applied to images classification, due to its impressive results in this field: for instance, small visible changes made to "Stop" traffic signs tricked autonomous cars into misclassifying them into speed limit signs [27].

In this paper, we will use this ability of GANs to disrupt trained classifier by training them to generate attacks able to bypass the detection algorithm, *i.e.,* attacks classified as benign traffic by our IDS. Adding subtle modifications to the features of existing attacks could in fact lead to misclassification.

B. Metaheuristics

A metaheuristic is an algorithm used to find, generate, or select a heuristic (i.e., a partial search algorithm) that can provide a sufficiently good solution to an optimization problem with incomplete information. This type of algorithm is usually employed to solve computationally hard problems for which regular optimization would be too costly. Even if they do not guarantee finding the optimal solution for the problem, they usually provide good results, often close to the optimal solution [14]. A metaheuristic approach could be either single-solution based or population based.

A single-solution based approach could be local optimization: we randomly initialize a solution and explore the $neighbourhood$ of the solution by applying local changes to the current solution. The search continues until a solution meeting the initial stopping criteria is found or a time bound is elapsed. Local optimization could be very effective in case the criterion to maximize only has a single optimum. In other cases, the local search algorithm can converge to a local optimum, therefore not giving the best possible solution.

The hill climbing algorithm is an example of local optimization algorithm. This algorithm is an iterative algorithm that tries to improve a solution by making an incremental change to it. If the change produced a better solution, then it becomes the new solution, and another incremental change is made to this new solution. This algorithm runs until there is no further improvement possible.

Genetic algorithm [24] is an example of a population based metaheuristic. It is inspired by the process of natural selection. It starts with a population of solutions, where each solution is randomly generated. The population then evolves until the stopping criteria is met or until a certain number of generations is reached. From this pool of solutions, we select the best solutions (selection) and recombine them into a new population of solutions (crossover). We then apply random mutations to this population, in order to have a diverse population of solutions and possibly exploring other parts of the solution space that were not explored yet. As a global search algorithm, the genetic algorithm metaheuristic is more likely to find global optima for multimodal functions but it is slower at converging [19].

Since the genetic algorithm is rather slow to converge, it is possible to combine those two approaches (local optimization and a population based solution) to have a faster convergence. We then refer to this method as an hybrid algorithm [21]. It consists of a slight modification of the genetic algorithm to incorporate a local optimization element: after the selection process, we optimize each solution of the population with the local search algorithm. This leads to overall better results,

 ISBN: 978-1-61208-862-4

since the local search can only improve a solution, and could mean a faster convergence [14].

The overall processing of the hybrid algorithm is exposed in algorithm 1.

Algorithm 1: Pseudocode of the hybrid algorithm

Input: Instance (I), size of population (α), selection rate(β), mutation rate (m), number of iteration (nb_it), Local search algorithm (local_search)
Output: Population of solutions to I

```
/* Initialization                                  */
/* Generate α random solutions to I                */
solutions = generate_random(α);
for i = 1 to nb_it do
    /* Local Search                                */
    solutions = local_search(solutions);
    /* Selection                                   */
    n = α ∗ β ;
    pop = select_best(n, solutions) ;
    /* Crossover                                   */
    p = (α − n);
    for j = 1 to p do
        randomly select Sol_A and Sol_B from solutions;
        generate X_AB by mixing Sol_A and Sol_B;
        save X_AB to offsprings;
    end
    /* Mutation                                    */
    for child in offsprings do
        mutate child with probability m;
    end
    solutions = pop + offsprings
end
return solutions
```

This hybrid method has often been used to solve complex problems with good results [14].

In the intrusion detection domain, algorithms can be used to generate attacks that the intrusion detection system is unable to detect. In this case, a solution would be an actual attack, and all operations (cross-over, mutation, etc.) would be modifications of the attacks features.

III. Related work

Over the last two decades, researchers have built several intrusion detection datasets by extracting different network features from real networks during cyberattacks. [11] [22] Different machine learning algorithms have been explored to build IDS: from a simple feed-forward neural network, to Extreme Machine Learning [12], to complex Recurrent Neural Networks [23]. Studies show that even simple algorithms, such as a Support Vector Machine or J48 decision trees, could lead to good detection results, with 95% accuracy for the SVM and more than 97% accuracy for the decision tree [13]. Those algorithms could be used in practice by smart objects as intrusion detection systems. In fact, they don't require as much energy as complex Deep Learning models, which is an important factor to consider with resource-constrained environment [23].

Nonetheless, these machine learning methods suffer from a severe flaw as an IDS: they are totally vulnerable to new types of attacks. Successful attacks can lead to terrible consequences: economic loss, important privacy issues for smart objects users, etc. Moreover, it is now possible to automatically build new attacks against which those systems will be utterly useless, thanks to metaheuristics and Generative Networks, for example.

The use of metaheuristics for attack generation has been explored by Jan et al. [3]. In this work, Hill Climbing and Genetic algorithms are used to generate malicious XML injections. These generated attacks were used for testing purposes but demonstrated the possibility to automatically create attacks using metaheuristics techniques; in particular, the genetic algorithm managed to create a wide variety of attacks evading the web application sanity check more than 95% of the time.

Hu et al. [8] leveraged generative adversarial networks for malware detection, by considering the detection algorithm as a black-box (as would an attacker). The attacker does not know the internal structure of the detection model, but only knows the detection result of the detection model under attack. Even without having any information on the detection system, this approach led to very impressive results; GANs deceiving the malware detection algorithm almost every time.

Furthermore, recent work has shown that it is possible to generate adversarial examples with intrusion detection datasets. In particular, Lin et al. [7] used a Wasserstein GAN [25] to generate adversarial attacks against different classifiers considered as black-box algorithms by the attacker, trained with the NSL-KDD [13] dataset. The GAN was able to mislead several classification algorithms into classifying generated attacks as benign traffic. Nonetheless, the NSL-KDD dataset is now 10 years old and its relevance is then questionable. Moreover, researchers have pointed out several problems with NSL-KDD [16], e.g., the lack of Remote to Local and User to Root attacks, as well as the lack of more recent type of attacks [18].

These last few years, some progress has been made on protecting Intrusion Detection Systems against generated adversarial attacks. Generative models are a double-edged sword, as they can be preemptively used to train the detection model as well.

Cordy et al. [10] created increasingly resilient defense strategies to detect training attacks against a clustering-based IDS. The IDS was improved by simultaneously searching for attacks against the IDS and constantly improving the defense strategy: two genetic algorithms (one for creating attacks, the other to elaborate defense strategies) were used. Their resulting system detected 98% of the generated attacks, whereas the attack generation process systematically found a way to deceive the IDS without defense strategy. This promising result suggests that metaheuristics can be successfully used to preemptively strengthen an IDS against generated attacks. Nonetheless, this work does not provide any insight regarding the vulnerability of this strategy to other types of generated attacks (GAN generated instances for example).

The work of Ferdowsi and Saad [17] presents an approach to deploy a distributed intrusion detection architecture capable of detecting adversarial generated attacks. In this work, GANs were trained to generate adversarial attacks, and were then used to train a discriminator, which determined whether the current internet traffic was benign or an attack. However, this

system might be susceptible to iterative generated attacks: once the discriminator is trained, it may still be possible to find ways to generate instances able to bypass the detection system. An IDS resilient to generated iterative attacks has not yet been explored in the intrusion detection domain.

IV. SIGMA: An approach to improve the robustness of IDSs

In order to increase the robustness of IDS, we propose the following SIGMA method.

We take as input a Machine Learning based Intrusion Detection System, and a dataset consisting of attacks and benign traffic. We iteratively generate attacks with two different methods to train the IDS.

Each training iteration is designated by its number. We note $Score_i$ the detection rate of generated attacks by the IDS at iteration number i, meaning:

$$Score_i = \frac{number\ of\ detected\ generated\ attacks}{total\ number\ of\ generated\ attacks}$$

We consider that the generated attacks detection rate of our IDS has converged if: For $\epsilon > 0$, there exist an iteration number N, such that for all iterations i after N, we have:

$$|Score_i - Score_{i+1}| < \epsilon$$

The SIGMA method instructions are as follows:

While the generated attacks detection rate of our IDS has not converged:

- Step 1: We train a GAN to generate adversarial attacks against the IDS, considered as a black-box. The goal for this algorithm is to generate attacks deceiving the intrusion detection system. Considering the same notation as in Section II.B, the function to maximize for the generator is $1 - D(G(z))$ where $G(z)$ is the generated attack, $D(x)$ is the probability (computed by the IDS) that x is an attack: the IDS plays the role of discriminator.
  At each iteration, the generative algorithm generates new attacks to fool the Intrusion Detection System.

- Step 2: We use the trained GAN to generate attacks against the IDS. We evaluate the score of the detection system for these generated attacks. If the score has not improved for 3 consecutive rounds, we stop the algorithm.

- Step 3: We run a Search-based method in order to search for other possible attacks deceiving the IDS that the GAN might have missed.
  The function to maximize for this generative algorithm is: $1 - D(sol)$ where sol is the solution generated by the Search-based algorithm, and $D(x)$ is the probability (computed by the IDS) that x is an attack.

- Step 4: We use the Search-based method to generate attacks against the IDS. We then train the IDS with the generated instances from both algorithms (i.e., GAN and metaheuristics) and with data from the original dataset. Exposing its classifier to real data and generated attacks prevents it from overfitting to generated instances and losing accuracy on other type of traffic.

The overall proceedings is illustrated in the algorithm 2.

Algorithm 2: Pseudocode of the SIGMA process

```
Input: IDS to improve (IDS), training set (train_set),
Output: Improved IDS
converged = False;
counter = 0;
previous_score = 0;
while converged = False do
    /* Step 1: GAN training                          */
    generator = GAN.train(IDS, train_set));
    /* Step 2: Attack generation and
       evaluation                                    */
    GAN_attacks = generator(noise);
    predict = IDS(GAN_attacks);
    score = nb_attacks(predict) / length(GAN_attacks);
    if score ≤ previous_score then
        counter = counter + 1;
    else
        counter = 0;
        previous_score = score;
    end
    /* If the score has not improved after 3
       rounds, we stop the algorithm                 */
    if counter = 3 then
        converged = True;
        break;
    end
    /* Step 3: Search-based method                   */
    search_based.run(IDS);
    SB_attacks = search_based.generate();
    /* Step 4: IDS Training                          */
    IDS.train(GAN_attacks);
    IDS.train(SB_attacks);
    IDS.train(train_set);
end
return IDS
```

By combining attacks from both the Machine Learning and the Metaheuristics methods, we expect to explore a larger solution space since the two techniques are significantly different; we expect the generated attacks to be widely distinct. Being confronted with a large sample of diverse attacks, an IDS is likely to gain in robustness.

V. Evaluation of SIGMA

In this section, we evaluate the effectiveness of SIGMA at improving the effectiveness of an IDS. The quality focus is the improvement of the attack detection rate, through iterative reinforcement using GANs and metaheuristics. The perspective is that of researchers interested in developing efficient IDS, and practitioners interested in improving the robustness of their IDS. The context consists of the CICIDS2017 benchmark dataset [22], containing 11 types of networks attacks, and four machine learning-based IDS (i.e., a 3-layers Neural Network, a Random Forest, A Support Vector Machine (SVM), and A

Naive Bayes Classifier). In the following, we provide detailed information about the CICIDS2017 benchmark dataset and the implementation of SIGMA using the four selected machine learning-based IDS.

A. Dataset

The CICIDS2017 benchmark dataset [22] consists of more than 80 network flow features (flow duration, destination port, etc.). Table I provides a summary of those characteristics. This recent intrusion detection dataset contains 11 types of attacks along with benign traffic. Each entry of the dataset consists of more than 80 columns (namely the extracted network flow features) and is labeled as one of those 11 types of attacks or as benign traffic. We grouped the 11 different attacks into four different groups as shown in Table II, building four different balanced binary datasets (Attack, Benign), to counterbalance the unbalanced number of attacks per type.

TABLE I
SOME NETWORK FEATURES USED BY CICIDS 2017.

Feature name	Description
fl_dur	Flow duration
tot_fw_pk	Total packets in forward direction
tot_bw_pk	Total packets in backward direction
fl_pkt_s	Number of packets transferred per second
ack_cnt	Number of packets with ACK
pkt_size_avg	Average size of packet
idl_avg	Mean time a flow was idle

TABLE II
ATTACKS LABELS AND DISTRIBUTION IN THE CICIDS2017 DATASET.

Attack group	Number of attacks	Types of attack
Denial of Service	252661	DOS Hulk DOS GoldenEye DOS Slowloris DOS Slowhttptest
Distributed DOS	128027	DDOS
Bruteforce	15342	FTP-Patator SSH-Patator Bruteforce Portscan Botnet
Infiltration	720	SQL Injection XSS Heartbleed Infiltration

We first deleted the constant columns of the dataset, as they don't provide any useful information for classification. Data now consists of 71 columns, 70 of them being network flow features, and the last one being the label ($i.e.$, 0 if it is benign traffic, 1 if it is an attack).

Then, since the values of each feature throughout the data widely varies, each column was normalized to have values between 0 and 1. Feature scaling allows for much faster convergence for neural networks.

We normalized data by applying the min-max normalization, namely:

$c'_i = \frac{c - c_{min}}{c_{max}}$.

Where:

- c_i is the column from the original dataset.
- c'_i is the normalized column.
- c_{min} is the minimum value of the column.
- c_{max} is the maximum value of the column.

Each dataset was split into a training set and a test set, respectively representing 90% and 10% of the overall dataset.

B. Implementation of SIGMA

Step 1: GAN training

We chose to implement SIGMA with a 4-layers Wasserstein GAN. The architecture of the GAN is detailed on Fig. **??**. The dimensions of hidden layers were chosen experimentally, being the ones with the best results.

As mentioned in Section II.B., the Wasserstein GAN takes random numbers (or random noise) as input to generate attacks. We refer to the number of random numbers as the random noise size.

The goal of this generator is to generate attacks able to deceive the IDS. To ensure that the output of the generative algorithm is indeed an attack, we keep the functional features of an original attack.

Since every feature of our data has been normalized, each feature is represented as a number between 0 and 1. As shown on Figure 2, we keep the functional features of real attacks for our generated attacks.

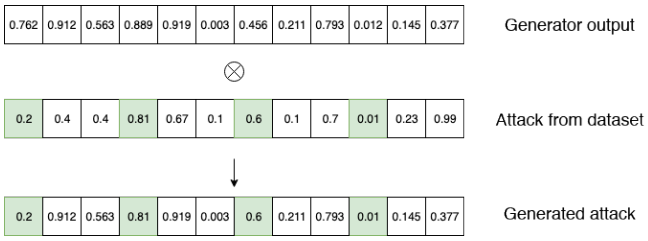


Figure 2. Diagram of the generative algorithm's process. In green, the functional features of the attack.

The functional features per attack type were identified by a statistical analysis of the datasets, with the help of the analysis conducted by the creators of the dataset [22]. They are presented in Table III.

With the aim to have a Generative Adversarial Networks with the best performance and therefore explore the largest attack space possible, we trained several Generators with different sizes of noise as input. Furthermore, since there is unpredictability in the training of Generators due to the randomized weights initialization, we trained the Generators several times. We then select the GAN with the best performance among those, *i.e.,* the most able to deceive the IDS.

 ISBN: 978-1-61208-862-4

TABLE III
FUNCTIONAL FEATURES PER ATTACK TYPE. THOSE FEATURES ARE NOT GOING TO BE MODIFIED BY THE GENERATOR.

Attack group	Functional features
DOS	Flow Duration, Active Mean, Average Packet Size, Packet Length Std, Flow IAT Mean, PSH Flag Count, Idle Max
DDOS	Flow Duration, Bwd Packet Length Std, Average Packet Size, Packet Length Std, Flow IAT Std, ACK Flag Count
Bruteforce	PSH Flag Count, Flow Duration, Total Length of Fwd Packets, Init Win bytes forward, Packet Length Std, Subflow Fwd Bytes, Fwd PSH Flags
Infiltration	Subflow Fwd Bytes, Total Length of Fwd Packets, Flow Duration, Idle Mean, Active Mean, Init Win bytes backward, PSH Flag Count

Algorithm 3: Pseudocode of the GAN training process

```
Input: IDS (IDS), training_set (train_set), Maximum noise size
       (max_noise_size), Number of training epochs
       (nb_epoch)
Output: Trained GAN
/* Initialization                                   */
best_score = 1.0;
noise = 1;
for attempt = 1 to 5 do
    for noise_size = 1 to max_noise_size do
        /* We construct a GAN with the
           corresponding noise size as input
           */
        GAN = Generator(noise_size);
        for epoch = 1 to nb_epoch do
            for (batch, labels) in train_set do
                /* First select the attacks from
                   the training set                 */
                is_attack = non_zero(labels);
                attacks = select(batch, is_attack);
                /* Then generate attacks            */
                z = random_noise(noise_size);
                generated_attacks = GAN(attacks,z);
                /* Backpropagation                  */
                loss = mean(IDS(generated_attacks));
                loss.backward();
                optimizer.step();
                if loss ≤ best_score then
                    best_score = loss;
                    noise = noise_size;
                    best_GAN = GAN;
                end
            end
        end
    end
end
return best_GAN
```

The process followed to train the GAN is presented in algorithm 3.

Step 2: Attack generation and evaluation

This step is to evaluate the current score of our IDS. To do so, we need to generate attacks with the GAN and gauge the robustness of our IDS against those attacks.

After the GAN has been trained at step 1, we use it to generate attacks. Generated attacks will use the functional features of attacks from the test set.

We evaluate the score of the IDS with those generated attacks. With previous notations, we consider that an instance x is considered an actual attack by the IDS if $D(x) > 0.5$. The score is therefore the number of generated attacks $G(z)$ with $D(G(z)) > 0.5$, divided by the total number of generated attacks.

If the score has not improved in three rounds, we stop the algorithm.

Step 3: Search-based method

In this step, we run a metaheuristic algorithm in order to generate additional attacks to further improve our detection system.

As our Search-based method, we used an hybrid genetic local-search algorithm. Indeed, local search and the Genetic Algorithm both have their pros and cons. The Genetic Algorithm is rather slow to converge whereas the Local search could converge to local optima. We chose to combine the two with an hybrid genetic algorithm [21], as it has been demonstrated to have been more efficient in complex problems, such as the Traveling Salesman [20].

The hybrid algorithm that we chose is a modification of the genetic algorithm: before proceeding to the selection process of the algorithm, every solution from the solution pool is improved by the local-search algorithm. As each solution is enhanced before the selection process, this algorithm allows for overall better performances, and usually a faster convergence than the standard Genetic Algorithm.

The goal of this metaheuristic algorithm is also to generate attacks against the IDS. Similarly to the proceedings of the GAN, functional features of our generated attacks will be from real attacks from the original dataset.

We first create a population of random solutions. We chose a population size of 30, as the recommended values in the literature are within the range of 30 to 80 [26]. Having a bigger population affected the performances of our algorithm.

Before the selection process, we optimize each solution of the population with a local search algorithm. The pseudocode for this local search method is given in algorithm 25.

Crossover is made by selecting two parents in the solution pool. We select only members of the population with the highest score (meaning, the attacks the most able to fool the IDS). The offspring will have the first half of its features from its first parent, and the other half from its second parent.

The mutation process is carried out to the entire population of children of this iteration. For each child, a non-functional feature selected at random is modified. The modification follows a uniform distribution, varying from -0.01 to 0.01. Then, the new generation is equally composed of parents from the previous generation, and of its offspring. The fact of having members of the previous generation prevents the deterioration of the ability of the overall population to deceive the Intrusion Detection System.

We stop the hybrid genetic algorithm after 500 generations, or after 50 generations without improvement. These numbers were found to be experimentally sufficient for successfully training the four different IDS.

 ISBN: 978-1-61208-862-4

Algorithm 4: Pseudocode of the used local search algorithm

```
Function score(sol: array): float is
    score = 1 - max(discriminator(sol),classifier(sol));
    return score;
end
Input: Population of solutions (solutions), Discriminator,
       Classifier, functional features (func_feat)
Output: Optimized population of solutions
for sol in solutions do
    /* For each solution in the population,
       we slightly modify all the non
       functional characteristics to find
       the best solution in the neighborhood
       */
    for characteristic in sol do
        if characteristic not in func_feat then
            modif = -0.01;
            current_value = characteristic;
            best_value = characteristic;
            best_score = score(sol);
            /* We test all modifications from
               -0.01 to 0.01                  */
            while modif < 0.01 do
                modif = modif + 0.001;
                characteristic = current_value + modif ;
                score = score(sol);
                if score > best_score then
                    best_value = characteristic;
                    best_score = score;
                end
            end
            characteristic = best_value;
        end
    end
end
return solutions
```

This population-based approach makes the solution pool iteratively evolve to better evade the detection system, and therefore generates a wide variety of adversarial attacks.

Step 4: IDS training

In this final step, we aim to retrain the detection system for it to take the generated attacks into account. We train the IDS with:

- All the attacks generated by the hybrid algorithm during its run at step 3.
- The trained GAN generated attacks from the training set.
- Examples from the original training set.

C. Execution of SIGMA

We executed SIGMA on the CalculQuebec Cloud service with the following computing resource: 15 X Intel Xeon @2,5Ghz, 128Go RAM, 10 core, 8 X Nvidia K20-GK110 GPU.

The Pytorch module was used to implement all the neural networks.

In Table IV, we present all the parameters used to train the Neural Networks.

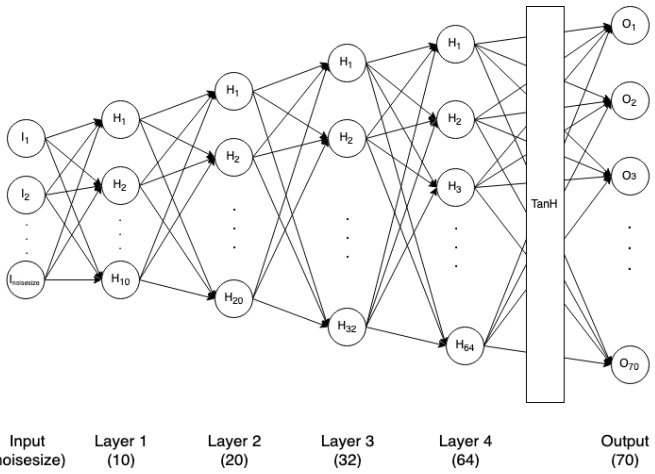


Figure 3. Architecture of the Generator.

TABLE IV
TRAINING PARAMETERS FOR OUR GENERATIVE ADVERSARIAL NETWORK, AND FOR NEURAL NETWORKS USED AS IDS.

Number of training epochs	30
Batch size	64
Learning rate	0.01
Loss function	$L1(\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}) = \sum_i \lvert u_i - v_i \rvert$
Optimizer	Adam

D. Research questions

To evaluate the effectiveness of SIGMA at improving the effectiveness of IDSs, we formulate the following two research questions:

- (RQ1) *To what extent SIGMA can generate adversarial attacks able to deceive trained classifiers, acting as Intrusion Detection System?*
  This research question aims to assess the effectiveness of SIGMA at generating meaningful adversarial attack queries.
- (RQ2) *To what extent is the effectiveness of IDS improved using SIGMA?*
  This research question aims to examine if through the successive re-training steps of SIGMA, IDSes are successfully improved.

In the following, we describe the approach followed to answer RQ1, RQ2.

For RQ1, we use four different classification algorithms as IDS: Neural Network, Random Forest, Support Vector Machine and a Naive Bayes Classifier. We generate attacks against each of the IDS for all four attacks datasets (DOS, DDOS, Bruteforce, Infiltration) by using a GAN, trained with the methodology described above.

We compute the score of each of the detection systems for the GAN generated attacks, and therefore assess if SIGMA is able to deceive standard classification algorithms acting as IDS.

 ISBN: 978-1-61208-862-4

For RQ2, we use a more complex intrusion detection system. We build an IDS consisting of two classifiers: an attack classifier, and a discriminator. The attack classifier is trained with the entries from the original dataset, whereas the discriminator is trained with both regular attacks from the dataset as well as with generated entries to classify the input as a generated attack or as regular traffic. Traffic is first analyzed by the discriminator to determine whether it is an adversarial instance or real traffic. If the input is labeled as real traffic, it then comes through the attack classifier whose role is to recognize attacks. This architecture prevents from training the classifier with the adversarial examples, which could lead to a loss of performance for previously seen regular attacks because of overfitting to adversarial instances. It consists of a simple adaptation of the GAN discriminator to detect both generated instances and attacks from the dataset. Therefore, since the goal of the discriminator is to identify generated instances, it will be the part of the IDS trained with the SIGMA generated attacks.

The overall process of the Intrusion Detection System studied is detailed on Figure 4.

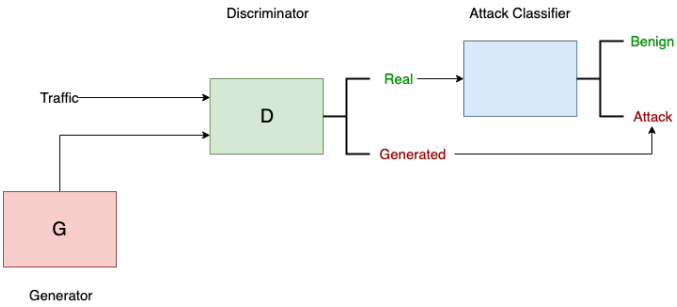


Figure 4. Diagram of the Intrusion detection system.

As attack classifier and discriminator, we used the same algorithms as for RQ1: Neural Network, Random Forest, Support Vector Machine (SVM), and a Naive Bayes Classifier.

We study the largest dataset of the four (the DOS attacks dataset). For each round of improvement of SIGMA, we compute the score of the IDS.

To measure the performance of SIGMA, we compare our strengthened model to a baseline, in which the discriminator is trained only with GAN generated instances. We also verify that metaheuristics alone are not enough to train our system against generated adversarial attacks by comparing the model strengthened by SIGMA with a model trained only with the metaheuristics attacks, and submitting it to GAN generated attacks.

We will judge the quality of the reinforcement by:

- The speed of convergence of the detection rate.
- The value of the limit of the detection rate.
- The overall performance of the model for all iterations.

It should be noted that for the first iteration of the algorithm, the discriminator has not yet been trained: the generator is thus only trained against the classifier at the first iteration.

E. Results of the Evaluation of SIGMA

In this section we present the answers to our two research questions that aim to evaluate SIGMA.

RQ1: To what extent SIGMA can generate adversarial attacks able to deceive trained classifiers, acting as Intrusion Detection System?

The results of the detection of normal and generated attacks are presented in Table V, and on Figure 5.

All four classifiers in our study (Neural Network, Random Forest, SVM, Naive Bayes) have good results in classifying standard entries of the datasets. Even though our classifiers are standard machine learning algorithms, they are sufficient to obtain high accuracy, with the Random Forest algorithm performing with the best results with an overall 99,9% accuracy, followed by the Support Vector Machine with 97,1%. In fact, those two algorithms have often been used in intrusion detection thanks to their good performances [28].

However, the generated attacks detection rates is significantly low for all classifiers with most type of attacks. Both the Random Forest and the Naive Bayes are utterly unable to detect the GAN generated adversarial attacks. The neural network and the SVM are the most resilient classifiers, but the generator still manages to deceive our IDS with over a 90% evasion rate for the DOS, Bruteforce and Infiltration attacks.

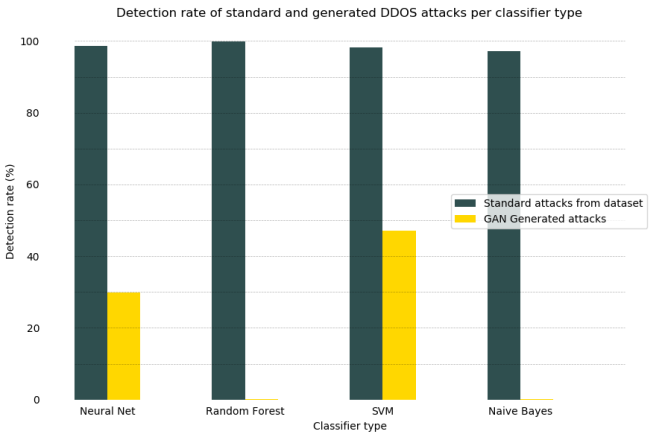


Figure 5. Detection scores per classifier with test set attacks and generated attacks for the DDOS dataset.

The results show very good performance of the Generative Adversarial Network for all different types of attacks. It is therefore possible to generate attacks able to fool Machine Learning based classifiers for all four types of attacks.

RQ2: To what extent is the effectiveness of IDS improved using SIGMA?

We compared the evolution of our model trained with the hybrid local-search-genetic reinforcement and adversarial attacks with a model trained only with adversarial attacks.

The results are presented in Table VI and Figure 6.

First, we notice that both models with the SVM and the Naive Bayes as classifiers only need one step to detect adversarial attacks: those two classifiers are the most able to generalize from the previously seen data. The generated attacks detection rate converges after only one iteration for both the strengthened and the standard model.

TABLE V
DETECTION RATES FOR TEST ATTACKS FROM THE DATASET AND GAN GENERATED INSTANCES.

Classifier type	Neural Net		Random Forest		SVM		Naive Bayes	
Type of attack	Normal	Adversarial	Normal	Adversarial	Normal	Adversarial	Normal	Adversarial
DOS	94,9%	0%	99,8%	0%	97,6%	0%	95,9%	0%
DDOS	98,6%	29,9%	99,9%	0%	98,3%	47,1%	97,1%	0%
Bruteforce	95,6%	1,9%	99,9%	0%	96,3%	0%	97,8%	0%
Infiltration	95,4%	5,8%	100%	0%	96,2%	4,1%	97,4%	0%

TABLE VI
EVOLUTION OF THE DETECTION RATES OF ADVERSARIAL ATTACKS FOR OUR MODEL

Classifier type	Neural Net		Random Forest		SVM		Naive Bayes	
Iteration number	Normal	Reinforced	Normal	Reinforced	Normal	Reinforced	Normal	Reinforced
1	0%	0%	0%	0%	0%	0%	0%	0%
2	6,3%	0%	51%	0%	100%	100%	100%	100%
3	49%	100%	99%	100%	100%	100%	100%	100%
4	100%	100%	18%	100%	100%	100%	100%	100%
5	68%	100%	100%	100%	100%	100%	100%	100%
6	100%	100%	100%	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%	100%	100%	100%

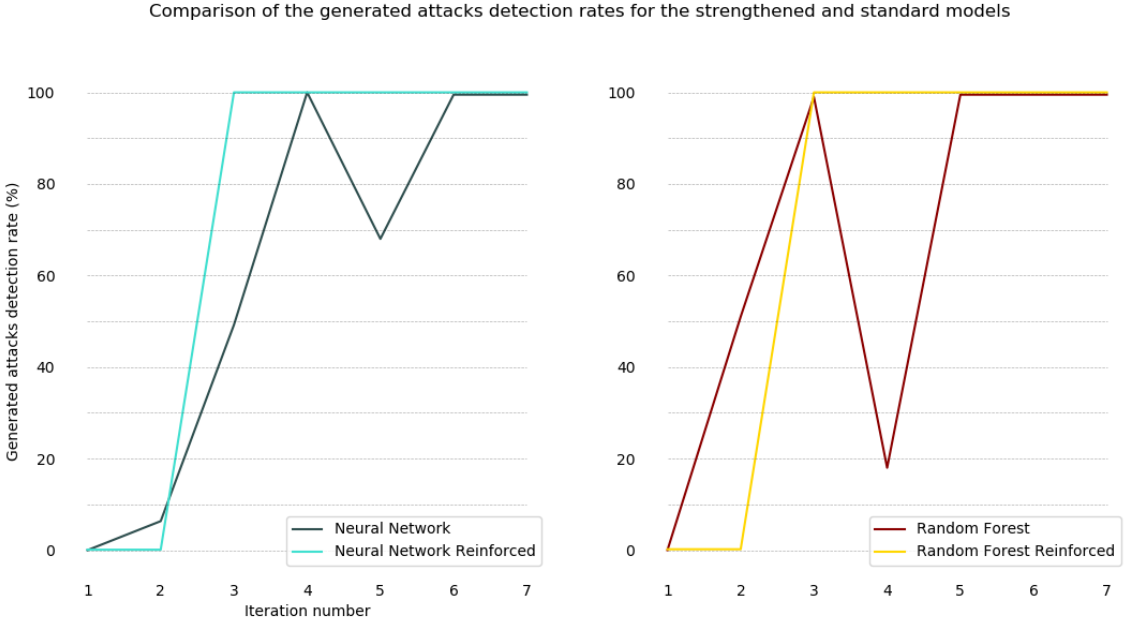


Figure 6. Time evolution of our reinforced model with two different classifiers.

The multi-layer Neural Network and the Random Forest standard models both take time to converge to a 100% generated attacks detection rate: 6 iterations for the model with the Neural Network as classifier, 5 iterations for the Random Forest model. Furthermore, we also note that both models suffered from overfitting: their performance increased (until iteration 4 and 3 respectively) before dropping significantly by 32% and 81%.

The SIGMA method improved the models' results: as we can see, the strengthened model converged faster than the standard model to a 100% detection rate for both the Neural Network and the Random Forest classifiers; the reinforced versions took only two iterations to detect all adversarial instances, that is to say respectively four and three iterations less. As the other two classifiers, namely the SVM and the Naive Bayes classifier, detected all attacks from iteration 2, the reinforcement method did not affect their performance.

Furthermore, we can observe that the SIGMA method prevented the Neural Network and the Random Forest model from overfitting to generated attacks, therefore preventing a performance drop of the algorithm. The combination of the metaheuristic algorithm and the Generative Adversarial Network permitted to generate a sufficiently wide variety of attacks; avoiding fitting closely to previously seen attacks.

Table VII presents the results of models trained only from the Metaheuristics generated attacks. From these results, we can also conclude that Metaheuristics alone are not sufficient to train an IDS against generated adversarial attacks: every classifier, except the Support Vector Machine, was utterly unable to detect any instance generated by our Wasserstein

 ISBN: 978-1-61208-862-4

TABLE VII
EVOLUTION OF THE DETECTION RATE OF ADVERSARIAL ATTACKS FOR A MODEL TRAINED ONLY WITH METAHEURISTICS GENERATED ATTACKS.

Iteration	Neural Net	Random Forest	SVM	Naive Bayes
1	0	0	0	0
2	0	0	0	0
3	0	0	40.1	0
4	0	0	100	0
5	0	0	58.1	0
6	0	0	0.8	0
7	0	0	52.9	0

GAN. The SVM stands out from the other classifiers thanks to its ability to generalize, but fails at consistently detecting GAN generated attacks.

We can conclude that the attacks generated by the Metaheuristics algorithm complement the ones generated by the Generative Adversarial Networks, as the Metaheuristics algorithm alone was not enough to successfully train the IDS.

VI. THREATS TO VALIDITY

This section discusses the threats to validity of our study following common guidelines for empirical studies [6]

Construct validity threats concern the relation between theory and observation. This is mainly due to possible mistakes in the generation of attacks. Even though we kept the functional features of real attack untouched for our generated attacks, we can not guarantee that the generated attacks metrics are indeed plausible attacks.

Internal validity threats concern the selection of tools and analysis methods. We split the dataset into a training and a test set in order to ensure the validity of our results. This prevents having a biased evaluation of our model. As the aim of the method was to try to detect as many generated attacks as possible, we chose to study the generated attacks detection rate as a metric to gauge the quality of the strengthening.

Reliability validity threats concern the possibility to replicate our study. All the tools used in this study are open-source.

Conclusion validity threats concern the relation between treatment and the outcome. We paid attention to not make too broad statements about the performances of our model.

External validity threats concern the possibility to generalize our results. The results of the SIGMA method have to be interpreted carefully, as they may depend on the dataset used to run the experiment and on the used Intrusion Detection System. The iterative strengthening method has only been studied for DOS attacks of the CICIDS 2017 dataset [22]. We used four different classifiers acting as IDS, and were able to significantly improve the results of two of the four IDS. We therefore suggest that our results can be generalized to other detection systems and other datasets.

VII. IMPLICATION FOR PRACTITIONERS AND THE INDUSTRY

Artificial Intelligence is a really powerful tool that could and will be used in future cybersecurity systems: IBM's Watson is one of the illustrations of the application of Machine Learning in this field. Nonetheless, this work illustrated possible vulnerabilities of such systems as Artificial Intelligence can also be leveraged by attackers to disrupt detection systems.

Generative Adversarial Networks can be used to forge almost undetectable adversarial attacks for systems that have not already faced such attacks. Our method confronted our studied Intrusion Detection Systems with attacks generated with both GANs and Metaheuristics in order to improve the systems resilience, as our analysis has shown that the more attacks the system faces, the more it will be able to efficiently generalize to other potential attacks.

Repetitively training an IDS with generated attacks is a way to anticipate for every possible generative scheme that could target the system. By doing so, our method SIGMA is able to detect all the attacks generated by our GAN, thus preventing future intrusion by adversarial generated attacks.

These methods should be applied to any AI-based cybersecurity system in the industry to preemptively confront them to new types attacks, therefore preventing them from possible threats.

VIII. CONCLUSION

The novel ability to use Machine Learning techniques to generate adversarial attacks requires the development of a robust IDS able to detect unusual behaviors. Generative Adversarial Networks are both a terrible weapon for detection systems, and an incredible opportunity to preemptively strengthening IDSs against adversarial attacks.

We have shown experimentally that it is possible to effectively evade intrusion detection classifiers with Generative adversarial networks. We demonstrated the possibility to forge undetected adversarial attacks with GANs against four standard Machine Learning algorithms acting as IDS, with the generated attacks detection rates dropping near 0% for most of them.

To prevent adversarial generated attacks, we presented in this paper a method SIGMA, to improve the robustness of IDS. This method is based on the iterative generation of attacks by a Machine Learning Generative algorithm and Metaheuristics. We have shown that applying this method to Machine Learning based IDS can speed up the convergence of the generated attacks detection rate, and prevent overfitting to previously seen generated attacks.

Our model may help design Intrusion detection systems robust against recurrent generative attacks and improve the security of Machine Learning systems.

Further considerations are the explorations of other more complex detection algorithms, such as Recurrent Neural Networks, the application of the SIGMA method to other datasets and the design of a distributed detection system robust to adversarial attacks.

REFERENCES

[1] J. Lopez, R. Roman, Jianying Zhou. "On the features and challenges of security and privacy in distributed Internet of things," *Computer Networks*, Vol. 57, no. 10, pp. 2266-2279, 2013.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets," In *Advances in neural information processing systems*, pp. 2672-2680, 2014.

[3] J. Sadeeq, "Automatic generation of tests to exploit XML injection vulnerabilities in web applications," IEEE Transactions on Software Engineering, Vol. 45, no. 4, pp. 335-362, 2017.
[4] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas "DDoS in the IoT: Mirai and other botnets," In *Computer*, Vol. 50, no. 7, pp. 80-84, 2017.
[5] C. Tsai, Y. Hsu, C. Lin, W. Lin, "Intrusion detection by machine learning: A review," In Expert Systems with Applications, Vol. 36, no. 10, pp. 11994-12000, 2009.
[6] R. K. Yin "Case Study Research: Design and Methods"Third Edition,3rd ed. SAGE Publications, 2002.
[7] Z. Lin, Y. Shi, Z. Xue, "IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection," *arXiv e-prints*, 2018.
[8] W. Hu, and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.
[9] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, "Synthesizing robust adversarial examples," *arXiv preprint arXiv:1707.07397.*, 2017.
[10] M. Cordy, S. Muller, M. Papadakis, Y. Le Traon, "Search-based test and improvement of machine-learning-based anomaly detection systems," In Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 158-168, 2019
[11] H. Kayacik, A. Zincir-Heywood, M. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," In *Proceedings of the third annual conference on privacy, security and trust*, 2005.
[12] S. Prabavathy, K. Sundarakantham, S.M. Shalinie "Design of cognitive fog computing for intrusion detection in internet of things," Journal of Communications and Networks, Vol. 20, no. 3, pp. 291-298, 2018.
[13] L. Dhanabal, and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, no. 6, pp. 446-452, 2015.
[14] I. Oh, J. Lee, B. Moon, "Hybrid genetic algorithms for feature selection," IEEE Transactions on pattern analysis and machine intelligence, Vol. 26, no. 11, pp. 1424-1437, 2004.
[15] C. Ledig, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2017.
[16] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Transactions on Information and System Security (TISSEC)*, Vol. 3, no. 4, pp. 262-294, 2000.
[17] A. Ferdowsi, and W. Saad, "Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things," arXiv preprint arXiv:1906.00567, 2019
[18] I. Sharafaldin, A. Gharib, A. Lashkari, A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," *Software Networking*, pp. 177-200, 2018.
[19] M. Mitchell, J. Holland, S. Forrest. "When will a genetic algorithm outperform hill climbing," Advances in neural information processing systems. 1994.
[20] N. Ulder, "Genetic local search algorithms for the traveling salesman problem. International Conference on Parallel Problem Solving from Nature," Springer, Berlin, Heidelberg, 1990.
[21] H. Ishibuchi, and M. Tadahiko, "Multi-objective genetic local search algorithm," Proceedings of IEEE international conference on evolutionary computation, IEEE, 1996.
[22] I. Sharafaldin, A. Lashkari, A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," In *ICISSP*, pp. 108-116, 2018.
[23] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, D. Gan "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," IEEE Access, Vol. 6, pp. 3491-3508, 2017
[24] M. Gen, and L. Lin, "Genetic Algorithms," Wiley Encyclopedia of Computer Science and Engineering: pp. 1-15, 2007
[25] M. Arjovsky, C. Soumith, L. Bottou. "Wasserstein GAN," arXiv preprint arXiv:1701.07875, 2017.
[26] H. Cobb, J. Grefenstette, "Genetic algorithms for tracking changing environments," Naval Research Lab Washington DC, 1993.
[27] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, D. Song, "Robust physical-world attacks on deep learning visual," classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1625-1634, 2018
[28] M. Hasan, A. Mehedi, "Support vector machine and random forest modeling for intrusion detection system (IDS)," Journal of Intelligent Learning Systems and Applications, Vol. 6, no. 1, pp. 45-52, 2014

 ISBN: 978-1-61208-862-4

Methods to Prevent Registration Using Fake Face Images

Luis Cárabe
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
Madrid, Spain
e-mail: luiscarabe@gmail.com

Eduardo Cermeño
Research Department
Vaelsys
Madrid, Spain
e-mail: eduardo.cm@vaelsys.com

***Abstract*—Face identification is increasingly being used to register and access specific applications and online services. This opens up new possibilities for malicious attacks, such as users registering multiple times with different images or impersonating other users. Morphing is often the preferred method for these attacks as it allows the physical features of a subject to be progressively modified to resemble another subject. Publications focus on impersonating this other person, usually someone who is allowed access to a restricted area or software app. However, there is no such list of authorized people in many other applications, just a blacklist of people who cannot enter, log in, or register again. In such cases, the morphing target person is not relevant as the criminal's main objective is to minimize the probability of being detected. We present a comparison of the identification rate and behavior of 5 recognizers (Eigenfaces, Fisherfaces, Local Binary Patterns Histograms, Scale-invariant Feature Transform, and FaceNet) against morphing attacks. We also show the performance that a morphing detector could achieve. We prove that the use of FaceNet along with a morphing detector is an optimal resource to maintain a high level of security, identification rate, and attack detection.**

***Index Terms*—Access control; biometrics; deep learning; FaceNet; face recognition; identification; morphing; security; spoofing attack.**

I. Introduction

Face recognition is gaining momentum. Continuous improvements in this well-known research field [1][2][4][5][12] have led to an increasing number of commercial applications. Many sectors have found this technology the perfect match for their security concerns and requirements. Face recognition is used in a wide range of processes: sign up, log in, ID verification, and more broadly in any application that needs to comply with "Know Your Customer" policies.

Like in any other biometric technology, people have tried to deceive face recognition systems [19]. We can find several approaches in the literature. For instance, using a print of a photograph of a subject might allow someone to impersonate as that subject [19][20]. A well-known technique to try to fool face recognition systems is morphing. Morphing techniques consist of generating intermediate frames between two images to achieve a smooth transition between them. If we use it on two images of different faces, we could get frames that merge features of both faces in one. Depending on the level of morphing being applied, one person will be recognized better than the other. In the context of Automated Border Control (ABC), Ferrara *et al.* [3] studied a way to take advantage of morphing to use only one photo ID to verify two different subjects successfully.

The verification process differs from the identification one because the former is a one-to-one matching with only one possible output: match or mismatch. On the contrary, the latter is a one-to-many matching where an image is presented to a face recognizer that compares it against all the stored subjects in its database and outputs the closest match or a top matches list. When morphing an original subject's image to attack a verification system, it is necessary to care about the person's identity recognized by the face recognition algorithm as it must be the target subject. Whereas in the attack to a face identification system, we only need to make sure that the original subject is not identified correctly, it is not relevant who the system thinks the image belongs to, as far as it is not the original subject. This increases the chances of a successful attack because the attacker can reduce the morphing level applied. It is not required to make it look like somebody, but change the image enough to make the face recognition system fail.

In this work, we study the behavior of different face recognition techniques with morphed images. Our aim is to find the most robust one, considering robustness as the quality of requiring a higher amount of morphing alteration to misclassify a subject. We resort to morphing detectors, algorithms designed to detect whether an image is the result of a morphing process and if they can, therefore, be used to endorse a face recognition algorithm against morphing attacks. Furthermore, we analyze the value of implementing a morphing detector along with the face identification algorithm to build a stronger solution that can be used for registration processes or similar ones.

In Section II, we present a brief review of past spoofing attacks to face recognition algorithms and spoofing detection methods. In Section III, we describe the morphing, face recognition, and morphing detection methods used in our study. In Section IV, we describe the scenario of our experiments and the implementation of the methods and database used. In sections V and VI, we present the results of the experiments and their discussion. Finally, in Section VII, we

 ISBN: 978-1-61208-862-4

make conclusions about the findings of our experiments.

II. Related work

We have divided the section into two subsections spoofing attacks and spoofing detection.

A. Spoofing attacks

Spoofing attacks can be undertaken under different approaches. Hadid *et al.* [19] and Mohammadi et al. [20] explore several databases with *presentation attacks*. These attacks consist of showing a printed image (or printed mask) to a camera with facial recognition software to fool it. Apart from this, Ferrara *et al.* [21] study the effects of geometric distortions (barrel distortion, vertical contraction, and extension) and digital beautification on face recognition accuracy. Other digital manipulation techniques can be very harmful, e.g., face synthesis, attribute manipulation, and identity or expression swap [22].

Ferrara *et al.* [3] were the first to present a successful morphing attack in a simulation of an ABC control, using two commercial face recognition software tools. They manually created morphed images to verify the two contributing subjects with the same photo. They were able to achieve that for eleven pairs of subjects in both face verification tools. Ferrara *et al.* [21] expand the experiment proving that human experts (border guard group) and non-experts, in most cases, do not detect morphed images. However, Robertson *et al.* [23] reveal that although the attack may go more unnoticed in untrained subjects, when the subjects receive morphing training, they tend to detect morphing with higher probability. Wandzik *et al.* [24] and Scherhag *et al.* [25] present more examples of verification attacks. In the first one, they carried out the experiment using FaceNet, utilizing more than 3000 pairs with 22 morphed images between each pair, working with triplets of images (impostor-accomplice-morphing). In the second one, experiments were conducted to prove face verification's vulnerability both with printed and scanned images.

B. Spoofing detection

Galbally *et al.* [33] present a survey on hardware-level and software-level methods to detect presentation attacks in images and videos. Hadiprakoso *et al.* [34] and Wu *et al.* [35] present more recent studies. In the first one, they combine a Convolutional Neural Network (CNN) analysis with face liveness detection module to be able to detect static and dynamic attacks, such as masks, photos, or video replays. In the latter, they compare the performance of some methods to detect spoofing attacks.

Focusing on morphing, the first detector was presented by Raghavendra *et al.* [36], which successfully verified all the 450 morphed face images from a database. Additional approaches can be found in [37]–[40]. In order to detect morphing successfully, the authors use different techniques, such as Fourier spectrum of sensor pattern noise, Local Binary Pattern (LBP), or a *demorphing* process. Scherhag *et al.* [16] and Raja *et al.* [41] present a review of these methods, along with others.

III. Methods

We have divided this section into three subsections: Morphing attacks, Face Recognition and Morphing Detection.

A. Morphing attacks

The first method used in our study is the morphing attack. A morphing attack is the alteration of a subject's portrait using morphing techniques leading to his misidentification.

Most of the morphing methods found in the literature [16] are based on Delaunay triangulation [18][28]. It includes three stages: feature specification, warping, and blending. In the first step, a correspondence between the two images is created by determining the face key landmarks (eyes, mouth, nose, face contour, etc) either manually or automatically (using software). Then, a Delaunay triangulation is applied using the landmarks as vertices for the non-overlapping triangles. During warping [15], the corresponding triangles of both images suffer a geometrical transformation in order to be aligned. The last step requires to merge each pixel's color value, where a linear blending is applied.

At the warping and blending steps of the process, a parameter α is taken into account. In the case of warping, it conditions how much each position of each face's landmarks contributes to the morphed image. If $\alpha = 0$, only the first image's landmarks are taken into account. If $\alpha = 1$, only the landmarks of the second image are considered. The in-between values achieve a linear combination of the positions of the landmarks of both contributing images. That is to say, if l_r represents the landmark positions of the resulting image and $l_{0,1}$ the landmark positions of the first and second images:

$$l_r = (1 - \alpha)l_0 + \alpha l_1.$$

The blending step has a similar behavior. The color of all the correlated pixels are combined using a linear transformation. $\alpha = 0$ only considers the first image and $\alpha = 1$ the second. If c_r represents the color of the pixels of the resulting image and $c_{0,1}$ the colors of the pixels of the first and second images:

$$c_r = (1 - \alpha)c_0 + \alpha c_1.$$

α is used as a quantifier of the morphing process. For example, a morphing process (amount) of 5% means that $\alpha = 0.05$. The first subject of the pair will contribute to the final image by 95% in both the landmarks' position and the pixels' value. The second subject will contribute with the remaining 5%.

B. Face recognition

A key component of any user registration system using faces is the face recognition algorithm. There are different approaches in the literature that can be classified into four families: holistic, local, hybrid and deep learning [4][5]. The local approach classifies according to specific facial features, whereas the holistic approach considers the whole face as a unit. The hybrid approach combines both techniques. Many recent advances have been made in the deep learning approach, using CNNs that offer better speed and accuracy.

 ISBN: 978-1-61208-862-4

We have selected the more promising ones with care to include at least one from each category (except hybrid, due to its high complexity [4]).

1) Holistic: In the holistic approach, we have selected Eigenfaces [6] and Fisherfaces [7]. Eigenfaces is based on the Principal Component Analysis (PCA) technique. It tries to reduce the dimensionality of the data space by projecting the face images into a subspace called feature space. It also tries to find a basis of that subspace for the dataset. This is achieved by finding the eigenvectors (referred to as eigenfaces) of the covariance matrix of the set of faces. The resulting eigenfaces form the basis of the feature space. To identify faces, the testing image is projected into this subspace using a linear combination of the eigenfaces basis.

Fisherfaces has the same objective as Eigenfaces: reduces dimensionality. Nevertheless, instead of using only an unsupervised technique (PCA), it also uses Linear Discriminative Analysis (LDA), which works with a supervised learning technique. The LDA technique attempts to model the difference between two distinct classes (individuals). That is, by using scatter matrices, it tries to find a linear combination of features that separate two or more classes. This method achieves excellent results even with severe illumination changes.

2) Local: In the local category, we have chosen Local Binary Patterns Histograms (LBPH) [30] and Scale-invariant Feature Transform (SIFT) [9]. The LBPH algorithm works by creating histograms of the binary patterns extracted by LBP [8]. Those binary patterns are obtained as follows: First, the image (in gray scale) is divided into 3x3 pixel regions. Then, for each region, the central pixel's value is taken as a reference, which will act as a threshold for the neighboring pixels. We look at the value of each pixel in the grid, if it is above the threshold (the value of the central pixel), it is assigned a 1. If it is below, a 0. Then the binary values are concatenated, and the result is assigned to the central pixel. To classify an image, it finds its closest histogram from the training database.

SIFT generates image features that are highly distinctive and invariant to certain transformations, such as translation, scaling, and rotation. To obtain those features, the algorithm first tests different image scales, looking for invariant key points. Then, among all the key points obtained, the most stable ones are selected. Meaning that those with the highest sensitivity to noise (points with low contrast) and those located on edges are discarded. Later, the algorithm assigns one or more orientations to each key point, based on the directions of the local gradient of the image, achieving rotation invariance. Finally, each key point is assigned a feature descriptor, ensuring that they are highly distinctive and invariant to lighting changes.

3) Deep learning: In the deep learning group, we have chosen FaceNet [10]. It uses convolutional layers to create a 128-dimensional embedding for every image. The FaceNet model is trained with a Triplet Loss technique. It selects combinations of three images: two images from the same subject (one image is called the anchor and the other one the positive input), and another image from a different subject (negative input). The Triplet Loss tries to minimize the anchor's embeddings distance with the positive input and maximize it with the negative input. Once the model is trained, FaceNet can compute the 128-dimensional embedding for each image in our training database. In the face identification process, FaceNet will return the subject whose embeddings are most similar to those obtained in the testing image.

As seen in [4][5][7][9][26], all of these face recognition techniques have been well studied and have good performance when using frontal views of faces.

C. Morphing detection

Apart from observing how the recognizers behave against morphing, it may be interesting to consider a morphing detector capable of classifying images as morphed or bonafide (unaltered).

We have selected a morphing detector that operates in Single Image Morphing Attack Detection scenarios (S-MAD). It refers to algorithms that only analyze one photograph to check its morphing. In contrast, Differential Morphing Attack Detection (D-MAD) groups algorithms that analyze a pair of images, one of them being a trusted unaltered photograph that the algorithm uses to verify the morphing on the other image. Our scenario falls into the first category since we only provide one image (the one that the subject uses to access) to the detector to get a morphing verification.

IV. Experiments

The experiments found in the literature do not take into consideration morphing attacks against face identification. We wish to study the approach that performs better against these attacks from two perspectives: a basic one, where we analyze the performance of the recognizers in correct subject identification, and, an advanced one where we study the ability to detect fraudulent registrations.

In the first case, from our point of view, good performance means that the algorithm can correctly identify the original subject in images that have been morphed. Since morphing is an incremental process, the most robust algorithm should be the one requiring the highest amount of morphing to force its failure. Therefore, the selection criteria should be based on the first frame where the face recognition algorithm does not recognize the original subject but another (either the target subject or any other person). The higher the alteration percentage required to avoid the correct identification by the recognizer, the more robust it has to be considered.

In our study, the original image (first contributing subject) is morphed into 100 images with $n\%$ morphing ($n \in \{1, .., 100\}$). We consider that the original image has been morphed 0%, the target image (second contributing subject) has been morphed 100%, and any other image in between has $n\%$ ($n \in \{1, .., 99\}$) as the amount of morphing.

Regarding the advanced scenario, we aim to study which recognizer is better to prevent multiple registrations of the same subject. For this purpose, we assume that a recognizer will accept a person as a new record when it has 0 identified

 ISBN: 978-1-61208-862-4

subjects with confidence above a threshold. If there are subjects identified, the person will be rejected. This gives us two rates. The False Acceptance Rate (FAR, impostors being able to register again), and the False Rejection Rate (FRR, genuine subjects not being able to be registered for the first time). As the FRR decreases and the FAR grows with the threshold, the best performing recognizer will be the one whose FRR decreases earliest and whose FAR grows latest.

Moreover, we are going to study the performance of the morphing detector applied to both perspectives.

A. Implementation

1) Morphing: For the morphing implementation, we have used the Python code presented by Patel [27], based on OpenCV functions [17][18]. To find the face landmarks, it uses Dlib's facial landmark detector [29]. Then, as we have seen, those landmarks are employed as vertices of the Delaunay triangles. Using the corresponding triangles, it performs warping and blending to obtain all the intermediate frames.

TABLE I
CLAIMED ACCURACY OF THE SELECTED RECOGNIZERS.

Category	Recognizer	Accuracy (database)
Holistic	Eigenfaces	97.5% (ORL) [26]
	Fisherfaces	92.7% (Yale) [7]
Local	LBPH	76% (FERET) [30]
	SIFT	84.03% (BANCA) [9]
Deep Learning	FaceNet	99.63% (LFW) [10]

2) Face recognizers: Table I shows the accuracy claimed for all the selected recognizers. For the first three face recognition algorithms (Eigenfaces, Fisherfaces, and LBPH), we have employed a Python implementation of Raja [31] that uses the Face library of OpenCV to cover the feature extraction and classification. Besides, a Haar cascade classifier is used for face detection. Slightly modifying the previous implementation, we have gotten a SIFT deployment, using the xfeatures2d OpenCV class to perform the SIFT feature extraction and the Scikit-learn library for classification using a Support Vector Machine (SVM). Moreover, we have used a Tensorflow implementation of FaceNet [32] written in Python. It uses a pre-trained model that employs VGGFace2 as the training dataset and the Inception-ResNet-v1 architecture, achieving an accuracy with the verification problem in the Labeled Faces in the Wild database (LFW) [11] of 99.65+-0.00252%. It also uses an SVM for classification.

The testing subjects are to be included in all the recognizers training database, what is known as closed-set identification. In order to get similar behavior in all the implementations, we introduced small changes in the code files. Every algorithm used can output the top 5 identification matches of the face presented. The parameters of the Haar cascade classifier that worked better with our database were *scaleFactor=1.001, minNeighbors=2, minSize=(90,90), outputRejectLevels=True*. Regarding the SVM used on SIFT, we have employed the settings *kernel="poly", C=10, gamma=0.0001*. We have left all the other configurations according to the original sources.

3) Morphing detector: Regarding the morphing detector, we have tried the algorithms of [37]–[39]. The one that had the best performance and integration in our scenario has been the detector presented by Raghavendra *et al.* [38], which has better results than other state-of-the-art alternatives. Although it is designed to detect morphing in printed-scanned photographs, it achieves excellent detection results in our context (see Figure 2), and therefore, it is the morphing detector used.

B. Database

1) Basic scenario: We recommend that face recognition algorithms should be trained with a database composed of N subjects, with a number of photos per subject between 5 and 20. This quantity helps to avoid imbalanced data and biased results. To test the morphing, we have chosen pairs of similar-looking subjects. This should reduce the amount of alteration required to pass from the original image (referred to as A) to the target image (referred to as B).

We have created a database based on LFW [11]. As seen in [5], it is a widely used unconstrained database to test state-of-the-art face recognizers. Usually, algorithms struggle with lighting, location, setting, pose, or age variations, as well as occlusions or misalignment [12]–[14]. However, over time, algorithms have improved significantly in this area.

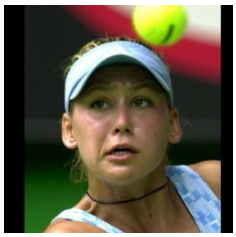
The database has 5749 subjects, but, as stated above, we want only the ones that have between 5 and 20 images each (both numbers included). That filters the database to 366 people with a total number of 3062 images. The Haar cascade face detector does not correctly detect the subject face in 5 of the 3062 images because those images have more than one face present and the wrong face is detected. We deleted those images from the database. The deleted images are *Erika_Harold_0003*, *Hugh_Grant_0008*, *Igor_Ivanov_0014*, *Jean_Charest_0004*, and *Joe_Lieberman_0004*. That implies that Erika Harold now has four images instead of 5, considering this an exception.

TTo determine the pairs of subjects who look more alike, we have used the Similar-looking LFW database (SLLFW) [42], which offers 3000 pairs of similar-looking faces (using the images of LFW). We have picked 25 pairs of images from it, taking into account two factors: first, the individuals must be included in our 366 subjects database; second, once the similar-looking images selected are removed from the training database, the subjects need to have more than five photos to train. Figure 1 shows an example of one selected pair.

Considering all the pairs, there are 49 different images (*Renee_Zellweger_0009* appears twice). The training database of the recognizers consists of $3062 - 5 - 49 = 3008$ images of 366 subjects. In Table II, we provide all the pairs used.

2) Registration scenario: Regarding the advanced perspective, we use the same training database seen in the previous section. For the testing, we need two groups of subjects: impostors and genuine ones. Considering the first case, we have used the 49 different subjects (already registered) seen in Table II. We have randomly morphed them with people not included in the training database (LFW subjects with n images,

 ISBN: 978-1-61208-862-4



(a) Anna_Kournikova_0011.

(b) Jelena_Dokic_0007.

Fig. 1. Similar-looking pair.

TABLE II
SIMILAR-LOOKING PAIRS SELECTED.

No.	Original subject	Target subject
1	Amelia_Vega_0003	Norah_Jones_0015
2	Ana_Guevara_0002	Ian_Thorpe_0006
3	Andy_Roddick_0008	Richard_Virenque_0004
4	Angelina_Jolie_0002	Britney_Spears_0004
5	Anna_Kournikova_0011	Jelena_Dokic_0007
6	Ben_Affleck_0002	Ian_Thorpe_0007
7	Bill_McBride_0010	Jon_Gruden_0002
8	Bill_Simon_0011	Ron_Dittemore_0001
9	Catherine_Zeta-Jones_0001	Salma_Hayek_0001
10	Edmund_Stoiber_0004	John_Snow_0003
11	Eduardo_Duhalde_0006	George_HW_Bush_0005
12	Fidel_Castro_0018	Mohamed_ElBaradei_0003
13	Hillary_Clinton_0010	Renee_Zellweger_0009
14	Howard_Dean_0003	Kevin_Costner_0005
15	James_Blake_0006	Mark_Philippoussis_0003
16	Jason_Kidd_0003	Leonardo_DiCaprio_0003
17	Jean-Pierre_Raffarin_0001	Joschka_Fischer_0012
18	Jimmy_Carter_0006	John_Snow_0004
19	Joan_Laporta_0007	Pierce_Brosnan_0006
20	John_Kerry_0005	Robert_Redford_0002
21	Julianne_Moore_0019	Nancy_Pelosi_0002
22	Kate_Hudson_0008	Mariah_Carey_0006
23	Matthew_Perry_0007	Rubens_Barrichello_0011
24	Mike_Martz_0005	Paul_ONeill_0003
25	Renee_Zellweger_0009	Sheryl_Crow_0001

$n < 5$ *or* $n > 20$), selecting arbitrarily, for each subject, nine morphed images (between 1% and 80% of alteration) and the unaltered image. The impostors database has $49*(9+1) = 490$ images. We have selected 490 unaltered images of different subjects not included in the training database used for the genuine subjects.

3) Morphing detector: To train and test the morphing detector, we have picked the LFW subjects' images not used in the other experiments. We have split the subjects randomly into two groups, one for testing and the other one for training. Due to Matlab memory limitations (we have used Matlab Online to train the model, which provides up to 16 GB of RAM [43]), we have trained the detector using 3000 bonafide (not altered) images from the training group and 3500 morphed images. The morphed images were created randomly using pairs from the subjects included in the training group, covering all percentages between 1 and 99. Analogously, we have tested the detector using 500 bonafide images and 500 morphed images. Figure 2 represents the Receiver Operating Characteristic (ROC) curve obtained, showing the excellent performance achieved.

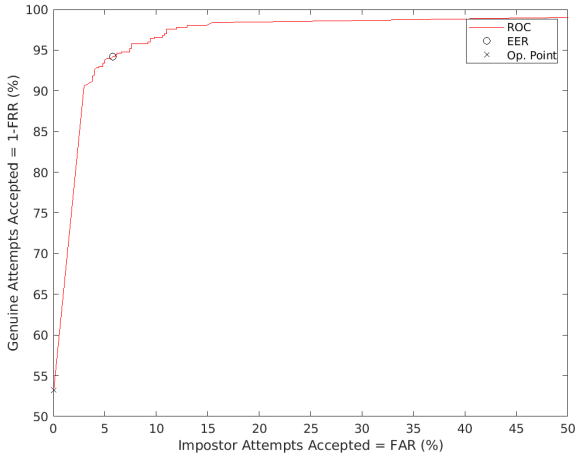


Fig. 2. ROC curve of the morphing detector.

V. RESULTS

Figure 3 shows the face identification algorithms' robustness results in the basic scenario (correct identification). It is divided into three plots. Figure 3a exhibits the face recognizers' comparison analyzing the top 1 identification matches. Figure 3b analyzing the top 3. Figure 3c the top 5. Their x-axes represent the level of morphing in the pairs. 0% morphing symbolizes the unaltered image of the first subject of the pair (original subject), 100% the second subject, and the rest of percentages the in-between morphings. Their y-axes reflect the percentage of couples who still have their original subject identified within the top analyzed for each morphing level.

It has to be observed that the identification percentages rise as we increase the top analyzed. However, the three graphs show similar robustness ranking.

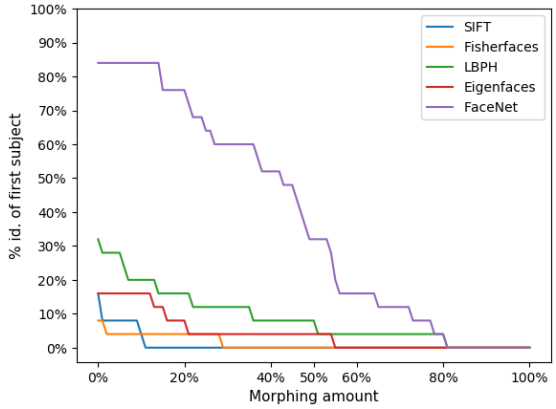
For each face recognizer, we have elaborated a table that shows the average confidence percentages outputted when the original subject is included in the top 1. The first row (*Morph*) shows the most relevant morphing percentages. Rows *1–5* show the first five identified subjects' average confidence.

For each recognizer, confidences have been normalized taking 100% as the best result obtained in our experiments (when the subject is correctly identified), and 0% as the confidence obtained in the last recognition position in images of subjects not included in the training database.

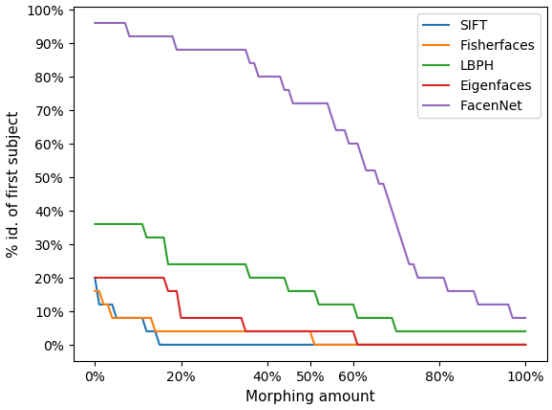
Also, we have included the FAR vs FRR plot of the advanced scenario, with and without using the morphing detector (mor. det.) to filter the accepted subjects. In the first case, we accept a subject (as a new register) when there are no identified subjects above the confidence threshold (x-axis). In case of having a morphing detector, to accept a subject, the previous condition must be met, and the morphing detector must output less than 50% of morphing confidence. Otherwise, the subject will be rejected.

In the following subsections, we describe the performance achieved by each method in both scenarios.

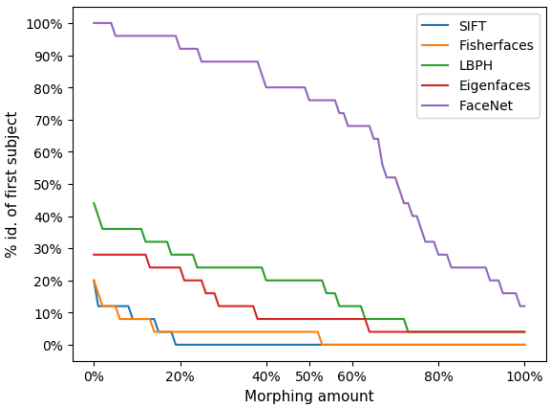
 ISBN: 978-1-61208-862-4



(a) Top 1.



(b) Top 3.



(c) Top 5.

Fig. 3. Percentage of morphed images identified as the original subject for each level of morphing.

A. *FaceNet*

It achieves the best identification scores for each top, with 0% morphing, being 84%, 96%, and 100%, respectively. FaceNet manages to maintain a high identification rate even with a considerable morphing alteration. For instance, at 50% morphing, it achieves 32%, 72% and 76% identification of the original subject for each top. Looking at the top 3 and 5, it even identifies more than 8% of the images with the original subject totally transformed (100% morphing). FaceNet takes the longest time (most significant morphing alteration) to misidentify the original subject.

TABLE III
AVERAGE CONFIDENCE PERCENTAGES OF FACENET WHEN THE ORIGINAL SUBJECT IS INCLUDED IN THE TOP 1.

Morph	**0**	**10**	**20**	**25**	**30**	**40**	**50**	**60**	**70**	**80**
1	48,5	52,6	39,3	45,1	38,8	34,7	28,9	31,1	22,5	20,8
2	19,7	19,6	20,0	18,7	19,9	22,7	21,0	19,9	20,5	19,9
3	15,2	16,0	16,0	13,6	14,7	16,3	15,0	18,8	19,0	16,6
4	13,3	12,1	13,8	12,3	12,9	13,8	13,9	15,4	18,6	16,1
5	12,2	11,4	12,9	11,6	12,2	12,5	12,7	14,5	13,0	13,7

Table III shows the average confidence of FaceNet for different morphing levels. The best result is obtained with an alteration of 10%. The average confidence is computed only with the subjects that were correctly classified in the top 1. The average confidence distance between the first and second place of the top is 16.1%.

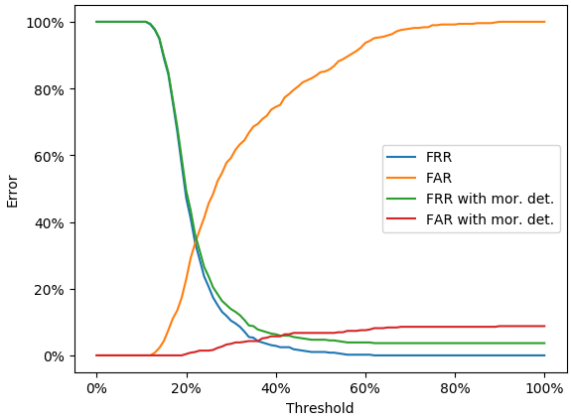


Fig. 4. FAR vs FRR of FaceNet.

Figure 4 shows how the FRR remains below 10% from the 31% threshold. Once the 55% threshold is reached, this error drops to almost 0%. Even with the morphing detector's application (which can cause extra false rejections), it performs well, adding less than 3.7% extra error. On the contrary, the

FAR rises significantly from the 15% threshold, reaching 90% error at 57%. The morphing detector strongly reduces this error, dropping it below 9% for all possible thresholds.

B. LBPH

Its best scores for each top are 32%, 36%, and 44%, respectively. At 50% morphing, it achieves 8%, 16%, and 20% identification of the original subject for each top. Looking at the top 3 and 5, it identifies 4% of the images with the original subject totally transformed (100% morphing). LBPH is the second most robust algorithm, having a distance with FaceNet of more than 50% misidentification in some cases. In general, its recognition rate decreases more slowly than FaceNet, but LBPH is always below it, getting a tie only above 78% morphing in the top 1.

TABLE IV
AVERAGE CONFIDENCE PERCENTAGES OF LBPH WHEN THE ORIGINAL SUBJECT IS INCLUDED IN THE TOP 1.

Morph	0	10	20	30	40	50	60	67	77	79
1	59,2	56,3	64,9	74,1	58,8	61,9	63,3	63,3	44,8	44,3
2	45,0	48,0	51,6	53,6	48,4	53,4	51,5	61,7	37,3	42,3
3	42,5	44,6	48,1	48,4	44,9	52,0	49,9	61,0	27,9	29,8
4	41,2	42,7	45,0	45,5	42,4	47,3	40,0	56,7	26,8	28,5
5	39,4	40,2	41,7	42,2	36,4	46,6	39,6	56,4	16,6	23,7

Table IV shows that its highest confidence peak is 74.1%, obtained with a morphing alteration of 30%. However, the number of individuals used to calculate it is lower than FaceNet since only three were correctly classified, whereas FaceNet classifies fifteen properly with the same amount of morphing. The average confidence distance between the first and second place of the top is 9.8%.

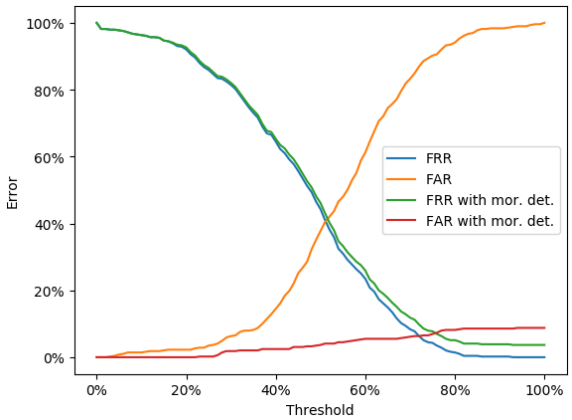


Fig. 5. FAR vs FRR of LBPH.

Figure 5 shows how the FRR remains below 10% from the 70% threshold. Once the 82% threshold is reached, this error drops to almost 0%. The application of the morphing detector adds less than 3.7% extra error. On the contrary, the FAR exceeds 10% from the 37% threshold, reaching 90% error at 75%. The morphing detector strongly reduces this error, dropping it below 9% for all possible thresholds.

C. Eigenfaces

Its best scores for each top are 16%, 20%, and 28%, respectively. At 50% morphing, it achieves 4%, 4%, and 8% identification of the original subject for each top. Looking at the top 5, it identifies 4% of the images with the original subject totally transformed (100% morphing). Eigenfaces takes the third position. In some percentages, it achieves a distance with LBPH of, at most, 16% identification. Although its performance is low, it maintains 8% and 4% identification for a long time. For example, between 38% and 100% morphing in the top 5.

TABLE V
AVERAGE CONFIDENCE PERCENTAGES OF EIGENFACES WHEN THE ORIGINAL SUBJECT IS INCLUDED IN THE TOP 1.

Morph	0	5	10	15	20	30	36	43	46	54
1	77,4	85,8	76,8	70,9	98,5	66,2	92,8	82,2	74,6	87,3
2	69,1	76,3	76,0	69,7	84,8	66,1	78,6	82,1	68,6	81,6
3	65,5	71,4	71,0	68,5	71,6	64,3	73,1	75,4	65,3	79,6
4	61,3	66,1	69,9	64,5	71,1	62,1	73,0	72,8	65,0	78,6
5	58,9	65,5	69,5	63,4	71,1	59,4	72,7	71,8	62,3	78,2

Table V shows that its highest confidence peak is 98.5%, obtained with a morphing alteration of 20%. In this case, only two subjects were correctly classified. The average confidence distance between the first and second place of the top is 6%.

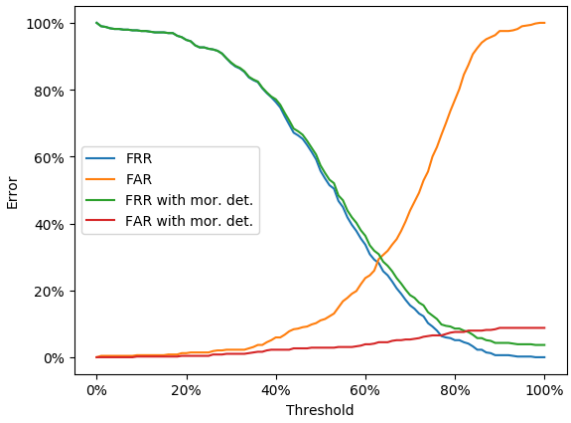


Fig. 6. FAR vs FRR of Eigenfaces.

Figure 6 shows how the FRR remains below 10% from the 75% threshold. Once the 93% threshold is reached, this error drops to almost 0%. The application of the morphing detector adds less than 3.7% extra error. On the contrary, the FAR exceeds 10% from the 49% threshold, reaching 90% error at 84%. The morphing detector strongly reduces this error, dropping it below 9% for all possible thresholds.

D. Fisherfaces

Its best scores for each top are 8%, 16%, and 20%, respectively. At 50% morphing, it achieves 0%, 4% ,and 4% identification of the original subject for each top. Fisherfaces fails to identify any original subject with 100% alteration. As with Eigenfaces, although its performance is low, in some

cases, it manages to maintain a 4% identification rate for a long range, for instance, in 14%–52% morphing in the top 5. However, at all the percentages, it has equal or lower recognition rates than Eigenfaces.

TABLE VI
AVERAGE CONFIDENCE PERCENTAGES OF FISHERFACES WHEN THE ORIGINAL SUBJECT IS INCLUDED IN THE TOP 1.

Morph	0	2	3	5	6	10	15	20	25	28
1	63,2	92,1	68,6	94,8	51,3	97,3	100	88,2	89,8	88,0
2	59,8	84,5	64,3	87,2	51,2	91,5	93,1	84,8	88,8	87,8
3	58,4	80,2	61,3	83,8	49,5	89,5	91,4	83,6	86,0	83,4
4	56,6	79,8	60,1	83,8	49,5	88,9	90,5	80,9	85,4	83,3
5	55,6	79,2	59,6	83,6	49,4	88,4	89,1	80,1	84,0	81,7

Table VI shows that its highest confidence peak is 100%, obtained with a morphing alteration of 15%, with just one person correctly classified. The average confidence distance between the first and second place of the top is 4%.

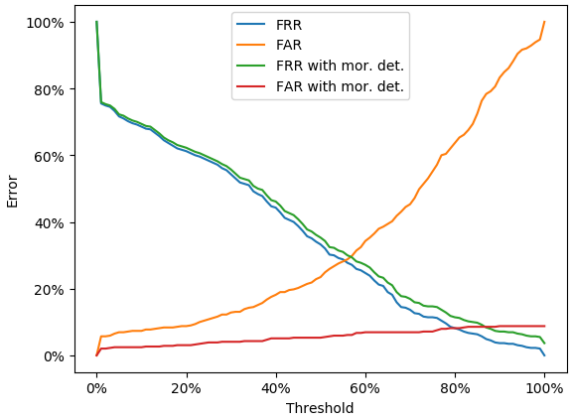


Fig. 7. FAR vs FRR of Fisherfaces.

Figure 7 shows how the FRR remains below 10% from the 78% threshold. This error drops to 0% only when the threshold is 100%. The application of the morphing detector adds less than 3.7% extra error. On the contrary, the FAR exceeds 10% from the 23% threshold, reaching 90% error at 94%. The morphing detector strongly reduces this error, dropping it below 9% for all possible thresholds.

E. SIFT

Its best scores for each top are 16%, 20%, and 20%, respectively. Once we reach 20% morphing, SIFT obtains 0% identification in all cases. Although the values achieved at 0% morphing are better than those obtained with Fisherfaces, SIFT's decrease rate is higher.

Table VII shows that its highest confidence peak is 69.4%, obtained with a morphing alteration of 8%, but only two people are correctly classified in that case. The average confidence distance between the first and second place of the top is 18.7%.

Figure 8 shows how the FRR remains below 10% from the 59% threshold. This error drops to 0% only when the threshold is 100%. The application of the morphing detector

TABLE VII
AVERAGE CONFIDENCE PERCENTAGES OF SIFT WHEN THE ORIGINAL SUBJECT IS INCLUDED IN THE TOP 1.

Morph	0	1	2	3	4	5	6	7	8	10
1	33,4	45,7	43,6	41,6	60,2	42,9	43,7	48,7	69,4	43,9
2	30,8	28,2	28,3	28,1	25,6	30,9	30,2	26,8	27,1	30,2
3	23,8	23,9	23,9	24,7	25,6	26,0	28,6	24,4	27,1	30,2
4	22,7	21,7	21,8	23,6	23,2	26,0	25,4	24,4	19,7	30,2
5	20,6	21,7	21,8	22,5	23,2	18,8	23,0	24,4	19,7	21,9

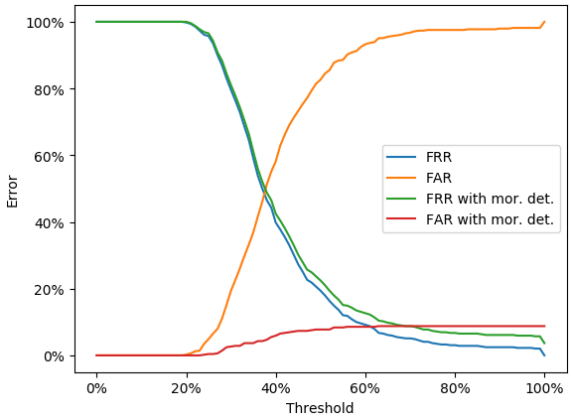


Fig. 8. FAR vs FRR of SIFT.

adds less than 3.7% extra error. On the contrary, the FAR exceeds 10% from the 28% threshold, reaching 90% error at 56%. The morphing detector strongly reduces this error, dropping it below 9% for all possible thresholds.

F. Morphing detector

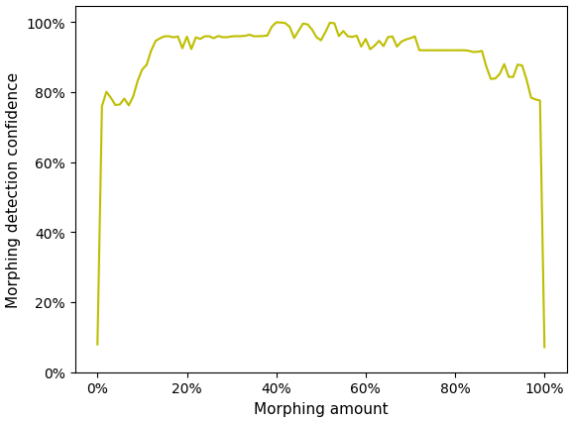


Fig. 9. Average morphing detection confidence.

Figure 9 shows the performance of the morphing detector in the morphed images used in the basic scenario. It displays the average detection rate at every quantity of morphing alteration from 0% to 100% (reflected in the x-axis), computed using all the morphed and unaltered images from the 25 similar-looking pairs.

We can observe that with the non-morphed images (0% and 100%), the detector provides less than 10% confidence. On

the contrary, between 1% and 99% of morphing alteration, it returns an average morphing confidence above 70%. Also, confidence is over 90% in 15%–85% morphing. At some morphing percentages around the maximum alteration (50%), it reaches a confidence level near 100%, proving its excellent performance.

VI. Discussion

In the basic scenario, FaceNet obtains the best performance identifying the in-between morphed images correctly. This means that in the case of a real attack on FaceNet, the attacker would need to significantly alter the image to fool the recognizer. Looking at Table VIII, we can see that analyzing the top 1, the attacker would need a 43% morphing alteration to have more than a 50% chance of the attack being successful. If we analyze the top 3, the required morphing alteration is higher than 66%. Finally, if we analyze the top 5, the alteration needed rises to 71%. FaceNet shows such good results that some attacks will fail even with the original image wholly modified (100% morphing) if we consider top 3 or top 5 lists.

TABLE VIII
Accuracy of FaceNet at different percentages of morphing. Italicized values represent the point at which the accuracy drops below 50%.

Morph	0%	43%	50%	66%	71%	100%
Top 1	84%	*48%*	32%	12%	12%	0%
Top 3	96%	80%	72%	*48%*	32%	8%
Top 5	100%	80%	76%	64%	*48%*	12%

As for the remaining recognizers, the identification rate is much worse, being extremely low in some cases. Our results for facial identification on the LFW database are notably worse than those obtained in verification. This might be expected since, for identification, we work 1 vs. N ($N = 366$ in our database), and regarding verification, we work 1 vs. 1. Thus, as mentioned in [44], the difficulty of identification is related to the number of subjects contained in the database. Some examples are Eigenfaces, in which we have obtained 16% of identification accuracy in contrast with 60.02% of verification accuracy [44], and FaceNet, with 84% and 99.6% of identification and verification accuracy, respectively [44].

The only possible alternative to FaceNet would be LBPH. When dealing with images with a considerable morphing amount (e.g., $> 75\%$), their accuracy is similar, however, LBPH offers greater distances (between the confidence of the first and second position) than with FaceNet. With both recognizers, we get the best confidence distances for 0% morphing, 28.8% for FaceNet, and 14.2% for LBPH.

We have also shown that the morphing detector has an excellent performance, outputting morphing detection confidences above 90% when the alteration is considerable (15%–85%). That would mean that most attacks that require some alteration in order to be successful would very likely be detected.

In the advanced scenario, FaceNet is the recognizer with the best FRR since it is the one that achieves an error below 10% with the lowest threshold (31%). It is followed by SIFT, which needs a threshold of 59% to achieve the same error. However, as we have seen, SIFT has low performance in the basic scenario, so it might not be recommended in a general system.

Eigenfaces is the recognizer with the best FAR since it is the one that achieves an error above 10% with the highest threshold (49%). Nevertheless, as in the case of SIFT, its performance from the basic perspective is poor, so we do not recommend its use in a general system either. Its FAR results are followed by LBPH (37% threshold), which would be a preferable option.

The inclusion of the morphing detector has a significant impact on all recognizers. It causes the FAR to always be below 9% and the FRR to grow at most 3.7%.

As the morphing detector fixes the FAR problem, FaceNet is the best algorithm in either correctly identifying subjects (basic scenario) or registering new subjects (advanced scenario). It is the best performing method in a general system.

VII. Conclusion and future work

If we want to prevent registration using fake face images, the recommended option is FaceNet or, as a second option, LBPH. Our experiments show that these techniques have significantly better results than others like Eigenfaces, Fisherfaces, or SIFT. The difference between FaceNet and any other technique is impressive. With 0% of morphing, only FaceNet presents an accuracy of over 80%. The second option, LBPH, has an accuracy below 35%, while the rest of the techniques cannot reach 20%. Even with a small amount of morphing, less than 20%, the error of more classic techniques jumps over 90%.

FaceNet is a robust technique against morphing attacks when used in combination with an S-MAD morphing detector. Both the False Rejection Rate and the False Acceptance Rate are lower than 6% when a threshold of 41% is used. This threshold can be recommended for most cases since FaceNet recognizes most attackers using images with less than 15% of morphing. Above 15%, the morphing detector can detect 95% of the potential impostors.

Therefore, we can conclude that a reasonable solution for preventing registration and login using fake face images can be built using face recognition and morphing detection state-of-the-art techniques. We have tested algorithms from different families of facial recognition techniques and found a clear difference between the one based on Deep Learning (FaceNet) and the rest. We will test newer and promising facial recognition algorithms that fall into this family of algorithms in our future work. Since the detection results are pretty robust against morphing processing, it would be interesting to challenge the solution proposed in this paper with better-designed algorithms for fooling its detection systems.

References

[1] S. F. Kak, F. M. Mustafa, and P. Valente, "A review of person recognition based on face model," *Eurasian Journal of Science & Engineeting,* vol. 4, issue 1, pp. 157–168, 2018. [Online]. Available doi: 10.23918/eajse.v4i1sip157.

[2] A. Shwetank, P. Neeraj, and B. Karamjit, "Future of face recognition: A review," *Second International Symposium on Computer Vision and the Internet,* vol. 58, pp. 578–585, 2015.

[3] M. Ferrara, A. Franco, and D. Maltoni "The magic passport," in *IEEE International Joint Conference on Biometrics,* Clearwater, FL, 2014, pp. 1–7. [Online]. Available doi: 10.1109/BTAS.2014.6996240.

[4] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2:342, 2020. [Online]. Available doi: 10.3390/s20020342.

[5] M. Wang and W. Deng, "Deep face recognition: A survey," 2018, *arXiv:1804.06655*. [Online]. Available: https://arxiv.org/abs/1804.06655.

[6] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of Computer Vision and Pattern Recognition IEEE Computer Society*, June 1991, pp. 586–591.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997. [Online]. Available doi: 10.1109/34.598228.

[8] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Proc. of the 5th Chinese conference on Advances in Biometric Person Authentication,* 2004, pp. 179–186. [Online]. Available doi: 10.1007/978-3-540-30548-4_21.

[9] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the Use of SIFT Features for Face Authentication," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, New York, NY, USA, 2006, pp. 35–35. [Online]. Available doi: 10.1109/CVPRW.2006.149.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015, pp. 815–823. [Online]. Available doi: 10.1109/CVPR.2015.7298682.

[11] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Technical Report 07-49, Oct. 2007.

[12] A. K. Agrawal and Y. N. Singh, "Evaluation of face recognition methods in unconstrained environments," *Procedia Computer Science,* vol. 48, pp. 644–751, 2015.

[13] X. Zhang and Y. Gao "Face recognition across pose: A review," *Pattern Recognition,* vol. 42, no. 11, pp. 2876–2896, 2009. [Online]. Available doi: 10.1016/j.patcog.2009.04.017.

[14] G. H Givens *et al.* "Introduction to face recognition and evaluation of algorithm performance," *Computational Statistics and Data Analysis,* vol. 67, pp. 236–247, 2013. [Online]. Available doi: 10.1016/j.csda.2013.05.025.

[15] G. Wolberg, *Digital Image Warping,* IEEE Computer Society Press, Los Alamitos, California, 1990.

[16] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face recognition systems under morphing attacks: A survey," *IEEE Access,* vol. 7, pp. 23012–23026, 2019. [Online]. Available doi: 10.1109/ACCESS.2019.2899367.

[17] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library.* Sebastopol, CA, USA: O'Reilly Media, 2008.

[18] S. Mallick, *Face Morph Using OpenCV - C++/Python,* Learn OpenCV, March 11, 2016. Retrieved: April, 2021. [Online]. Available: https://learnopencv.com/face-morph-using-opencv-cpp-python/.

[19] A. Hadid, N. Evans, S. Marcel, and J. Fierrez, "Biometrics systems under spoofing attack: An evaluation methodology and lessons learned," *IEEE Signal Processing Magazine,* vol. 32, no. 5, pp. 20–30, Sept. 2015. [Online]. Available doi: 10.1109/MSP.2015.2437652.

[20] A. Mohammadi, S. Bhattacharjee, and S. Marcel, "Deeply vulnerable: A study of the robustness of face recognition to presentation attacks," *Institution of Engineering and Technology Biometrics,* vol. 7, issue 1, pp. 15–26, 2018. [Online]. Available doi: 10.1049/iet-bmt.2017.0079.

[21] M. Ferrara, A. Franco, and D. Maltoni, "On the effects of image alterations on face recognition accuracy," in *Face Recognition Across the Image Spectrum.* Springer Nature, 2016, pp. 195–222.

[22] R. Tolosana, R. Vera-Rodríguez, J. Fiérrez, A. Morales, and J. Ortega-García, "DeepFakes and beyond: A survey of face manipulation and fake detection," *arXiv preprint arXiv:2001.00179*, 2020.

[23] D. J. Robertson, R. S. S. Kramer, and A. M. Burton, "Fraudulent ID using face morphs: Experiments on human and automatic recognition," *PLoS ONE,* vol. 12, no. 3:e0173319. [Online]. Available doi: doi:10.1371/journal.pone.0173319.

[24] L. Wandzik, R. V. García, G. Kaeding, and X. Chen, "CNNs under attack: On the vulnerability of deep neural networks based face recognitionto image morphing," in *Proc. 16th Int. Workshop on Digital Forensics and Watermarking,* 2017, pp. 121–135.

[25] U. Scherhag *et al.* "On the vulnerability of face recognition systems towards morphed face attacks," in *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, Coventry, 2017, pp. 1–6. [Online]. Available doi: 10.1109/IWBF.2017.7935088.

[26] M. Sharif, F. Naz, M. Yasmin, M. A. Shahid, and A. Rehman, "Face recognition: A survey," *Journal of Engineering Science and Technology Review,* vol. 10, no. 2, pp. 166–177, 2017.

[27] S. Patel, *Face Morphing*, Github, 2018. Retrieved: April, 2021. [Online]. Available: https://github.com/cirbuk/face-morphing.

[28] B. Delaunay "Sur la sphère vide. A la mémoire de Georges Voronoï [On the empty sphere. In memory of Georges Voronoï]," *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et naturelles* vol. 6, pp. 793–800, 1934.

[29] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research (JMLR),* vol. 10, pp. 1755–1758, 2009.

[30] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 2037–2041, Dec. 2006. Available doi: 10.1109/TPAMI.2006.244.

[31] R. Raja, *Face Recognition with OpenCV and Python*, Github, 2017. Retrieved: April, 2021. [Online]. Available: https://github.com/informramiz/opencv-face-recognition-python.

[32] D. Sandberg, *Face Recognition using Tensorflow*, Github, 2016. Retrieved: April, 2021. [Online]. Available: https://github.com/davidsandberg/facenet.

[33] J. Galbally, S. Marcel, and J. Fiérrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access,* vol. 2, pp. 1530–1552, 2014. [Online]. Available doi: 10.1109/ACCESS.2014.2381273.

[34] R. B. Hadiprakoso, H. Setiawan, and Girinoto, "Face Anti-Spoofing Using CNN Classifier & Face liveness Detection," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT),* Yogyakarta, Indonesia, 2020, pp. 143–147. [Online]. Available doi: 10.1109/ICOIACT50329.2020.9331977.

[35] B. Wu, M. Pan, and Y. Zhang, "A review of face anti-spoofing and its applications in China," in *International Conference on Harmony Search Algorithm,* Springer, 2019, pp. 35–43. [Online]. Available doi: 10.1007/978-3-030-31967-0_4.

[36] R. Raghavendra, K. B. Raja, and C. Busch. "Detecting Morphed Face Images," in *8th IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS),* pp. 1–8, 2016.

[37] L. Zhang, F. Peng, and M. Long, "Face morphing detection using Fourier spectrum of sensor pattern noise," in *2018 IEEE International Conference on Multimedia and Expo (ICME),* San Diego, CA, 2018, pp. 1–6. [Online]. Available doi: 10.1109/ICME.2018.8486607.

[38] R. Raghavendra, K. Raja, S. Venkatesh, and C. Busch, "Face morphing versus face averaging: Vulnerability and detection," in *2017 IEEE International Joint Conference on Biometrics (IJCB),* Denver, CO, 2017, pp. 555–563. [Online]. Available doi: 10.1109/BTAS.2017.8272742.

[39] L. Spreeuwers, M. Schils, and R. Veldhuis, "Towards robust evaluation of face morphing detection," in *2018 26th European Signal Processing Conference (EUSIPCO),* Rome, 2018, pp. 1027–1031. [Online]. Available doi: 10.23919/EUSIPCO.2018.8553018.

[40] M. Ferrara, A. Franco, and D. Maltoni, "Face demorphing," *IEEE Transactions on Information Forensics and Security,* vol. 13, no. 4, pp. 1008–1017, April 2018. [Online]. Available doi: 10.1109/TIFS.2017.2777340.

[41] K. Raja *et al.*, "Morphing attack detection - database, evaluation platform and benchmarking," *IEEE Transactions on Information Forensics and Security,* 2020. [Online] Available doi: 10.1109/TIFS.2020.3035252.

[42] W. Deng, J. Hu, N. Zhang, B. Chen, and J. Guo, "Fine-grained face verification: FGLFW database, baselines, and human-DCMN partnership," *Pattern Recognition,* vol. 66, pp.63–73, 2017.

[43] V. K. Vishnoi, *What Is The Configuration Of The System That Is Being Used In Matlab Online Web Based Version,* MATLAB Answers, January 24, 2020. Retrieved: April, 2021. [Online]. Available: https://www.mathworks.com/matlabcentral/answers/500853-what-is-the-configuration-of-the-system-that-is-being-used-in-matlab-online-web-based-version.

[44] E. Learned-Miller *et al.*, "Labeled Faces in the Wild: A Survey," in *Advances in Face Detection and Facial Image Analysis*, 2016, pp. 189–248. [Online]. Available doi: 10.1007/978-3-319-25958-1_8.

 ISBN: 978-1-61208-862-4

MAEVA: A Framework for Attack Incentive Analysis with Application to Game Theoretic Security Assessment

Louai Maghrabi
Department of Cybersecurity
School of Engineering, Computing & Informatics
Dar Al-Hekma University
Jeddah, Saudi Arabia
LMaghrabi@DAH.edu.sa

Eckhard Pfluegel
School of Computer Science & Mathematics
Faculty of Science, Engineering & Computing
Kingston University
London, United Kingdom
E.Pfluegel@Kingston.ac.uk

***Abstract*—This paper is concerned with the risk assessment of cyber security attacks on an organisation. We develop the novel attack incentive analysis framework Motive, Ability, Exploitability, Visibility and Attractiveness (MAEVA) based on taking into account a multiplicative function of the attacker's anticipated attack effort and expected reward. We argue that our approach can complement and enhance the standard approach based on estimating risk as a function of attack likelihood and impact on the organisation. We then present an application of our framework to game-theoretic risk assessment, illustrating how it can be used to inform the modelling of attacker-defender scenarios using complete information games. This helps to establish more realistic game-theoretical modelling of security assessment scenarios for practical use.**

***Index Terms*—Cybersecurity, risk assessment, game theory, security games, Nash equilibrium analysis**

I. Introduction

With the advancement and continuous growth of the digitally connected world through the Internet, cyber security has become a matter of global interest and importance to governments and private organisations to ensure achieving the major security requirements of Confidentiality, Integrity and Availability (CIA) of critical assets. To put this into some context, for example, a large organisation, such as SolarWinds recently had a data breach through hidden malicious code inserted into widely-used SolarWinds software, without being detected for several months. The attack gave adversaries access to systems of multiple U.S. government departments, including the Energy Departments nuclear arsenal. In another recent incident, Garmin, makers of Global Positioning System (GPS) devices, smart wearable devices and aviation technology, suffered a ransomware attack that brought down its own systems affecting the availability of data [1].

Security incidents of such high severity highlight the importance of security controls and mitigation techniques, and most governments and organisations nowadays have developed some form of strategies to categorise risks, apply vulnerability controls and mitigate threats in order to protect critical assets. National and international standards exist, to recommend formal frameworks and security management methodologies. *Security management* refers to a collection of activities that seek to, in the most general sense, the identification, assessment, analysis, establishment and evaluation of the security of a system or an organisation. This process can be carried out in different contexts such as information security, network security, system application or software security or nowadays generally in cyber security. Managing the security of an organisation can reduce the risk of running unexpected costs, help with standardising security practices, and show effective compliance with legislation and regulatory policies.

Risk management is the risk-based, top-down approach of security management. According to the National Institute of Standard and Technology (NIST), risk management is established as a risk context by producing a risk management strategy on how to identify, assess, respond, mitigate and monitor risks within an organisational context [2]. Generally, the following are typical risk management activities:

- Decide on how to implement a protection strategy and design risk mitigation plans by developing an action plan;
- Implement the detailed action plan;
- Monitor the action plans for schedule and effectiveness;
- Control variations in plan execution by taking appropriate corrective actions.

In this paper, we are studying the fundamental problem of how to compute the risk that an organisation faces from external attacks, and how to respond to it. According to [3], there are many approaches to assess risks. Risks can be assessed through qualitative or quantitative approaches, with underlying mathematical models of various degrees of complexity. Fundamentally, risk assessment attempts to measure the impact of an attack on an asset, mitigated by the probability (likelihood) that the attack will occur. In [4], the additional difficulty of a large (and ever-growing) attack surface of typical organisations and their assets, and the fact that risk can be seen as a map with different values at each point of the enterprise attack surface, is reported. Risk is seen as a

 ISBN: 978-1-61208-862-4

function of vulnerabilities in the system, their exposure to an attacker, the presence of active (relevant) threats, the existence of mitigating controls and the impact on the organisation. In this paper, we are interested in the risk assessment stage. We assume that prior to this step, critical assets and their security requirements were identified and that the above-stated relevant attack surface parameters are known.

This paper presents two contributions. The first contribution is a novel framework for risk assessment of cyber security attacks on an organisation. The framework is based on analysing the incentive an adversary may have to attack the organisation when weighing up the potential gain from the attack and the effort it takes to breach the system. We argue that this point of view, which is fundamentally different to that taken in traditional risk assessment, can complement and enhance the standard approach based on estimating risk as a function of attack likelihood and impact on the organisation. The second contribution is an application of this framework to game-theoretic risk assessment. We show that our framework is very convenient when wishing to inform the design of complete information games, modelling attacker-defender scenarios. It is hence a natural first step an organisation can take to prepare a game-theoretic risk assessment, and to reap the benefits from this approach which might have advantages compared to standard risk assessment.

This paper is organised as follows. Section II reviews modern risk assessment methodologies and formulates the main research question. In the subsequent section, the novel framework is introduced. Section IV proposes the application to game theory. The last section is the conclusion.

II. Security Assessment Based on Risk Analysis

The fundamental problem of how to compute the risk that an organisation faces from security attacks is the subject of numerous security risk assessment methodologies. In this section, we will recall the principles of risk computation and its challenges, review some popular mature security assessment frameworks, and discuss how they can help with the task of attack likelihood assessment and impact analysis.

A. The Challenges of Risk Computation

Using formal notation, the risk R can be expressed as an expected impact I, computed using the equation:

$$R = p \cdot I \tag{1}$$

where p is the probability of an attack occurring, often referred to as *attack likelihood*. From this equation, one can see that the problem now is to quantify and compute p and I and the difficulty is to perform a realistic estimate of these variables. Likelihood assessment is the process of establishing an estimate for p [5]. However, as pointed out in Tripwire [6], likelihood assessment appears, in general, to be a challenging and elusive task. Informally, the impact I is the overall damage that the targeted asset owner suffers from, this includes any indirect cost to the organisation such as a loss of reputation or business revenues. Impact is a central concept in the various security assessment frameworks, although it is defined slightly differently. This will be explored further in the following sections.

B. NIST

NIST Risk Management Framework (RMF) is a popular and detailed framework. Quoting [2], it states that *"the level of **impact** from a **threat** event is the magnitude of harm that can be expected to result from the consequences of unauthorised disclosure of information, unauthorised modification of information, unauthorised destruction of information, or loss of information or information system availability."* In other words, the impact from an attack on an asset is the degree of harm that affects the security requirements of confidentiality, integrity and availability for an asset. In this definition, the impact is created by a threat event, in line with the risk-based approach explained earlier. It is assumed that one is able to determine the attack likelihood. This leads to a table containing risk response actions, such as defending critical assets, recovering from an attack, planning for defense or choosing not to respond at all [2]. An appropriate response action is then determined by indexing the table rows with attack probabilities using qualitative metrics (low, medium or high) and its columns with a measure for the impact (minor, moderate or major) of the attack on the asset, or more generally, the organisation as a whole. This table, referred to as *Risk Response Matrix* (RRM) in this paper, is illustrated in Figure 1.

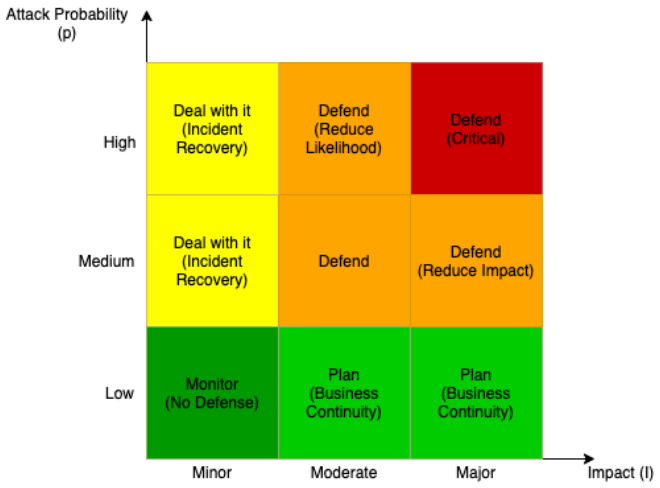


Fig. 1. Risk Response Matrix (RRM) [2]

C. OCTAVE

The Operationally Critical Threat and Vulnerability Evaluation (OCTAVE) framework [7] can be used to relate impact to both threats and vulnerabilities: *"All aspects of risk (assets, **threats**, **vulnerabilities**, and organisational **impact**) are factored into decision making, enabling an organisation to match a practice-based protection strategy to its security risks."* This

framework does not explicitly link the analysis of risk to the probability of an attack occurring. Instead, it informs the analysis based on threat profiling, enhanced by impact statements, leading to risk profiles. OCTAVE recommends at least looking at the following impact areas: safety, health, productivity, reputation, financial and fines. The analysis is done in a qualitative manner, but approximated scores could be derived from this.

D. CVSS

The Common Vulnerability Scoring System (CVSS) primarily focuses on software vulnerabilities, and the assessment of their severity. The idea is to provide a *base score* $\mu_B(v)$ for a CVE-indexed vulnerability v based on open criteria, and to make the score publicly available on the National Vulnerability Database (NVD) website [8]. This overall base score is further refined using an *impact score* $\mu_I(v)$ and an *exploitability score* $\mu_E(v)$. Quoting from [9], *"...the **impact** metrics reflect the direct consequence of a successful **exploit**, and represent the consequence to the thing that suffers the **impact**, which is referred to, formally, as the impacted component."* CVSS formulates the impact as the direct damage to an asset through an exploited vulnerability. In the context of risk assessment, we can hence use $\mu_I(v)$ for impact computation for the subset of suitable assets. It is less clear, how this could help with attack likelihood computation.

E. STRIDE and DREAD

STRIDE [10] was introduced in 1999 by Microsoft as a threat profiling scheme for categorizing potential threats according to their impact on common security requirements. The STRIDE acronym is formed from the first letter of each of the following categories, which cover a fairly complete range of threats when considering the original context of secure application development:

1) **S**poofing identity: illegally accessing and using another user's authentication credentials.
2) **T**ampering with data: malicious modification, fabrication or deletion of data.
3) **R**epudiation: the denial of having performed an action, in an environment lacking the capability to prove otherwise.
4) **I**nformation disclosure: exposure of information to individuals who are not authorised to have access to it.
5) **D**enial of service (DoS): an attack that interrupts the availability of a service to valid users.
6) **E**levation of privilege: an unauthorized or unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire asset or system.

Hence, risk assessment with STRIDE consists of eliciting threats using the approach above, followed by a rating system in order to rank threats by their criticality. This can be done using the less well-known DREAD [11] approach, based on the following key categories:

1) **D**amage potential: the degree of the potential damage a specific threat can inflict on an asset.
2) **R**eproducibility: this gives an understanding of the level of complexity of the threat, by assessing how easily it can be replicated by different adversaries.
3) **E**xploitability: this aims to quantify how easy is it for an attacker to succeed in exploiting the vulnerability targeted by the threat.
4) **A**ffected users: an estimate of the number of affected users in the aftermath of the attack.
5) **D**iscoverability: How difficult is it to discover vulnerabilities in the system, targeted by the threat.

By inspecting all of these DREAD categories and adding up individual scores, a risk rating is determined for each threat and the vulnerabilities affected by it.

III. A Framework for Attack Incentive Analysis

Assessing and responding to risk based on estimating attack likelihood and impact, and deciding on suitable response actions by forming and inspecting the corresponding risk response matrix seems natural and intuitive. While it is indeed a mainstream approach used in popular mature and standard security assessment frameworks as reviewed in the previous section, it is somewhat self-centric and might only lead to a limited view of the external threat and attack landscape. In particular, it fails to take into account the attacker's capabilities and perspective, in terms of his or her underlying motivation of the attack, knowledge of the target and its vulnerabilities, as well as the expected benefits gained. In this section, an alternative approach for informing risk assessment is devised, based on estimating the incentive to attack, that the adversary may have. While it seems very reasonable to assume that an informed attacker would wish to follow this framework, we will also suggest that the framework could be useful for the organisation that might be targeted by the attacker, as an alternative security assessment approach. This aspect will be further explored in the discussion part of this section.

A. Attack Incentive Matrix

An attacker is mainly motivated by the anticipated reward from the attack, which will be referred to as the *gain* in this paper, denoted by G. This gain however will be diminished by the *effort* he or she has to invest in order to implement the attack. This effort, denoted by e, is spent by exploiting (technical) vulnerabilities and breaching cyber security defences. Overall, the *attack incentive* A can be computed as

$$A = e^{-1} \cdot G. \tag{2}$$

Although this equation is simple and might not accurately reflect a potentially more complex inter-dependency of the involved parameters in real scenarios, we want to maintain a degree of simplicity which is comparable to that in the risk computation formula (1).

This idea leads to our proposed *Attack Incentive Matrix* (AIM) depicted in Figure 2, describing possible actions that the attacker might take, depending on the attacker's expected

 ISBN: 978-1-61208-862-4

gain (low, medium or high) and effort (minor, moderate or major) reference by the rows and columns of the matrix. We propose to specify the following actions: plan to attack (monitor target), information gathering (reconnaissance), and attack the target.

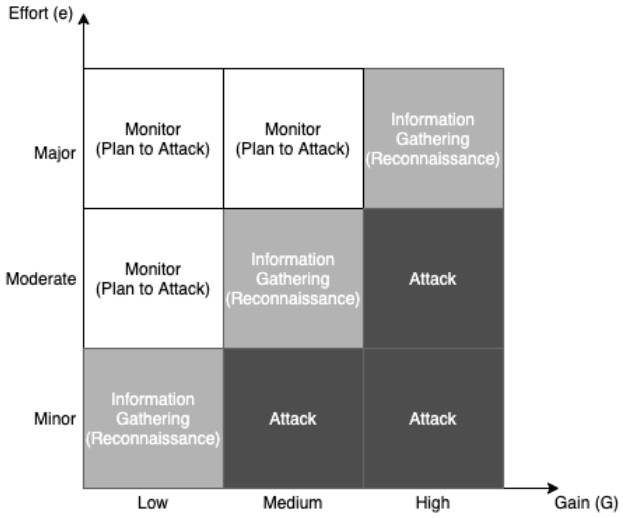


Fig. 2. Attack Incentive Matrix (AIM)

B. Proposed Framework: MAEVA

In this section, we present our MAEVA framework as guidance for computing the attack incentive A, based on estimating the effort e and the gain G. This proposed framework has several key characteristics partially inspired by considerations already used in the security assessment methodologies reviewed in the previous section, however in a different context. Our framework recommends considering the following categories when trying to estimate the required parameters, guided by following the MAEVA mnemonic:

1) **M**otive: the underlying reason for attacking the victim. This could be for the purposes of financial gains, revenge, personal satisfaction or thrill, or simply with the intention of creating damage. From a psychological point of view, the attacker's motive might affect the perceived gain, as well as the appreciation of the effort required.
2) **A**bility: the capability of the attacker to invest in resources for implementing the attack, as well as the technical knowledge available for breaching cyber security controls. A strong ability will make it easier to spend effort on the attack, and subjectively reducing the perceived value of e.
3) **E**xploitability: the ease by which the system can be penetrated, through exploiting a vulnerability. It would be reasonable to expect exploitability and effort to be inversely related in a proportional manner. This category could be explored similarly as in the CVSS exploitability score, taking into account possible attack vectors and attack complexities, as well as the required privileges and interaction with users, however, the discussion should not be restricted to software vulnerabilities alone.
4) **V**isibility of target: how prominent is the target, for example, does it have a popular website or brand name, does it have a large user base? Great visibility might promise a big gain, in the eyes of the attacker.
5) **A**ttractiveness of target: from the point of view of the attacker, how attractive is the target? This is linked to how much gain the attacker would estimate from achieving through the attack, and will strongly depend on the specific motive, as discussed in the first category.

While the MAEVA framework is intended to be used by the risk assessing organisation, it is an attacker-aware framework and the main assumption of its use is that an attacker would find it very natural to follow the same methodology in order to have a more systematic way to locate specific points within the AIM matrix in a given scenario, and use this as a guide for the decision to attack or not.

C. Discussion

We have reviewed the security assessment approach based on computing risk and introduced an alternative framework for modelling the attack incentive. Both approaches bring challenges in terms of achieving precise estimates for realistic results in practical scenarios. This will be briefly discussed and the advantages of combining both approaches outlined in this section.

In the risk matrix approach, the parameter that is more challenging to estimate is the attack likelihood p, as it depends a lot on external factors outside of our control. When trying to model attack incentive matrices, the difficult parameter is the effort e, since this has to be viewed as a relative quantity, depending on the capabilities of the attacker. Both approaches are complementary and if we use both, we can develop a better understanding of the risk that the organisation faces from an impeding cyber security attack.

By taking into account both perspectives (attacker, defender), a good understanding of the impact I can be developed by comparing it with the gain G. A discrepancy might reveal the need for correcting any of those two parameters. Furthermore, the attack likelihood p would be closely related to the attack incentive A, and this can help with computing R. As the effort e will depend, amongst other things, on the organisation's willingness to apply a security control, in other words, the perceived risk R, it might be necessary to adapt the estimate for A. After several iterations of estimations and adaptions, a final model should be obtained. We argue that the resulting figures are much more reliable and realistic, than those obtained without using MAEVA.

IV. Application to Game Theory

In this section, we will show that the RRM together with the AIM approach can be used naturally when modelling a non-cooperative two-player non-zero-sum complete information game, which is a specific type of security game useful for game-theoretic risk assessment. A complete information game

 ISBN: 978-1-61208-862-4

means that each player knows the strategies and payoffs of the other player in the game, but not necessarily the actions. For more background information on security games, we refer to [12].

A. Game Description

We are concerned with a single-target security game $G(\mathcal{D}, \mathcal{A})$ where the main focus is on a single asset that has an exploitable vulnerability. Our simple game comprises of two players: an attacker $\mathcal{A}$ and a defender $\mathcal{D}$ where each player has their own strategies as illustrated in Table I. The rows corresponds to the strategies available to the defender: $S_{\mathcal{D}} = \{\text{defend, not defend}\} = \{s_d, s_{-d}\}$, and the columns indicates the attacker's strategies: $S_{\mathcal{A}} = \{\text{attack, not attack}\} = \{s_a, s_{-a}\}$. Moreover, there is a payoff function (e.g., cost and benefit) that each player will incur depending on their chosen strategy: $c_{\mathcal{D}}$ is the defence cost, I is the defender's loss (impact) from an attack. By $c_{\mathcal{A}}$ we denote the attacker's cost, and G is the gain (benefit) of the attacker from an attack. Note that we have used notations that are compatible with the previous sections. The following natural assumptions [13] are usually made for this type of security game: the *Principle of Adequate Protection* prescribes that defence costs must not exceed potential losses: $c_{\mathcal{D}} < I$, and the *Principle of Easiest Attack* states that the attacker prefers to keep his or her cost for attacking bounded by the expected gain: $c_{\mathcal{A}} < G$. The game is described using its payoff matrix, which specifies its strategic normal form:

TABLE I
PAYOFF MATRIX FOR $G(\mathcal{D}, \mathcal{A})$

$\mathcal{D} \downarrow \mathcal{A} \rightarrow$	s_a	s_{-a}
s_d	$-c_{\mathcal{D}}$, $-c_{\mathcal{A}}$	$-c_{\mathcal{D}}$, 0
s_{-d}	$-I$, $G - c_{\mathcal{A}}$	0, 0

B. Game Analysis

When using a so-called *Nash Equilibrium strategy*, none of the players will have the incentive in deviating unilaterally from this strategy as this will reduce his or her expected utility. The following results are well-known properties of security games such as the game G, c.f. [12].

Theorem 1. *The security game $G(\mathcal{D}, \mathcal{A})$ has no pure Nash Equilibrium.*

Proof. By inspecting the payoff matrix of the game. □

Theorem 2. *A mixed Nash Equilibrium strategy pair $(x^*_{\mathcal{D}}, x^*_{\mathcal{A}})$ is obtained, where $q^* = 1 - c_{\mathcal{A}}/G$ and $p^* = c_{\mathcal{D}}/I$ are the probability of defense and attack respectively.*

Proof. Following Nash, as further detailed in [12]. □

In the context of security assessment, the outcomes of the game analysis have the following implications:

- Due to the lack of a pure equilibrium solution, there is no clear-cut decision whether to defend or not, as there is a dilemma between the conflicting non-cooperating players of the game.
- The mixed equilibrium solution can be interpreted as a means to compute risk, by interpreting the mixed strategy of the attacker as a probability value: $R = p^* \cdot I$.

Hence, a more systematic and theoretically justified way to compute risk can be achieved, based on game theory.

C. MAEVA Application

As we have seen, under the assumption of complete information about the strategies available to both players, the use of game theory improves the traditional risk assessment approaches as it combines both the non-cooperative nature of the defender and the attacker. Before the game can be solved, it needs to be specified in terms of the precise values for the payoff functions, and Table I reveals that the MAEVA framework can be used to determine (an estimate for) G. The parameter $c_{\mathcal{D}}$ is effectively the *defense budget* of the organisation and $c_{\mathcal{A}}$ can be related to the attacker's effort e. Hence, in a natural way, both the RRM and AIM methodologies provide the input parameters for the game. The analysis of the game based on computing the Nash equilibrium will then result in the desired risk value, following the computation as presented in the previous section.

V. CONCLUSION

In this paper, we have proposed a new framework entitled MAEVA, for analysing the attack incentive of a cyber security adversary of an organisation. Furthermore, we have shown how to use this framework in combination with traditional risk analysis, in order to achieve a more refined strategy to assess typical risk-related parameters such as attack likelihood and impact. We have also demonstrated that the framework is useful as preparation of game-theoretic modelling of risk assessment. To our knowledge, our framework constitutes a novel approach and we recommend using it as a practical methodology for any organisation wishing to assess risk, perhaps in combination with other mainstream methods.

The next step for this research would be an implementation of a real scenario, and a detailed evaluative comparison with existing approaches. For example, an organisation could review their information assets, apply both the RRM and AIM, and compare the resulting parameters. It would be interesting to relate this to historical information about cyber security incidents that happened in the past at this organisation, or in its sector. Ideally, we would expect an advantage resulting from the dual use of these frameworks, in terms of obtaining more realistic risk estimates. While not being the main focus of this paper, another interesting aspect that deserves further attention is to more deeply explore the link between traditional and game-theoretical security assessment. The authors believe that risk assessment modelling using game theory would have numerous advantages and that it should be considered for use in future versions of mainstream security assessment methodologies.

 ISBN: 978-1-61208-862-4

REFERENCES

[1] M. Gostovnikas, "The 9 worst recent data breaches of 2020," https://auth0.com/blog/the-nine-worst-recent-data-breaches-of-2020, January 2021. [Online].

[2] NIST, "NIST SP 800-30: Guide for conducting risk assessments," NIST, Tech. Rep., September 2012. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf{\%}5Cnhttp://csrc.nist.gov/publications/PubsSPs.html{\%}5Cnhttp://dx.doi.org/10.6028/NIST.SP.800-30r1

[3] S. Derakhshandeh and N. Mikaeilvand, ""New framework for comparing information security risk assessment methodologies"," *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 9, pp. pp. 160–166, 2011.

[4] G. Banga, ""Why is cybersecurity not a human-scale problem anymore?"," *Commun. ACM*, vol. 63, no. 4, p. pp. 3034, March 2020. [Online]. Available: https://doi.org/10.1145/3347144

[5] W. Kanoun, F. Cuppens-Boulahia, N. Cuppens, S. Dubus, and A. Martin, ""Success likelihood of ongoing attacks for intrusion detection and response systems"," in *2009 International Conference on Computational Science and Engineering*, vol. 3, 2009, pp. pp. 83–91.

[6] Tripwire, "Tripwire Vulnerability Scoring System," https://dsimg.ubm-us.net/envelope/160343/293772/1396040281_Tripwire_Vulnerability_Scoring_System_white_paper.pdf, p. 8, 2016. [Online].

[7] C. Alberts, A. Dorofee, J. Stevens, and C. Woody, "Introduction to the OCTAVE approach," Carnegie Mellon University, Tech. Rep., 2003.

[8] N. I. of Standards and Technology, "National Vulnerability Database," https://nvd.nist.gov/, March 2021.

[9] FIRST, "Common Vulnerability Scoring System Version 3.1," FIRST (Forum of Incident Response and Security Teams), Tech. Rep., June 2019. [Online]. Available: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

[10] Microsoft, "The STRIDE Threat Model," https://msdn.microsoft.com/en-us/library/ee823878(vcs.20).aspx, pp. 1–3, 2002.

[11] EC-Council, "DREAD threat modeling: An introduction to qualitative and quantitative risk analysis," https://blog.eccouncil.org/dread-threat-modeling-an-introduction-to-qualitative-and-quantitative-risk-analysis/, Dec 2020.

[12] T. Alpcan and T. Basar, *Network security: A decision and game-theoretic approach*. Cambridge University Press, 2010.

[13] C. Pfleeger, *Security in computing*. Prentice Hall, 2015.

 ISBN: 978-1-61208-862-4