Bacterial Carrier for DNA encoded data and Detection Approaches for Bio-cyber attack

Bacterial Nano Communication and Security in Sequencing

Mohd Siblee Islam

Phd Thesis

Supervised by:

Dr. Sasitharan Balasubramaniam & Dr. Stepan Ivanov

Department of Computing

South East Technological University

Ireland

I would like to dedicate this thesis to my beloved wife Anjuman Ara Kali, my mom Nasira Islam, my father Mohd Rafiqul Islam and my son Ahnaf Adib.

Declaration

I hereby declare that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Mohd Siblee Islam
Submitted to South East Technological University, Ireland, July 2023

Acknowledgements

First, I would like to thank Almighty Allah for the blessing He bestowed upon me that enabled the completion of my thesis. I am immensely thankful to my family members, grand parents, other extended family members, and friends who always encouraged me throughout the years to achieve a doctorate degree.

Few names must be mentioned, especially those without whom I could not get this far. First, I would like to show my gratitude to my two excellent supervisors, Sasi Balasubramanium and Stepan Ivanov. I am grateful to them as without their constant guidance, encouragement and support, it would be quite impossible to produce quality research and publications. These will always remain as a great asset to me and help me in my future career.

I would like to remember my late father Mohd Rafiqul Islam, who always inspired and encouraged me to do the best in life and taught me how to be humble, honest and helpful. I am forever grateful to my mother for her prayers and support. Although she always dreamed of me accomplishing a PhD degree, at the same time she helped me to remain calm and patient throughout this long journey. Her love and advice always kept me focused and helped me enjoy my research even in the most difficult times. I am more than thankful to my lovely wife Anjuman Ara Kali and two adorable sons Ahnaf and Ayaan for sacrificing countless weekends and holiday plans to let me work on my PhD obligations.

Finally, I believe that my preparation for the PhD began even before the start of the degree. Competence and confidence required to pursue a PhD can only be achieved over time through learning many hard and soft skills. Small suggestions, advice, sharing of experience, small favour, etc. can make a significant difference. Therefore, I would like to mention few names here who helped me in such ways. I really appreciate my cousin AFM Zakaria and Sirajum M. Fahim, my school teacher Tapan K. Mallik, my masters thesis supervisor Mobyen U. Ahmed, my brother in law Dr. Shah Alam, my father in law Ali Chowdhury, my sister Rukaiya Rafique, my friend Ranadip Barua, Dr Aminur Rahman and Dr. Jai Mehta for their supports.

Abstract

Internet of Bio-nano Things is the idea of using various bio-compatible nano and micro scale devices in the body that create networks and can connect to the existing cyber world. In recent research, bacteria are proposed as nano scale devices for such communication utilizing various existing characteristics of them or by introducing new properties with the help of genetics engineering. Therefore, in the future, bacteria can be used as information carriers, transmitters, receivers, nano devices, sensors, etc. The major advantage of using such devices is that the devices will be bio-compatible and no external conventional energy sources will be required to operate them.

Bacterial traits such as mobility and conjugations have been proposed for data transmission in the recent past. But most of the techniques involve sending one bit at a time using diffusion of bacteria. The first contribution of this PhD research is to propose a novel data transmission technique using bacterial mobility and bioluminescent properties, where we can send two bits at a time.

A common technique for bacterial data transmission is encoding the message in bacterial DNA, especially plasmid DNA, so that the bacteria will reach the receiver and offload the information into another bacteria by conjugation. We can assume that to read this information, a DNA sequence will be required. Moreover, many research studies have been performed on storing data in DNA as it shows immense promise of data storage without requiring any external energy. Sequencing pipelines are used in the decoding process of such stored data. In recent years, due to various needs (e.g., COVID-19), DNA sequencing has become quite common, and the number of applications that require DNA sequence is also growing day by day. Unfortunately, very little attention has been given to the possibility of vulnerabilities and the exploitation in the DNA sequencing pipelines. This doctoral research also contributes towards securing the DNA sequencing pipeline so that we can ensure secure data transmission in bio-nano communication.

In a recent research, the buffer overflow vulnerability in a tool in a DNA sequencing pipeline can be exploited using specially designed DNA. An attacker can attempt to insert malicious payload inside the DNA sequence in order to compromise the DNA sequencing pipeline. Further investigation is necessary to validate whether in a real world scenario, the malicious payload encoded into DNA can reach a sequencer after placing them into live bacterial plasmids. It is also very important to create countermeasures to detect such a sequence and use that detection mechanism as a safeguard for the DNA sequencing pipeline. So, in our research, we

have conducted an end to end evaluation of detecting malicious input for the buffer overflow exploit in the DNA sequencing pipeline. A machine learning based input control is proposed to classify every read of the sequencer machine to check if it contains any part of the encoded malicious payload. If detected, further processing can be terminated to protect the pipeline downstream from being hacked. For the machine learning solution, a *Case Based Reasoning (CBR)* approach is proposed. We achieved promising results where the performance improved with the increase in the number of cases in the case library. Furthermore, wet lab experiments were conducted to verify whether the encoded malicious payload can be sustained after sequencing if they are inserted into living bacteria. The experiment involved bacteria with malicious payload inserted in plasmid DNA to be sprayed over different materials, which were then collected for sequencing. These experimental results demonstrated that such malicious payload can successfully reach the sequencing pipeline.

For the buffer overflow exploit scenario, simple detection techniques, such as *CBR*, can be sufficient where natural DNA sequences are expected, as the insertion of malicious input can make the DNA sequence quite unnatural. However, to make the detection harder, we came up with a novel scenario of Trojan based attack in the DNA sequencing pipeline where the DNA sequence with malicious data will remain very natural. The assumption is that, the DNA sequence pipeline tool will already be affected by a Trojan and remain dormant. The Trojan will only be triggered with a specific input signal and the same signal is then used to compromise the target. The benefit of this scenario is that fragmentation, encryption and steganography can be applied to the malicious input signal and inserted into a natural DNA. A state-of-the-art bio-informatics algorithm was used to estimate the difference between sequence with malicious input and the original DNA sequence for various size of fragmentation, retention positions for steganography and various encryption keys. In order to keep the DNA close to original, The best possible locations for fragment insertions is chosen to control mutations. An end-to-end evaluation is also performed for Trojan attack scenario, where deep learning based technique is proposed as a detection method for input control mechanism. We achieved up to 100 percent accuracy in detection using the proposed technique. Even after applying smaller fragment size, encryption, and higher retention to make detection much harder, the accuracy remained very high. For scenarios with encrypted malicious input, the accuracy was higher with the knowledge of the encryption key the accuracy compared to having no prior knowledge about the key.

Contents

List of Figures

List of Tables

Special Terms

bioluminescence The behaviour of producing and emitting light of a living organism. 3, 5, 6, 46

buffer overflow When a computer program writes beyond the designated buffer memory. For example, in programming language like *C*, if we pass a very large string as a parameter of the *'strcpy'* function then it may cause buffer overflow. 7, 12, 13, 27, 32, 34, 37, 40, 41, 44, 45

DNA sequencing pipeline The steps and process of retrieving the sequence of Nucleotide bases in DNA molecules. 2–4, 7–9, 23, 24, 27, 29, 32–34, 40, 41, 44, 46, 47

DNA Synthesis The process of constructing DNA molecules. In a lab, a desired DNA sequence can be created by DNA synthesis. 2, 4, 8, 18, 20, 43, 47

GFP Green Fluorescent Protein (GFP) is a protein required for bioluminescence behaviour of bacteria. 25, 26, 46

IoBNT The concept of connecting biocompatible nano devices to the internet to collect data from the granular level and also perform actions in that level. 1–7, 10, 23, 37, 40, 43, 44, 46

plasmid DNA The circular and extrachromosomal DNA molecules. It is available in bacteria. 2–4, 6, 7, 11, 12, 24, 30, 31, 36, 43

Trojan A software or a hardware is claiming to do a legitimate work and also doing that work but doing doing harmful and malicious activities hiddenly. 7, 8, 10, 13, 27, 29, 32, 33, 36, 37, 40, 41, 44–46

Chapter 1

Introduction

Internet of Things (IoT) connects various devices used in our daily lives to the cyber-world with the help of advancements in *Wireless Sensor Networks (WSN)*, *Near-Field Communication (NFC)*, and *Radio Frequency Identification (RFID)* [52], etc. As far as processing is concerned, Cloud and Fog computing incorporates intelligence into these devices in order to analyse data locally. *IoT* will create opportunities like building smart cities, environment, health monitoring systems to name a few [52, 4]. As far as devices that connect *IoT* are concerned, we are now witnessing a shift towards miniature devices and incorporating different technologies. For example, *nanotechnology* facilitates the development of devices at nano and micro scale. Nano devices are of such a scale that can be used sense molecules at fine granular level. These devices will help people by collecting precise and detailed information. A new paradigm of the *Internet of Nano Things(IoNT)* is introduced in [5], and incorporates nano routers and nano devices performing the tasks of basic and simple sensors and actuators. The *IoNT* research area focusses on various concepts required to connect these nano devices to the internet as well, e.g., proposing system architecture, developing new algorithms and novel communication protocol to suit the properties of these devices. In *IoNT*, carbon based nano electronics devices are considered to be communicating using Electro-Magnetic (EM) signals. Alternatively, biologically engineered cells such as bacteria can be used to produce bio-nano devices where these devices communicate using *molecular Communication (MC)*[4]. This form of device takes the concept of "things" to a new level, where its vision is to engineer and construct devices that use biological components. These biocompatible devices that connect to the cyber world have led to a new paradigm known as the *Internet of Bio-Nano-Things(IoBNT)*[4]. As part of *IoBNT*, there is medium range communication, where molecules are produced by bacteria. As a comparison to a typical device that connects to the internet, bacteria are considered as data packets, where the instructions and data are encoded inside the *DNA*

plasmids. They are able to transfer this concent to another bacteria by a process called conjugation. If we are diving deeper into the information that is encoded into the plasmid DNA, recent research has investigated digital information storage using these molecules. Research in DNA based storage has drawn a lot of interest as it has shown good promises in terms of storage capacity, sustainability, as well as very low maintenance and cost [18].

Writing information into DNA has to go through a number of specialized processes and equipment. For example, reading will be performed through a sequencer (e.g., using Sanger sequencing and Next Generation Sequencing) and writing is based on DNA Synthesis techniques that combine individual nucleotides into a string. Using these kinds of methods, bit-by-bit data can be transferred from a source to a destination. This create opportunities for integrating the concept of *bacterial nanonetworks*, where bacteria communication systems are artificially created that has the ability to store synthetic plasmids with encoded information. This provides a number of advantages, such as novel "all biological" DNA storage infrastructures, where bacteria are used to transfer encoded information between cells for storage, as well as the ability to replicate and create redundancies in the stored data. The only challenge, which has not been addressed by the research community, is the mechanism of automating the entire process from encoding, synthesizing the DNA molecules, inducing uptake by the cells, and later retrieving the DNA molecules to sequence it. However, this thesis is not focused on this challenge, but rather on another key challenge that is not addressed by the research community. This challenge is in terms of cyberbiosecurity. While there are many definitions of cyberbiosecurity, we can define this as a discipline that deals with attacks that come from both biological as well as the cyber medium [59, 63].

We have seen in the recent past that a perpetrator can synthesize a specially design DNA for sequencing to exploit a specific type of vulnerabilities to compromise machines involved in the DNA sequencing pipeline [64]. Therefore, the DNA sequencing, which is part of the whole *IoBNT* system can be under the security threats. To investigate this type of attack further and other possible attacks is paramount important for the future success of the *IoBNT*. However, this goes beyond just *IoBNT*, where we also have to consider infrastructure for collecting, processing, and storing genetic data, which today is now very wide-spread due to a number of reasons. A very good example is the recent COVID-19 pandemic, where PCR testings are conducted to determine if people are infected. This requires removing the RNA from the virus, and perform reverse transcriptase to form the DNA molecules that will be passed through the sequencer in order to analyze if its the viral DNA. At the same time, we are also witnessing other verticals, such as smart agriculture, that collect a large amount of genetic data to improve efficiency. This

includes collecting genetic data of soil microbes, animal microbiomes, as well as plant microbes.

Therefore, improving the throughput of non-DNA encoding based data transmission, investigating possible security attacks in DNA sequencing pipeline and possible countermeasure to mitigate security issues are three broad areas investigated in this PhD thesis as urgent requirements to address for the advancements towards the future bacterial *IoBNT*. Our thesis is organized as follows. The motivation and background of the PhD work will be elaborated in the next section of this Chapter. The last two sections of this chapter discuss the challenges and limitations for the PhD thesis. Relevant work towards Internet of Bio-nano things using bacteria and related technologies considered for our work are discussed in Chapter 2. In Chapter 3, we formulate the research objectives in terms of research questions, describe the approaches to address these questions, and describe the contributions of the thesis. Finally, the thesis is concluded with a summary of research contributions and touches on the future work sections in Chapter 4.

1.1 Background and Motivation

In this section, the reasons, thoughts and main focus of conducting the PhD research are described in further detail. This section will help us understanding the scope of our research and elaborate on the ideas and its connection to security related to *IoBNT*.

To bring revolutionary changes in the applications in health monitoring systems, smart agriculture systems, environment monitoring, etc., where the objective is to improve and build a better life for the global society, will require collection of data from fine granular levels and develop actuations at that level. Collecting data and performing necessary actions in real time and seamlessly in the micro and nano levels will help us perform accurate, appropriate and faster decisions and take necessary actions precisely and also in time. *IoBNT* is the new paradigm and discipline that will guide us towards achieving the goal of such revolutionary changes and bring extraordinary improvements to our lives. Indeed the miniature organisms of bacteria is going to play an important role in *IoBNT* because of its prospects of being engineered into a bio-compatible nano device utilizing various traits. This includes chemotaxis motility, forming new structures such as biofilm and last but not least, having the ability of taking in synthetic plasmid DNA. Original works in the area of bacterial nanonetworks have investigated how single bit data transmission can be achieved using bacterial motility and bioluminescence behaviour [47]. Unfortunately, this kind of communications is super slow in nature because of the super slow

stochastic bacterial mobility that depends on numerous environmental conditions. Therefore, it is very important to discover ways to improve the transmission rate by finding ways of sending two or more bit of data in each time-slot to improve the overall throughout.

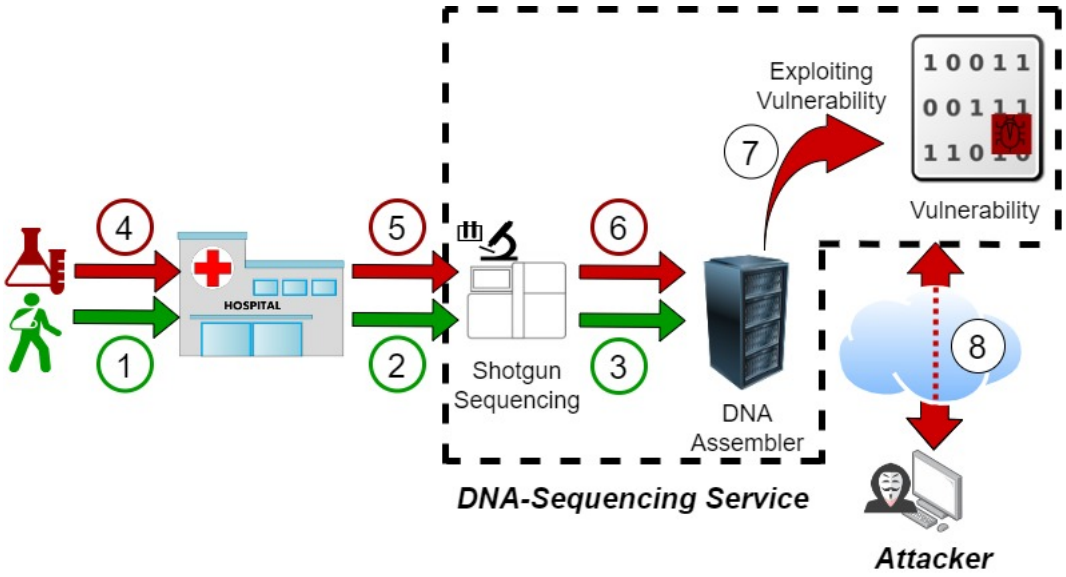


Figure 1.1: A targeted security attack in a DNA sequencing pipeline (through the path 4-5-6-7-8) [36].

On the contrary, bacteria can be used as a storage device in *IoBNT* due to the suitability of storing data into the DNA [18]. Moreover, data transmission by encoding the data in plasmid DNA of the bacteria is a complex process, but cannot be ignored as it will provide immense opportunity of storing large dataset inside bacteria to transfer. Although complex processes like DNA Synthesis and DNA sequencing will be involved in writing in and reading from the DNA, over a period of time, the technology of DNA Synthesis and DNA sequencing will be improved. Nanopore [39, 49] is an example of such improvement that make DNA sequencing process affordable and easy using a small device that can be connected to a laptop using the USB port similar to connecting an external hard disk. In addition, with the future improvement in DNA Synthesis in mind, researchers have already proposed using bacteria as programmable devices [45, 74, 11]. Therefore, the DNA Synthesis and sequencing will play a vital role in *IoBNT* indeed. This is where a new challenge came to our attention, and that is the possibility of compromising the DNA sequencing pipeline with the help of specially designed and synthesised DNA [64]. This indicates the possibility of taking control of a *IoBNT* system, incapacitate it, or cause harm by performing malicious activities by a perpetrator in the future, where they can implant bacteria containing specially designed DNA to hack

the system. Figure 1.1 is depicting an example of a targeted security attack using the vulnerabilities of a DNA sequencing pipeline by a specially designed DNA sample. If the DNA sample is not collected directly by the hospital (path 4-5-6 shown in Figure 1.1) then there is always a high chance of receiving a specially designed artificially created or tempered DNA sequence. The perpetrator can take advantage of this security risk to exploit vulnerabilities in the tools of the sequencing pipeline. Unfortunately, according to the best of our knowledge, no end-to-end evaluations have been performed to validate and confirm such possibilities. Moreover, it is also very important to investigate what are the best countermeasures to remedy such forms of attacks. In the past, we have experienced that it is quite common to keep non-functional system requirements, especially security, to the very end to address and this is also a very common phenomenon in many information technology driven systems [41]. Many a times, it is a reactive approach rather than a proactive mechanism. This will result in a lot of re-design, re-architecture and re-implementation of systems, tools and components and will also harm the reputation and trust of many systems. This is further complicated when we consider bio-cyber systems in the future, where the biological world is brought together with the cyber infrastructures. COVID-19 pandemic has shown us how there can be sudden pressure on a particular technology and system to diagnose and study the virus [67]. Compared to proactive approaches, reactive approaches can cost many lives and put society in danger. Therefore, to avoid such situations in the future, it is very important to consider the vulnerability in sequencing very seriously. This will have an impact not only in the future *IoBNT* but also in many other emerging applications in the near future.

This thesis contributes toward the future of IoBNT. As a contribution, we introduce a novel trigger based Trojan attack scenario in the DNA sequencing pipeline using a specially designed DNA. We also perform end-to-end evaluations of the state-of-the-art buffer overflow vulnerability exploit and our novel trigger based Trojan attack scenarios. In addition, we propose how we can identify DNA data that has been crafted to cause a buffer overflow in the control software of a DNA processing pipeline, or to trigger Trojan malware in a DNA processing pipeline. Another contribution of the thesis is to introduce a novel data transmission technique to increase the data transfer rate (bits/seconds) and to reduce the bit error probability rate; especially when for the data transmission bacterial traits like motility and bioluminescence are used, and complex data encoding into DNA is avoided.

1.2 Research Scope of the Thesis

In this section, we will discuss the challenges and limitations considered in this PhD thesis by discussing the scope.

1.2.1 Challenges

We discussed the opportunity created by using bacteria for *IoBNT* in the introduction section. The challenges towards achieving those opportunities considered for the PhD thesis are given below:

- **C1-Slow Nature of Communication:** A significant amount of research have been investigating the motility, chemotaxis and bioluminescence natures of bacteria, genetic engineering on them to modify existing characteristics or to introduce new characteristics such as bacterial conjugations and their communal behaviours. Several research works have been performed on bacterial motility and use them to transmit data. The bacterial movement is mainly a stochastic process, and it can be described by Brownian motion. Therefore, it is obvious that the communication process using bacterial motility will be very slow in nature as they will take long latencies to swim from a receiver to transmitter, which are placed a few millimetres apart. However, this form of communication is still very important as it provides us the opportunity of connecting Body Area NanoNetworks [8] to the Internet in future, while avoiding complex process of data encoding into DNA. This data transmission technique relies on the bacterial properties that include bioluminescence, mobility, where transmitting data uses diffusion-based techniques, which can send a single bit similar to On-Off keying. Based on this, it is really a challenge to send more than one bit of data at a time using bacteria while avoiding complex message encoding in plasmid DNA.
- **C2-Security Challenges of Data Encoding in DNA:** Data transmission using message encoding in plasmid DNA will require complex processes for data encoding and decoding. Despite this, it will remain as an important method in communication using the cells, as we can encode and send a large amount of data in its plasmids. Moreover, the conjugation process will assist in creating many replicas of the same message and make it more fault-tolerant. Researchers are also working to use DNA as a storage device and have already made significant progress toward that direction, e.g., a recent research showed how an image or even a movie can be stored inside the DNA of a bacterial population [85]. At the same time, DNA sequencing have become more and more common and improved with time. As mentioned earlier, a recent work

[64] has shown how we can design and build a DNA that can exploit the buffer overflow vulnerability, a well know vulnerability of computer hardware, to hijack the control of a machine in a DNA sequencing pipeline. This work reveals that *IoBNT* and DNA based data storage will be at risk as DNA sequencing pipeline can face such security threats. Therefore, it is very important to address this issue to further advance the paradigm of *IoBNT*.

- **C3-New Emerging Cyberbiosecurity Attacks:** Similar to the above mentioned buffer overflow vulnerability exploit threat, DNA sequencing pipeline might also be in the risks of many other kinds of security attack, which are already available in today's computers and networks. Yet this is still unknown to us. Moreover, if buffer overflow attacks are considered as more like a hardware or operating system platform level of attacks, since it utilises the limitations of hardware and the memory management of underlying system, then in contrast, the attacks such as using Trojan Horse, SQL and script injections can be considered as more like a software level attacks for cyberbiosecurity. This is because a malicious piece of code or script will be executed here, which does not depend on the limitation of the memory management or underlying hardware similar to buffer overflow exploit. It is important to know the possibilities of these kinds of software level attacks in the DNA sequencing pipelines as well. Furthermore, if such an attack is possible, then the next important question is to know the severity of such attacks. In addition, it is also important to know how sophisticated ways the payload and the attack can be designed, as the perpetrator will have more liberty compared to the hardware level attacks, so that the attack becomes deadly and very difficult to detect. If the scope of designing a DNA containing malicious payload is limited, then the solution might be very trivial, since the designed DNA is quite unnatural (the sequence is very different from the sequence of natural plasmid DNA). Thus, the presence of such malicious payload as DNA and occurrence of such attack might be detected in a short period of time. On the other hand, the solution will not be trivial in countering a sophisticated designed attack. It is also unknown whether building DNA for such a sophisticated attack is realistic or not. Without knowing the possibility of such an attack, the nature of such an attack, and how the attack will work, it will be hard to prepare the remedy for it in advance to comply with the advice "prevention is better than cure".
- **C4-Solutions to counter cyberbiosecurity attacks:** A possible attack (buffer overflow vulnerability exploit) is already known to us [64] but we do not know how to counter such attack as that recent research describing the attack simply suggest to follow standard security practices rather than proposing a solution to counter such attack. It is also unknown whether the solution can

be part of the DNA sequencing pipeline. Moreover, the solutions should also determine the software level attacks, for example Trojan attack, SQL injection attacks, or script injection attacks. Again, by using the term "software-level attacks", we mean that a piece of malicious code or script will be executed which will not rely on the underlying system limitations.

1.2.2 Limitations

In our PhD thesis we consider few specific limitations, in particular, to decimate. These limitations are related to the challenges mentioned in the last section. The limitations are as follows:

- **L1-Super slow bacterial stochastic motility:** The approach of sending data using bacterial traits such as motility is preferred over the technique of encoding data into DNA to avoid the complexity of DNA Synthesis and DNA sequencing for the required read and write operations. However, bacterial motility is a stochastic process that depends on numerous parameters. As a result, sending data bit by bit depending on the bacterial motion to a particular destination in sufficient numbers is a very slow process. This PhD work improves this limitation of sending of just one bit at a time based on the super slow stochastic bacterial motion, but relies on the structural formation of the topology of the bacterial nanonetwork related to **Challenge C1**.
- **L2-Lack of end-to-end evaluation:** In the previous work [64], only a particular vulnerability is considered to demonstrate the possibility of attacks in DNA sequence pipelines. However, there are no end-to-end evaluations to understand the complete process and validate such attacks in a real world scenario. In this PhD research, we consider this limitation and perform end-to-end evaluations for two types of attack scenario. This limitation is related to **Challenge C2** and **C3**.
- **L3-Lack of feasibility studies:** It is important to understand whether such an attack is feasible or realistic in terms of the sustainability of bacteria containing specially designed DNA. To validate this, we have developed collaborations with wet lab experimentalists to prove our end-to-end evaluations. This limitation is related to **Challenge C2**.
- **L4-Difficulties of explaining ML models based solutions:** The idea of exploiting different vulnerabilities in DNA sequencing pipeline is a new field. In this immature state of research, the detail understanding of the proposed solution (e.g., easy to interpret the result and how the *machine learning (ML)* model works) is also very important besides understand the problem in details. Unfortunately, on many occasions we are guilty of proposing sophisticated

solutions, e.g., black box *ML* models, which might also provide us with the best results but it become really hard to explain the solution at the end. *Explainable AI solution (XAI)* is an idea of using *Artificial Intelligence (AI)* and *ML* models that are easy to interpret compared to black box type of *ML* models. To overcome the limitations, initially we used Case-based Reasoning *CBR* as an *XAI* to solve our classification problem to detect an attack in a DNA sequencing pipeline. This limitation is related to **Challenge C4**.

Chapter 2

State-of-the-art

We consider few areas of research as primarily relevant fields to this PhD work, which are Bacterial motility and molecular communications, Security in Biotechnology, where we focus on two particular security attacks (Trojan Attack and Buffer overflow exploits) and two specific machine learning models (Deep learning, and Case Based Reasoning as an explainable AI model) as a solution to counter such attacks. In this chapter we will provide relevant background on these relevant areas.

2.1 Bacterial Motion and Communication

Here we discuss recent works related to mobility models and molecular communications in *IBoNT*.

The mapping of various components of biological cells to parts of a computer system is described in [4] to show how a biological device can act as a typical IoT machine. It describes various models and systems of molecular communications, such as Ca^{2+} signaling, molecular motor communications, communications through chemotaxis bacterial conjugations and long-distance communication through hormones. Various communications like short, medium and long ranges are also mapped to classical communication theory. How biological nanonetworks inside the body based on bacterial communication can interface with the cyber-Internet is also discussed. Finally, various opportunities and challenges of $IoBNT$ using molecular communication are addressed here.

The current state of theoretical models of molecular communications and experimental developments of membrane nano tubes, nano tubes formation in bacteria and artificial neural networks have been discussed in [8]. The potential opportunities from the state of the art works and challenges for the future woks have been stated here. The vision of building *Body Area NanoNetwork* and how the mentioned technologies can help in achieving that in future is also discussed, where components

such as nano particles can be used to transport information as part of treatment strategies in the future.

The social behaviour of bacteria and the opportunities from their co-operative behaviours and the challenges that arise from their social behaviour are described in [31]. The data transmission and reliability corresponding to these behaviours and various concentrations of chemoattractant are compared here using simulations, where the results show that the cooperative behaviour increases the reliability in communications. This shows how inherent natural properties of bacteria can be used to either improve or play a negative effect on the communication performance. This thesis explores how behaviour of movement using chemotaxis can help direct bacteria movement to a certain location to transfer bits of information.

Multi-hop directed or random protocols are discussed in [9] as two options for molecular communications where bacteria are carrying information encoded into the plasmid DNA. In this system, nanomachines will release the bacteria when it is senses diseases or infections. The bacteria are delivered to the gateway, either with or without the use of a relay nano machine in the environment for single hop or multi-hop communications. For multi-hop and directed communications, the conjugations process occurs inside the relay to copy codes from the plasmid DNA of the donner cells, where other types of bacteria were inside the relay nano machine already. In the case of multi-hop random communication, the conjugation occurs outside while relay nodes also releases its collections of bacteria. In Multi-hop directed nano-network, the chemoattractant is used to attract the bacteria towards the relay nanomachines. Inside both the relay and gateway nodes, nutrients are used to provide a friendly environment for the survival of bacteria, where antibiotics are used in the gateways to remove unwanted bacteria that contain plasmids with undesirable encoded information. The received bit ratio, delay, and number of conjugation are analysed for both networks for various times and distances. While this approach allows multi-hope communication, there are uncertainty that lies around how certain bacteria will conjugate to help transfer the encoded information. This is a complex stochastic process that can lead to uncertainty in the delivery of information.

A novel approach of target tracking using bacteria bio-sensor is proposed in [66] based on directed diffusion of bacteria. The bacteria are moving inside the search space with a certain velocity. It is assumed that the bacteria will be genetically engineered to emit two types of molecules, attractants or repellents, and they will move toward or move away by sensing the concentration of the attractants and repellents, respectively. The mobility of the bacteria is modelled here based on the Brownian movement and their chemotaxis nature. Furthermore, bacteria generally releases repellents but will start to release attractants for a certain period of time if they come close to the target. This way, the bacteria are spreading themselves

using repellents and moving towards the target using attractants collectively if any of them come close to the moving target. The work did not consider the three dimensional movement of bacteria and multiple moving target.

Decision making of biological cells and responding to the stimuli are described as an stochastic process in [76, 7]. The decision making of cells depends on the various parameters of stimuli like concentration, temperature and noise in the environment. Distortion function of Information Theory is used to estimate the cost of making certain decision or response to certain stimuli. The result shows significant improvement in error reduction. Based on the above description of using bacteria to carry encoded plasmid DNA, there is considerable decision making based on sensing the chemoattractant as well as controlling the motility process. These properties are also considered in this thesis when the bacterial cells mobilizes between locations.

The above mentioned works showed the potential of using bacteria as either nanomachines or supporting information transfer between nanomachines and how simple communications in a very simple network can be established. Encoding of the data inside a plasmid is proposed where we can send more than one bit of data. This will require the insertion and alteration of the plasmids, which at times can cause unwanted behavioural and functional changes in the bacteria. However, the system requires other devices such as relays that in certain cases will need to embed bacteria inside them. Beside this, it is important to address how to ensure the installation of sufficient number of relay devices in specific locations to assist bacterial motility. In this thesis, we address an alternative approach for transferring data by sending more than one bit from a transmitter to receiver without data encoding inside the plasmids.

2.2 Security in Biotechnology & DNA Sequencing

In this section, first the background of relevant security attacks considered for this thesis, and the basics of DNA, protein and RNA sequences will be described. Then, recent works on *biohacking*, *cyberbiosecurity*, *bioethics* and cryptography using the *DNA* sequence will be discussed as these are relevant to this PhD thesis and will assist in understanding the contributions of the thesis.

2.2.1 Background of Security Attacks

Buffer Overflow Vulnerability Exploit

The buffer overflow vulnerability exploit is a type of input validation attack [86] and also very common [81], where the attacker sends input to exploit vulnerabilities. This form of attack has increased significantly in the past few years [82] and researchers

are actively finding solution to detect such attacks. In recent past, machine learning algorithms are applied to detect such a problem, for example in [81] a decision tree based *ML* technique using various software metrics based on data flow in the code is proposed to detect vulnerabilities of the software. The authors also considered other types of machine learning algorithms in their works, where using a decision tree successfully detected vulnerabilities from software code developed in C++ and Java programming languages. Meanwhile, we see that buffer overflow attack is also possible in *IoT* devices and that they can be hijacked for remote access. The work in [22] demonstrated how buffer overflow attack in a scenario of using *IoT* firmware, which is used in popular smart TVs. Works have been done to mitigate such buffer overflow attacks and the work presented in [82] is an example of such works where the authors proposed a run-time solution for monitoring run-time memory space using a table called variable record table. To figure out the vulnerability in binary programmes and to generate exploits, a tool has been proposed in [97]. Such a tool can be really useful as far as the buffer overflow attack is used for testing a software in advance.

Trojan Attack

Hardware Trojan (*HT*), where a Trojan is planted inside the integrated circuit to change the behaviour of the device by delivering the payload and denying functions when the trigger is activated, is an interesting area for malware research [70]. HT detection strategies are broadly categorised into two types, test data generation for validation, also called logic testing, and side channel analysis [70]. A data flow graph from the register transfer level codes are analysed using a tool called *GNN4TJ* proposed in [99] for HT detection. A game theoretic framework is proposed in [58] for logic testing in HT detection to reduce the number of possible test inputs. The results of side channel analysis using delay measures for HT detection are improved by generating test data that can meet the conditions to activate the triggers [56]. Machine learning such as deep learning techniques are also applied in HT detection for both logic testing [84] and side channel based analysis [70]. HT detection has also been applied to the IoT. A temporal thermal information is used to detect HT as the execution of the Trojan will increase or change the power consumption significantly [28]. The Trojan can be implanted at both the software and hardware levels. Malware, which can be Trojans that are implanted at the API levels, can affect mobile software and even machine learning codes. Recent research works such as malware detection at the API level in Windows [78], android malware detection [46] and backdoor Trojan detection in deep learning [29] are such examples.

2.2.2 DNA, Protein and RNA sequence

Four nitrogen based molecules called nucleotides, which are *Adenine*(A), *Guanine*(G), *Cytosine*(C), *Thymine*(T), form a double stranded shape known as *DNA*. A bond between the nucleotides of two strands is made of a weak hydrogen bond and a bond between consecutive nucleotides of the same strand is made of a comparatively stronger sugar phosphate bond [6] (see Figure 2.1). The complementary parts of A,T,C and G are T,A,G and C, respectively. For example, if the nucleotide in one strand is A then it will be bonded to T in the other strand using a hydrogen bond. The DNA sequences are read from left to right (also called 5′ to 3′) or right to left (also called 3′ to 5′). Two sequences from these two reads are called reverse complements of one another [6]. If we have either of the sequences, then its reverse complement can be derived by reversing the sequence and using the counterparts of each nucleotide of the reversed sequence. For example, assume that the sequence from left to right is $AGTTCAGT$, then the reverse complement from right to left will be $ACTGAACT$. The device performing the read process is called a sequencer and is available on the market. On the other hand, *RNA* is another common type of sequence. An example of RNA application is its use for understanding disease dynamics [98]. Different techniques are used for RNA sequencing [98, 68]. Another very common sequence used in bioinfomatics is protein sequences. In total, 20 amino acids are used in a sequence (Figure 2.1(a)) and it can be derived from a DNA sequence using an asymmetric mapping table (Figure 2.1(b)). It is known that the functionality of a gene is defined by the structure of the protein and it is still impossible to predict the protein structure from the protein sequence [6].

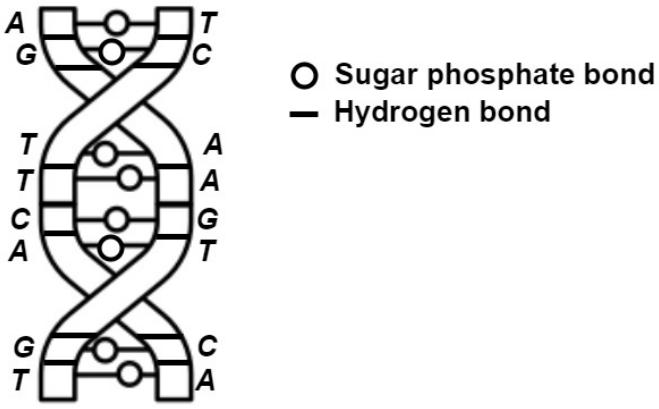


Figure 2.1: A DNA snippet to depict hydrogen and sugar phosphate bonds.

In recent past, both DNA and protein sequences have been used in many research works for diverse applications. For example, protein sequencing is used to separate healthy and cancer sequences in [24], while a hashing and *ML* based classification

Table 2.1: Protein Symbol and DNA Protein mapping

(a) Protein symbol and corresponding Amino Acid [102]

Symbol	Amino Acid	Symbol	Amino Acid
A	Alanine	C	Cysteine
D	Aspartic Acid	E	Glutamic Acid
F	Phenylalanine	G	Glycine
H	Histidine	I	Isoleucine
K	Lysine	L	Leucine
M	Methionine	N	Asparagine
P	Proline	Q	Glutamine
R	Arginine	S	Serine
T	Threonine	V	Valine
W	Tryptophan	Y	Tyrosine

(b) DNA to Protein Mapping [6]

DNA Code	Protein	DNA Code	Protein	DNA Code	Protein
GCA	A	GCG	A	GCT	A
GCC	A	TGT	C	TGC	C
GAT	D	GAC	D	GAA	E
GAG	E	TTT	F	TTC	F
GGA	G	GGG	G	GGT	G
GGC	G	CAT	H	CAC	H
ATA	I	ATT	I	ATC	I
AAA	K	AAG	K	TTA	L
TTG	L	ATG	M	AAT	N
AAC	N	CCA	P	CCG	P
CCT	P	CCC	P	CAA	Q
CAG	Q	CGA	R	CGG	R
CGT	R	CGC	R	AGT	S
AGC	S	ACA	T	ACG	T
ACT	T	ACC	T	GTA	V
GTG	V	GTT	V	GTC	V
TGG	W	TAT	Y	TAC	Y

techniques are used to classify proteins in [15]. *DNA* sequences and *Convolutional Neural Network (CNN)* to detect *N6-methyladenine* in rice is used in [55], while DNA sequences and deep learning are used to identify *N4-methylcytosine* in [94].

An interesting process of ransomware detection using DNA sequencing is proposed in [43]. The features will be extracted from the ransomware dataset and then the selected features will be converted into DNA sequences. The DNA sequences will then be converted into $k-mer$ vectors, and then ML will be applied to categorise the file into "Ransomware" or "Goodware". The article [35] shows how *DNA* sequences and few machine learning algorithms can help classifying cancer patients.

2.2.3 Bio-hacking

In short, *Bio-hacking* is a technique of manipulating the biological process to improve human's physical and cognitive ability[61, 101], where examples can include quick weight lose by changing food consumption habits or enhancing the performance of an organ in the body by implanting devices. *Nutrigenomics* is one type of *bio-hacking*, which is based on the change in the habit of food consumption. The argument is that the food can effect certain genes. This belief comes from research that has been shown to reduce *DNA* damage. [61, 73]. Another form of bio-hacking considers cells as hackable devices and takes the advantages of the progress in synthetic biology, which is most relevant to this thesis although we do not focus on changing functionalities of the cells, but rather on how we can do cyberhacking. This is also known as *DYIbio* [13]. Finally, the third form of *bio-hacking* technique is when devices are implanted into the body of the *citizen scientist* (also called *grinders*). Examples include implanting a microelectrode array in the body to control a robotic arm, implanting thermal sensors under the armpit that can be used to monitor the temperature of the body, and implanting *RFID (Radio Frequency Identification)* devices to control other devices [101]. However, these devices are foreign agents to the living body. Therefore, these devices are coated with materials that will not be affected by the immune system of the body. This new interdisciplinary research of (*bio-hacking*) creates plenty of opportunities and the need for more research that link to many fields. For example, in [88], the authors proposed the concept of "*bionic manufacturing*" to build new type of bionic systems for the next generation micromotors and sentient microbots, which can help in environment monitoring for example. Indeed *biohacking* will help in connecting the human body to internet, human machine interactions(*HMI*), human machine interfacing, health monitoring, environment monitoring applications, but it will bring lot of security challenges as well. Moreover, the devices used for *bio-hacking* can cause several health issues and even lead to death. For example, Deep Brain Stimulations (*DBS*) from Deep Brain Implant (*DBI*) devices are used for providing electronic stimulating to the nervous system to treat patients of Parkinson diseases, neurological disorders, epilepsy, or movement disorders. In [80], the authors argue that a perpetrator can hack im-

plantable devices to generate stimulations to induce pain and that can cause death. They have also proposed a deep learning based attack classifier for the *DBS* in their research work. Therefore, we have to consider the security issue of *bio-hacking* very seriously. A survey was conducted with the subject matter experts of the emerging threats of technologies, where the study concluded that *bio-hacking* and *HMI* with significant defence incapabilities [79]. It is good that concerns are raised about possible security threats from *bio-hacking* is raised. However, this area of research has not received enough attention in terms of security vulnerability from synthetic biology when using bacteria for bio-hacking. We should not forget that genetically engineered bacteria has the potential to be used as bio-compatible devices and can be considered for bio-hacking in the future and for health-related applications such as targeted drug delivery [71].

2.2.4 Cyberbiosecurity

Cyberbiosecurity is an emerging interdisciplinary field that mainly considers the security threats coming from the three inter related disciplines, namely, Biosecurity, Cybersecurity and Cyberphysical Security [40], as depicted in Figure 2.2. This emerging field has drawn a lot of attention recently due to economical, health and safety and even for the national security reasons. The article [59] described why the biopharmaceutical sector should consider *cyberbiosecurity* measures and how it can be affected by security threats and economic losses. Today, the manufacturing process such as quality control and integrity checks are all done by automations and AI systems. Therefore, the whole process can be affected if the machines involved are compromised. As a result, the manufacturer can end up with huge financial loss as the production process can be interrupted or there could be waste of batches of produced medicines due to quality issues. To understand how financially worthy the matter is, we can consider the article [63], where the authors discuss that the bioeconomy of the United States is estimated at $4 trillion annually and that is nearly 25% of their GDP. Furthermore, in the worst case, pathogens can be introduced intentionally to cause harms to patients, common citizens and even the people in the manufacturing plants. Therefore, the authors in [63] emphasized on addressing research, tests, education, technologies, standard practices, and policies for *cyberbiosecurity*. Meanwhile, the consequences due to the culture of having open data in the relevant research fields is described in [40]. For example, people can be blackmailed by their medical history, or a particular pathogen can be utilised to develop a biological weapon targeting a community, race, or nation, who are found vulnerable to that particular pathogen as a whole. Therefore, risks like this can be turned into a national level security challenge. On the other hand, we should consider cy-

berbiosecurity for applications related to DNA Synthesis, which is the focus of this thesis. For example, we can place an order to synthesise a DNA sequence [64] using web applications. However, in such web applications, the ordered *DNA* sequences can be manipulated by means of a man-in-the-middle attack as described in [77] and this can end up in an order request to create pathogens instead. Although the authors did not describe how someone can perform such an attack, it is sufficient to understand the urgency of addressing such a issue. Furthermore, the authors also mentioned challenges like not having relevant up to date US National health guidelines and the difficulties with the manual inspection of DNA, which can cause the attacks. Unfortunately, all the above mentioned works are mainly focussing on the security threats coming from the cyber-world to the bio-world, while we believe it is also highly important to think about how security threats can come from bio-world to the cyber-world. An example of this is the problem investigated in [64], where we have seen that DNA can be synthesised to compromise a *DNA* sequencing pipeline.

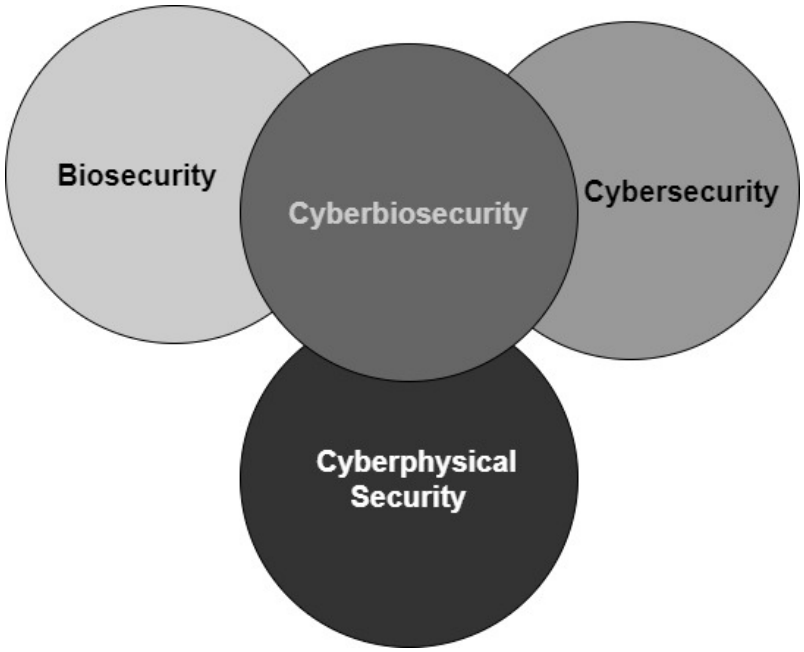


Figure 2.2: Cyberbiosecurity is a combination of three relevant security fields [40]

2.2.5 Bioethics

The term *'Bioethics'* was first introduced by Van Rensselaer in 1970 [25]. In short, Bioethics is a process to determine whether the activities involved in our research is morally correct. It can be done based on answering a few ethical questions or following a set of guidelines in dealing with biological specimens. However, this is a process that involves theoretical frameworks, interviews, questionnaires, existing laws, pri-

vacy policies, biotechnology, medical and health issues, etc. Recently, a number of works have been conducted to improve the process. For example, the impact of biotechnology on *bioethics* and the raised issues corresponding to this are discussed in [91]. Another example can be the demonstration of a potential digital tools to take moral decisions, where the authors utilized technological driven methodologies rather than relying on empirical research only [72]. In [20], the authors introduced experimental philosophical bioethics as an emerging discipline as they believe experimental philosophy can contribute to the improvements in *bioethics* research as it did for other disciplines. A distance learning for the students based on Virtual Reality(VR) is suggested in [30] to teach bioethics and this is by providing interactive sessions and a relevant virtual world to help them to take biotechnology related decision in situation with many dilemmas and high urgency. For example, this may include situations like vaccinations related decision in COVID-19 pandemic situation. Bioethics related work have also been developed for DNA based applications. For example, in [96], the authors argued that a separate dedicated bioethics framework for forensic's type work beside having the comprehensive common bioethics framework for medical related work. The authors think that though medical and forensic have many things in common, the purposes of each are different. For example, medical reports are used for diagnosis, whereas forensic reports are used for testimony purposes for victims or suspects by law enforcement agencies and judiciaries. The *DNA* plays an important role here. On the other hand, the *in vitro* gene modification technique using *Clustered Regularly Interspaced Short Palindromic Repeats (CRISPs)* brings a wealth of opportunities and helps to save people's lives, but it also raises many ethical questions at the same time. CRISP is a process of utilizing the existing anti-viral behaviour in bacteria to edit genes so that we can insert our preferred DNA snippets in them for altering the existing functionalities or to introduce new functionalities, where CAS protein play a vital role in the edit operation. Any arbitrary data can also be stored inside DNA using CRISPR-CAS [85]. Therefore, the challenges towards *bioethics* for using *CRISPs* are discussed in [25]. The bioethics on the consent and privacy point of view are analysed for commercial DNA tests and *Investigative Genetic Genealogy (IGG)* that uses it for criminal investigations [17].

2.2.6 Cryptography using DNA

DNA have been considered as a storage device in recent research [18]. Ensuring the security and integrity of the information have also become very important, which has received considerable attention recently [83, 65]. Cryptography techniques such as encryption and steganography are applied to DNA encoded information [69].

Steganography is a technique to hide information of one form into another, rather than scrambling the information to turn it into a human unreadable or non understandable form. An example is the process of hiding a secret text inside a picture. In the recent past, we also saw that DNA is used in cryptography to secure information. For example, DNA cryptography and dual hyper chaotic map are used to secure sensitive and confidential medical images [3]. A technique using biological operations has also been proposed in [42], where transcription and translation in combination with deep learning was used for encryption and decryption. The key for the encryption and decryption is generated using *Needleman Wunsch (NW)* algorithm. The use of DNA-based encryption keys was proposed in [57]to conceal the cloud storage. Interestingly, all the state-of-the-art works are considering cryptography to protect the information from unauthorised access. However, a question we have asked is - " *What about utilising cryptography by the hackers to dodge surveillance that remains unidentifiable for hacking activities?"* Ransomware can be considered as a use of cryptography by the hackers, but this is more a part of the exploit rather than using it as a tool to masquerade or cover the attack. In this thesis, we have considered the use of cryptography in DNA by hackers to make the detection of attacks harder, while ensuring that DNA Synthesis can still be performed.

2.3 Deep Learning

In this thesis, we also used Deep Learning to perform detection of encoded DNA that is part of bio-hacking. Deep Learning is a machine learning algorithm that is used in speech recognition, image classifications, autonomous driving, etc. Therefore, it has drawn interest in solving classification problems in various areas and it is considered as "a *de facto* classification techniques" [53]. A few examples of diverse applications are industrial defect detection [51], attack and intrusion detection in networks [90, 16], cyber security [14] and health care [23] including radiology [100].

Unlike typical neural networks, *Deep Neural Networks* (*DNN*) have more hidden layers between the input and output layers. The number of nodes in the output layers is equivalent to the number of classification items. *Convolutional Neural Net*(*CNN*) and *Recurrent Neural Net*(*RNN*) are two broad categories of *DNNs* [16]. The main difference between *RNN* and *CNN* is that the output of each hidden layer sends a feedback to its previous layer. *RNN* is good when contextual information is necessary for correct classifications, such as in the use of language translation and stock price prediction. In language translation, the correct translation depends on the contexts or previous sentences. Similarly, stock price prediction also depends on the earlier stock prices or the predicted stock prices. On the other hand, *CNN*

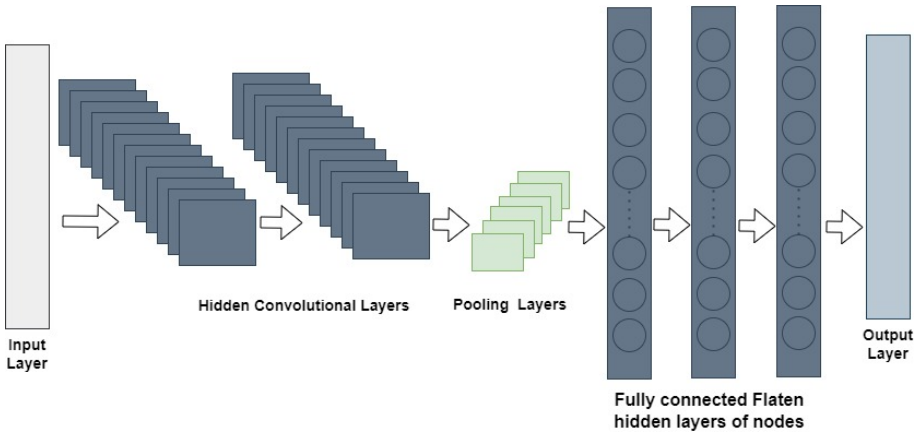


Figure 2.3: Example architecture of a *CNN* model.

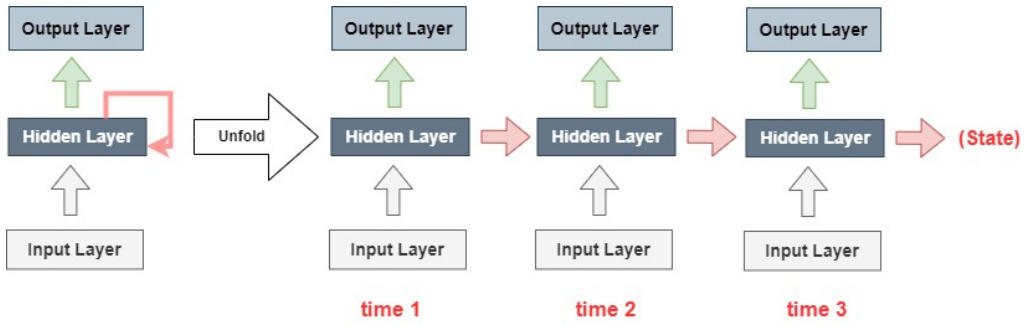


Figure 2.4: Example architecture of a *RNN* model.

can be applied when the classifications or predictions does not rely on the previous predictions, such in applications of classifying images. The examples of typical *CNN* and *RNN* models are presented in Figure 2.3 and 2.3, respectively.

CNN and *RNN* are also applied in the applications where DNA, protein, and RNA sequences are used. For example, A *CNN* based model is used for protein family classification using DNA sequences [103]. A model combining for both *CNN* and *RNN* is used to predict DNA-binding proteins from protein sequences [33] and to predict DNA and protein binding using the DNA sequence [104]. A similar approach is also proposed for classifying the chromosomal DNA sequences [19] and *pre-miRNA* sequences [89].

Different types of deep learning are being used for different types of malware detection [87]. The growth of attacks in *IaaS* using various malware will be very devastating as there will be a number of similar types of resources, e.g., virtual machines, will be created in it. If one machine can get infected by a malware, then other similar machines can easily be affected. The malware can be detected by analysing various behaviours, such as CPU, memory, and disk space usages. Both *CNN* models (e.g., [60]) and *RNN* models (e.g., [44]) have been proposed to detect such malware and attacks in *IaaS* using those behaviours. As *CNN* shows promising

results for various applications where DNA sequences are used and also to detect security issues, hence it is considered as a detection technique for our thesis.

2.4 Case Base Reasoning(CBR)

Case Base Reasoning(*CBR*) is a branch of supervised machine learning, where each case in the base is defined by a problem space and the corresponding solution space. The basic idea behind *CBR* is based on the principle of *"similar problems has similar solutions"* [50]. That means that if we get a new problem, then we can solve it by adapting the solution of a similar problem that happened in the past. In *CBR*, the problem can be formulated by a set of features and the solutions can be formulated by a single or a set of outcomes, results or classifications. The *CBR* cycle investigated in [1] is shown in Figure 2.5, where the steps are *retrieve, reuse, revise* and *retain*. In the *retrieve* phase, a past case will be selected from the case base based on the highest similarity between the problem space of the new case and the past cases. The feature is extracted from the new case first before the retrieval. A similarity function is used to compute the similarities between the problem space of all past cases and the new case. Example of similarity algorithms can be Euclidean distance and fuzzy similarity. Then the solution of the top one or few cases will be considered and used for the new problem in the step. Therefore, the step is called *reuse* as the past solutions are used again. In the *revise* step, the reused solutions are verified to know whether they were well enough to solve the new problem. Necessary adaptations are performed on the past solutions to solve new problems. Last but not least step is *retain*, where the learned knowledge is stored in the case base for future use. So, the new case with its successful solution is stored in the case base in this stage. In our research, we will mainly focus on the *retrieval* step of *CBR*.

CBR is proposed as a supervised machine learning technique for various applications like decision support system in health, diagnosis system for vehicles drivers[10], stress diagnosis using various physiological signals[12], diabetics diagnosis[21], post operative pain management[2], predicting recurrent status of liver cancer[75], business workflows [62], construction management[34] and traffic controls in signalized intersections[54]. Moreover, unlike *DNN*, *CBR* can be used as an *Explainable Artificial Intelligence* (*XAI*) machine learning model [48, 93].

2.4.1 Summary

The state-of-the-art research show how promising bacteria are as a nano machine, programmable device and as a data carrier by encoding it in the DNA, while they can also be used for sending data even without encoding it inside the DNA. To

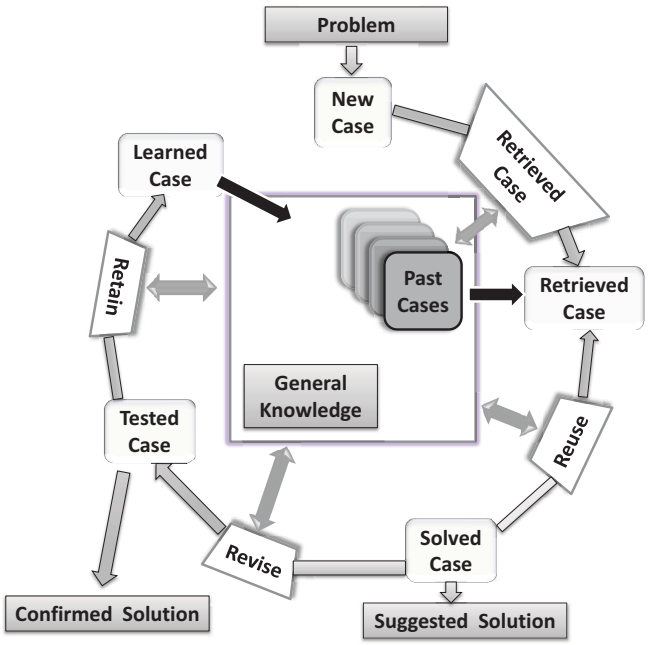


Figure 2.5: Case Base Reasoning System [1]

help the course toward the improvement of *IoBNT*, we need to address how we can send more than a single bit of data at a time, while not encoding the data into bacterial DNA. Again, in a recent research [64], we have seen how a DNA sequencing pipeline can be compromised using specially designed DNA. Therefore, it has raised the alarm regarding the security threats through similar attacks in the field of *IoBNT*. DNA and bacteria are very relevant to cyberbiosecurity, bio-hacking and bioethics. Significant progress has been made in these fields, but unfortunately, we hardly found any further research that addresses such security threats. As far as countermeasures for such attacks are concerned, choosing the detection algorithm is very important. In state-of-art research, we see that *CNN* has become very popular to solve detection problem in cybersecurity. Moreover, DNA sequences can be used as input data in *CNN* to solve detection problems, e.g., the works described in [55] and [94]. As *CNN* showed promising results to solve the detection problem using DNA data, we believe that it has the potential to play a role in the detection mechanism in cybersecurity, where DNA will be the means of attacks. Finally, if *explainable AI* is preferred as an alternative solution to *CNN*, then *CBR* can be used, as it was previously used for many other detection problems.

Chapter 3

Research Summary

In this chapter, the research questions will be formulated first based on the research scopes, challenges and limitations discussed in the previous chapter. Then the approaches to answer the research questions, the validation techniques and finally the contributions will be presented.

3.1 Research Objectives

To address the challenges discussed, the following three research questions are formulated for this thesis. The research questions are listed according to the order we believe they should be addressed.

- ***RQ1. Multi-bit Data Transfer using Bacterial Nanonetworks:*** *How can we improve the performance of data transmission using bacteria to exploit their traits while avoiding complex data encoding in the plasmid DNA?*
- ***RQ2. Bio-Hacking Security attacks based on DNA Encoded Data:*** *What can be a new type of attack in the DNA sequencing pipeline? Will it be possible to perform attacks without considering the limitations of the underlying OS and hardware? How feasible and sophisticated can such attacks be?*
- ***RQ3. Countermeasure for Bio-Hacking Security Attacks:*** *Finally, what can be a solution to mitigate a possible security attack in the DNA sequencing pipelines?*

3.2 Approach

We address the research questions one by one. Our approaches to address the research questions are given below.

- **RQ1. Multi-bit Data Transfer using Bacterial Nanonetworks:** To address research questions, we proposed a multi sender and receiver based

data transmission utilising mainly three traits of bacteria, which are motility, quorum sensing and bio-luminescent characteristics. A particular protein called green fluorescent protein (GFP) production is required for emitting light, which will require sufficient number of bacteria to make the light visible. The bacteria emits a particular chemical message and also senses it to determine the population in their vicinity. If that chemical density cross a threshold value then it triggers the GFP production and this process is called quorum sensing. Last but not least, the bacteria will have flagellas if it has a plasmid protein F^+, which helps them to swim. Otherwise, the bacteria cannot swim, and they remain stationary.

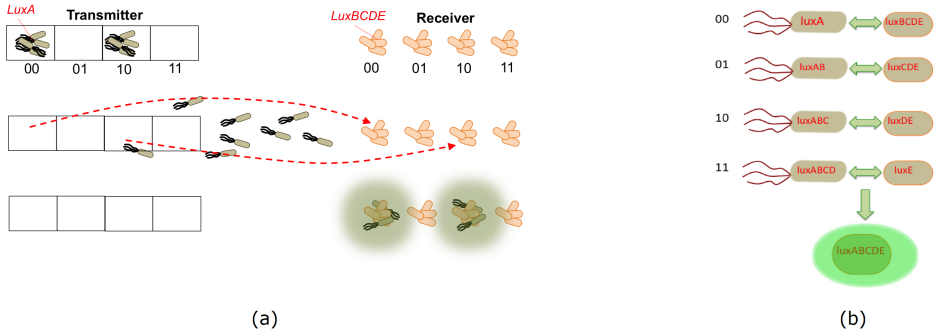


Figure 3.1: Distributed transmission of bacterial nanonetworks, (a) bacteria sent from transmitter to receiver, and (b) how the sequence for GFP gene is divided [92].

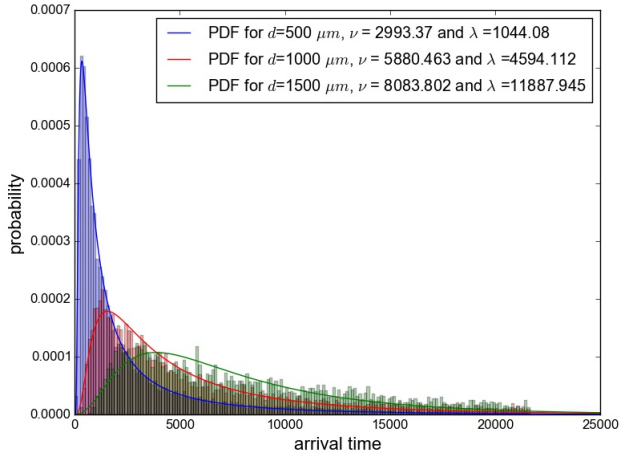


Figure 3.2: Probability distribution function (PDF) of the arrival time of bacteria for various distances in the nanonetwork [92].

Using the properties mentioned above, we come up with an idea of using genetically engineered bacteria for four senders and the corresponding receivers as shown in the Figure 3.1(a), where the bacteria will only be able to complete

their DNA/Protein sequence required for *GFP* production if conjugations occurs between the cells that belongs to the corresponding senders and receivers as shown in the Figure 3.1(b). The bacteria of the senders can swim but the cells in the receivers are stationary. This way we can send two bits at a time. For example, if we want to send two bits, **01**, then we need to release the bacteria from the corresponding sender. The bacteria will swim and reach the four receivers and conjugate with them but it will only complete the sequence in the receiver **01**. When there is a sufficient number of bacterial cells with a complete gene sequence, they will start *GFP* production and the presence of light will be detected and read as receiving bits **01**.

Table 3.1: Simulation parameters and Results

(a) Simulation Parameters.

Parameter Name	**Value**
Temperature	305 K
Viscosity	2.7×e-3 Pa s
Radius of the bacteria	1 micrometer
Flagella force	1 pN
Mean time to end a run	0.86
Mean time of end a tumble	0.14
The maximum tumbling angle	180 degree
BOLTZMANN constant	1.38×e-23

(b) Fitted Inverse Gaussian Parameters.

Distance (μm)	**Cathcment Area** (μm^2)	ν	λ
500	100	2993.37	1044.080
500	200	2971.459	1092.047
500	300	3033.642	1113.672
1000	100	5880.463	4594.112
1000	200	5726.056	4651.549
1000	300	5742.775	2532.088
1500	100	8083.802	11887.945
1500	200	8017.303	11568.980
1500	300	8126.618	11760.664

To estimate the bit transfer time and compare the performance with other state-of-the techniques, it is necessary to compute the *first passage time*, which is the time from releasing the bacteria and detecting the light in the corre-

sponding receiver. We executed simulations using the parameters listed in Table 3.1 (a) and (b) to estimate *first passage time*. A 3D simulator called BSim [26] is used for our bacterial motility simulation. The parameters listed in Table 3.1 (a) are related to the environment where bacteria will swim, the force generated by the movement of the flagella, the probability of the direction of spins of the flagella, angle of every tumble motion and the constant used in the function that define the bacterial movement. We have chosen the default values set by the simulator for these parameters. We measure the time to reach from a source to a target by the bacteria considering various distances between the source and target along with the catchment area of the target. The target is not a single point but an area with a radius. The distances and catchment areas used in our experiment are listed in Table 3.1 (b). We have observed that the first passage time for our data transmissions follows an *Inverse Gaussian Distribution* (Figure 3.2), which can be expressed as

$$f(t) = \left[\frac{\lambda}{2\pi t^3}\right]^{1/2} \exp\left(-\frac{\lambda(t-\nu)^2}{2\mu^2 t}\right), \tag{3.1}$$

where the coefficients λ and ν depend on the run-and-tumble parameters of the bacteria, the distance between the transmitters (senders) and the receivers, and the receiver's volume. To evaluate the performance of our proposed technique, first the first passage time is estimated for various distances and catchment areas. Then using that we compute the achieved rate (number of bacteria reached to the targeted receiver) and the bit error probability (converted from probability of error) with respect to average transmission power per bit. The average transmission power is the average number of bacteria released to send single bit of data. Finally, we have compared the error rate of our proposed model with other approaches and our results have outperformed other system models. Further details are available in Appendix A.

- **RQ2. Bio-Hacking Security attacks based on DNA Encoded Data:** A state-of-the-art work [64] showed how a DNA can be designed and synthesised to exploit a buffer overflow vulnerability in a DNA sequencing pipeline. The buffer overflow vulnerability exploit is a vulnerability exploit at the hardware level and depends on the underlying hardware and memory management of the operating system and software.

  To analyse the possibility of different types of vulnerability exploits or attacks, more specifically a vulnerability exploits or attacks at the software level, we consider a targeted Trojan attack. The scenario considered for such an attack is shown in Figure 3.3, where the Trojan will disguise itself in a software tool of the DNA sequencing pipeline. Bioinformaticians will use the tool by using

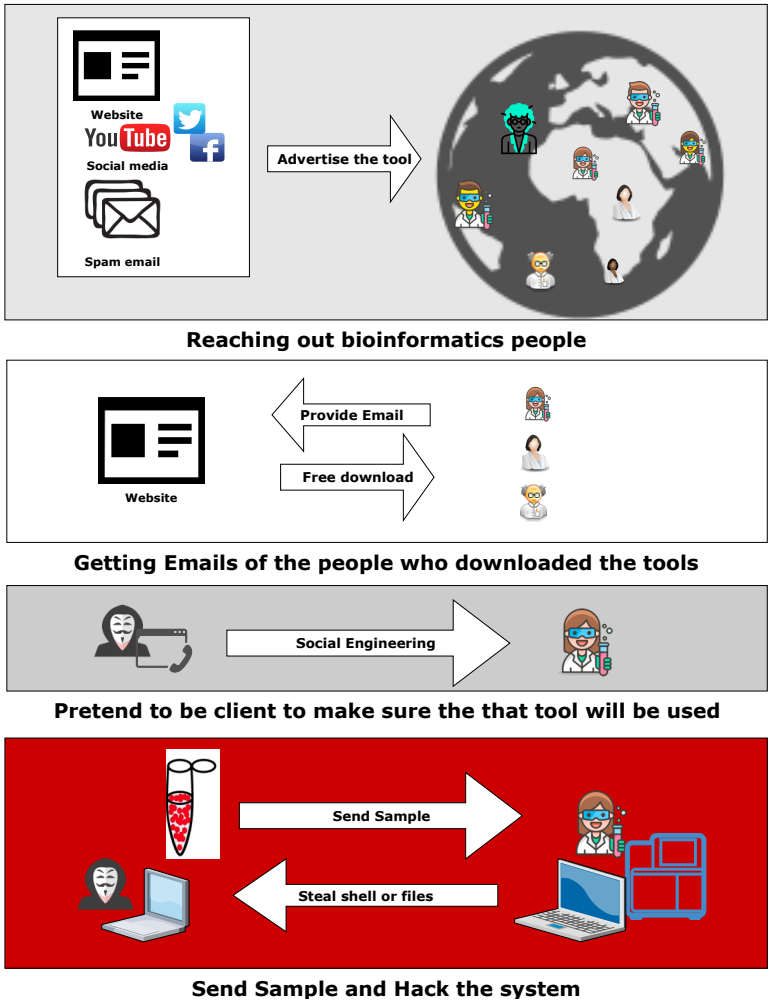


Figure 3.3: Social engineering scenario for a targeted Trojan attack used in biohacking.

Social Engineering. *Social Engineering* is a mechanism to induce and trap people to disclose private information [32]. In our considered scenario, the Trojan will remain dormant, and the tool will perform its usual legitimate activities. The Trojan will be activated if and only if the tools get a specially designed DNA sequence where the signal to trigger is encoded. This scenario gives perpetrators the opportunity to further split the encoded message and apply encryption and steganography techniques. The idea behind choosing such an attack scenario is to take full advantages (as it works at the software level) of designing an attack, where it will be so hard to detect and to trace or identify the hacker. The detailed process of how DNA will be synthesised to encode the Trojan payload trigger to activate it and disguise it in bacteria and finally sent to the targeted DNA sequencing pipeline is shown in Figure 3.4.

The payload encoded into the DNA has two-fold application in our attack scenario. It not only works as a trigger sample, but is also used to embed the information required to connect the machine running the tool to a remote location. This can potentially lead to transferring files from that machine to that remote location. The example in which the information embedded into the DNA is the IP address, web address and port number of the remote machine. Furthermore, the payload can be fragmented and encryption and steganography techniques can be applied to the information as described in [37]. However, there can be error in the read process during sequencing and if it effects a crucial nucleotide of the sequence then it can effect the success rate of such an attack. We propose the following equation to estimate the success rate, which is the equation used for the bit error estimation in communication theory.

$$prob_r = (1 - prob_e)^{N_p} \tag{3.2}$$

For a given read error probability $prob_e$, the probability of successful retrieval of the payload is $prob_r$, where the payload size in bit is N_p. The equation is extended as follows considering the possible mutation in favour, and steganography parameter, which is key $key2$ that represents the number of retention positions.

$$prob'_r = (1 - prob_e)^{\frac{2}{3}x} \text{ , where } x = \frac{2 \cdot N_p}{key2 + 1}. \tag{3.3}$$

We have compared the calculated success rate estimations for various read errors with the simulated results, which verified our estimation assumptions. Various payload sizes, fragment sizes and retention numbers are considered for

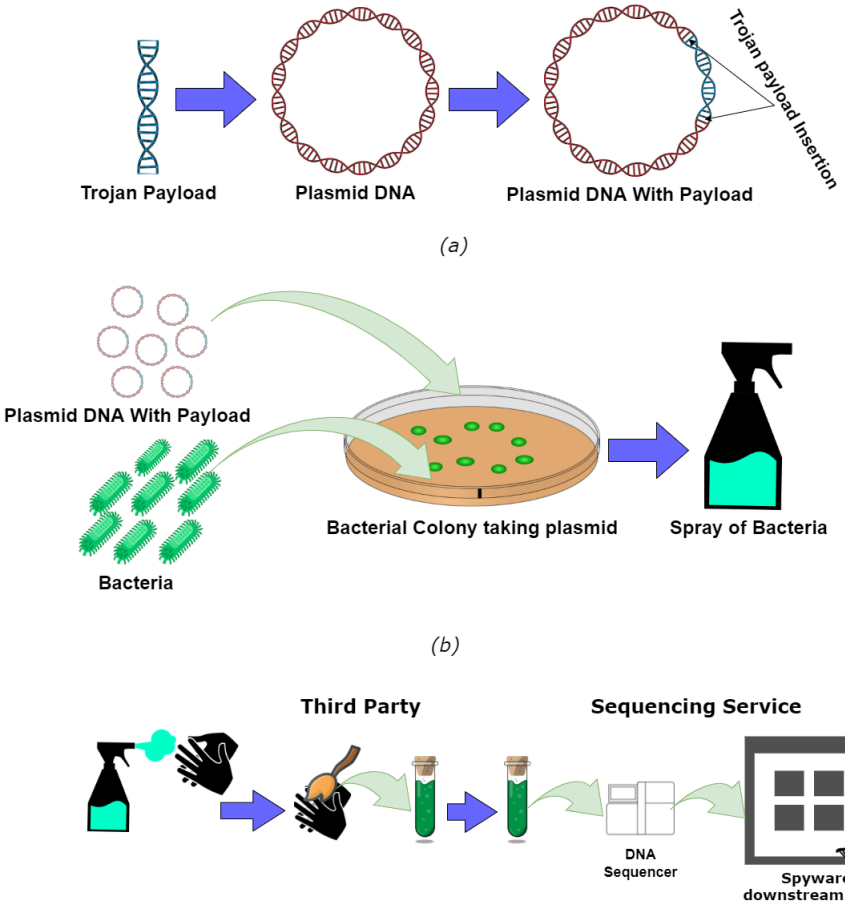


Figure 3.4: Steps in the Trojan attack scenario for bio-hacking: (a) trigger message payload is encoded into DNA sequence snippet and then inserted into plasmid DNA, (b) plasmid DNA is inserted into bacteria and (c) bacteria sample is collected by 3^{rd} party and send for DNA sequencing and one of the Trojan infected tools in the sequencing pipeline is activated to compromise the system [36].

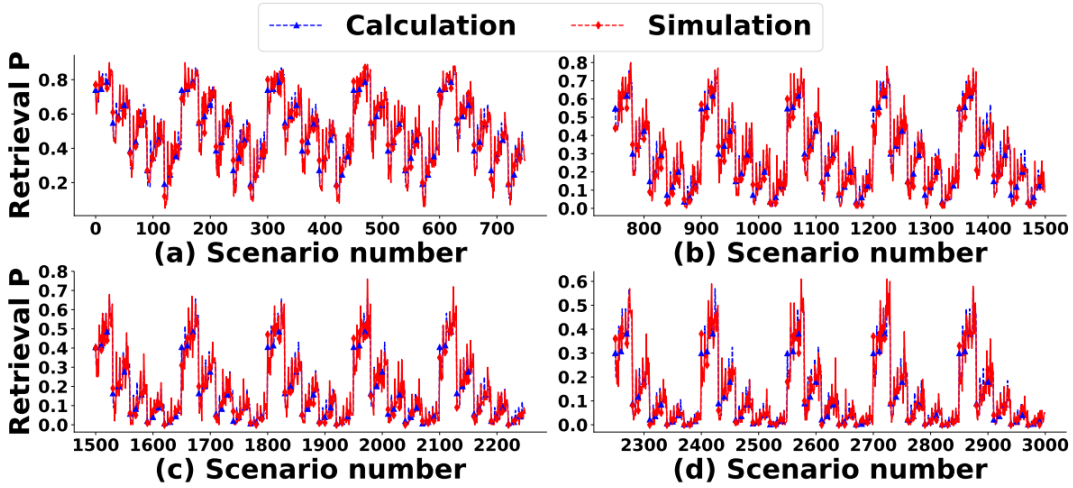


Figure 3.5: Comparing retrieval rates of the Trojan payload calculated using equation 3.3 and the simulation results considering the error rates (a) 0.0025 (b) 0.005 (c) 0.0075 and (d) 0.01 [37].

our calculations and simulations. The combination of a read error, a payload size, a fragment size and a retention number is considered as a scenario. In Figure 3.5, we can see the result clearly indicates that the proposed equations are good enough to estimate the success rates of such attacks with respect to variations in the error rates.

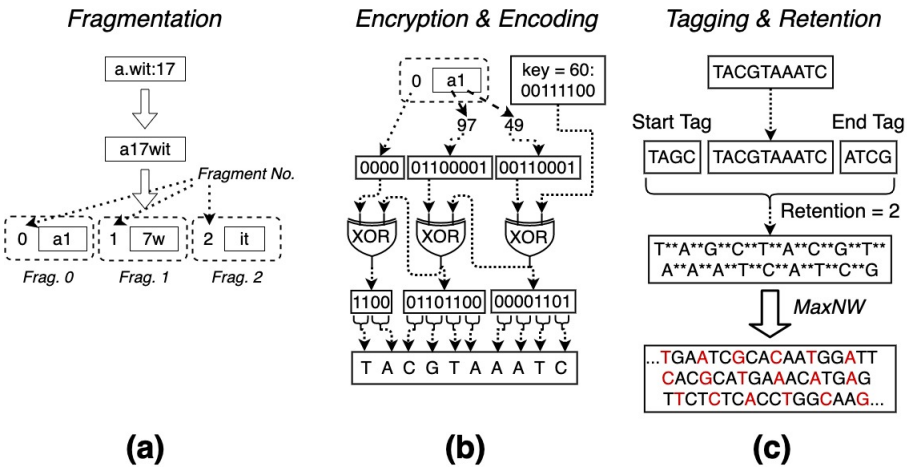


Figure 3.6: Trigger encoding into a DNA that includes (a) fragmentation (b) encryption and (c) steganography [36].

In our proposed technique in [37], the Needleman Wunsch algorithm is used to minimize the dissimilarity of the overall DNA in comparison to the host after encoding the payload (e.g., a plasmid DNA that is available naturally is encoded with the data). This process will increase the length of the overall

DNA after insertion compared to the original length. This spoils the idea of keeping the overall DNA sequence (containing the trigger encoded in it) as natural as possible. To counter this, we improved our approach [36] by applying the substitution based technique for injection. In this process, the size of the original DNA will not increase. Moreover, our approach will try to take the best possible substitutions to keep the overall DNA as natural as possible, considering the insertion with the best possible Needleman Wunsch score. The encryption and steganography techniques are also applied here to make the detection harder. The improved injection technique is described in Figure 3.6. Further details are available in Appendix C and D.

- **RQ3. Countermeasure for Bio-Hacking Security Attacks:** In the end-to-end scenario for a Trojan attack activated by a trigger sample and buffer overflow vulnerability exploits, the success of the attacks depends on the part of the DNA sequence with the malicious payload in the Trojan infected tools. Again, in a DNA sequencing process, the sequencer machine will produce a large number of reads, which will be arranged later in the stages of the sequencing pipeline. We put forward that if we can detect a read containing part of the malicious payload required for exploiting buffer overflow vulnerabilities or activation of the Trojan, then we can stop further processing in the downstream of the pipeline and protect it from attacks. We consider this problem as a classification problem where we need to classify a sequence, which will be either part of a longer sequence or from a read of the DNA sequencer, into clean parts of the sequence (no payload contained in the sequence) or malicious (have some part of the payload).

  First, we begin our work of countering the security attacks in the DNA sequencing pipeline with the detection of the payload and the classification approach in the case of buffer overflow vulnerability exploit. After analysing the recent works, where DNA or Protein sequence are used, we find the *Voss* Transformation [24, 105] is the most suitable for converting the sequences into signals. The transformation is similar to one hot encoding, i.e., 20 binary vectors of the length equal to the length of the DNA sequence will be there to present 20 protein bases of a protein sequence. For a protein base, in the relevant vector, a position will have value 1 if the sequence have that protein base in that particular position, otherwise it will be 0. Similarly, for the DNA signal the number of vectors should be 4 for 4 nucleotides. If the signal is *'GTAACTGCCAGA'*, then after *Voss* transformation it will be as follows:

For nucleotide A: [0,0,1,1,0,0,0,0,0,1,0,1]
For nucleotide T: [0,1,0,0,0,1,0,0,0,0,0,0]
For nucleotide G: [1,0,0,0,0,0,1,0,0,0,1,0]
For nucleotide C: [0,0,0,0,1,0,0,1,1,0,0,0]

Sometime the sequencer machine fails to determine a nucleotide base and then it is represented by *'N'* in the DNA sequence. In that case, we will have 5 vectors instead of 4. The *Voss* transformation is followed by applying signal processing techniques such as *Discrete Fourier Transformation (DFT)* to extract features. In [24], the technique of how a distance based algorithm can be used to classify healthy and cancerous sequences (e.g. mutations) using the features after applying the *Voss* transformation and *DFT* is investigated. We applied *CBR*, which is a classification framework in which we can also use distances between two sequences. The idea behind the concept is "similar problems have similar solutions".

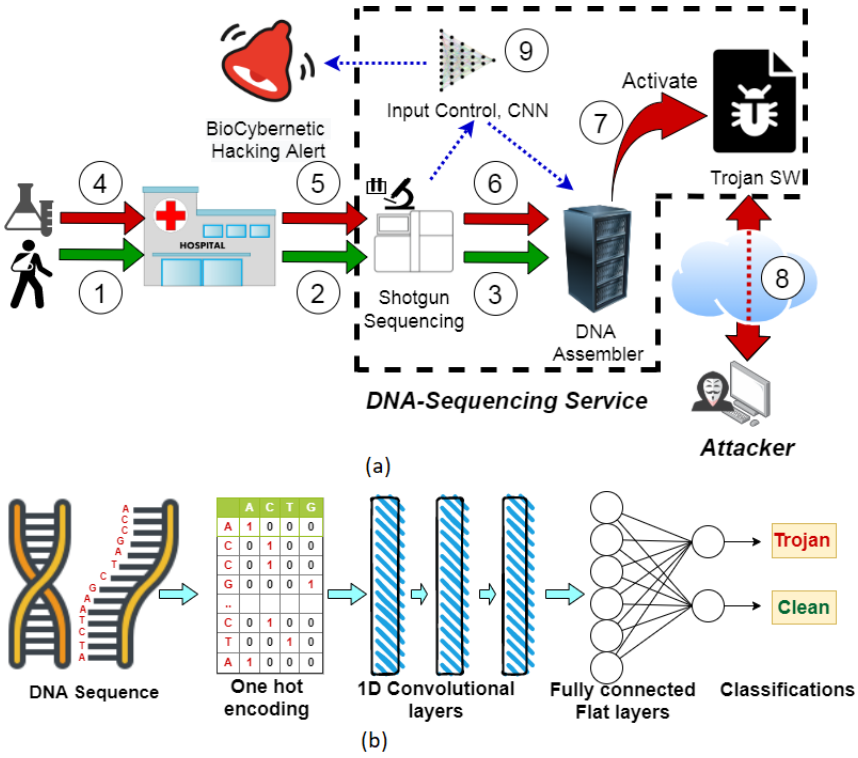


Figure 3.7: Countermeasure of Trojan attack: (a) Shows the overall scenario of the attack as well as the input control using CNN model and (b) architecture of the CNN model used for detection [36].

Our second consideration for countering the attack in the DNA sequencing pipeline is a software Trojan based end-to-end scenario, where it will be acti-

vated by a payload encoded in the DNA sequence for performing the malicious activities. In this scenario, the attack is more sophisticated as far as the detection is concerned, because the appearance of the DNA sequences with malicious payloads are more natural. While *CBR* is a simple and interpretable machine learning technique, our work on detecting the buffer overflow exploit has confirmed that the technique will not be good enough if the variations/mutations in the DNA sequences are kept to a minimum. Meanwhile, Deep learning has been proposed as a solution for classifications problems in applications using DNA and protein sequences in recent years. *Convolutional Neural Net (CNN)* is one of the deep learning techniques that works very well when the classification does not depends on the previous performance. Moreover, *CNN* helps in avoiding the implementation of the feature extraction related steps and we can leave this to the convolutional layers [95, 27]. Therefore, as a solution, we first proposed an input control technique using the CNN after the sequencer to protect further downstream of the DNA sequencing pipeline, as shown in Figure 3.7.

3.3 Validation

3.3.1 Detection of Encoded DNA for Buffer Exploit

To validate the effectiveness of the *CBR* algorithm as a detection technique, we have performed our experiment in [38] using real DNA sequences. DNA sequences from both eucaryotic and procaryotic cells are considered. For example, DNA sequences of E. coli plasmids as well as mammary, erythrocyte, and lymphocyte cells of humans are used in our experiments. For human cells, we used 254 mammary, 104 lymphocyte, and 48 erythrocyte DNA sequences, which are collected from publicly available data sets at the *National Center for Biotechnology Information (NCBI)* database.

We have implemented *CBR* from the scratch using Python programming language. Applying *CBR* instead of the technique used in [24] and using the same dataset, we have reproduced the result shown in Figure 3.8(a) to successfully create the baseline of our classification model. Our implemented CBR works based on the distances among features of the sequences. So, a new sequence will be classified with the classification of a existing sequence in the case library if the distance between their features is the lowest. To illustrate how CBR work based on the minimum distance, the ROC (Receiver Operating Characteristic) curves are also plotted for maximum and average distances alongside the minimum distance. Interestingly, we get 100 percent accuracy for all three kinds of distance. But unfortunately, one

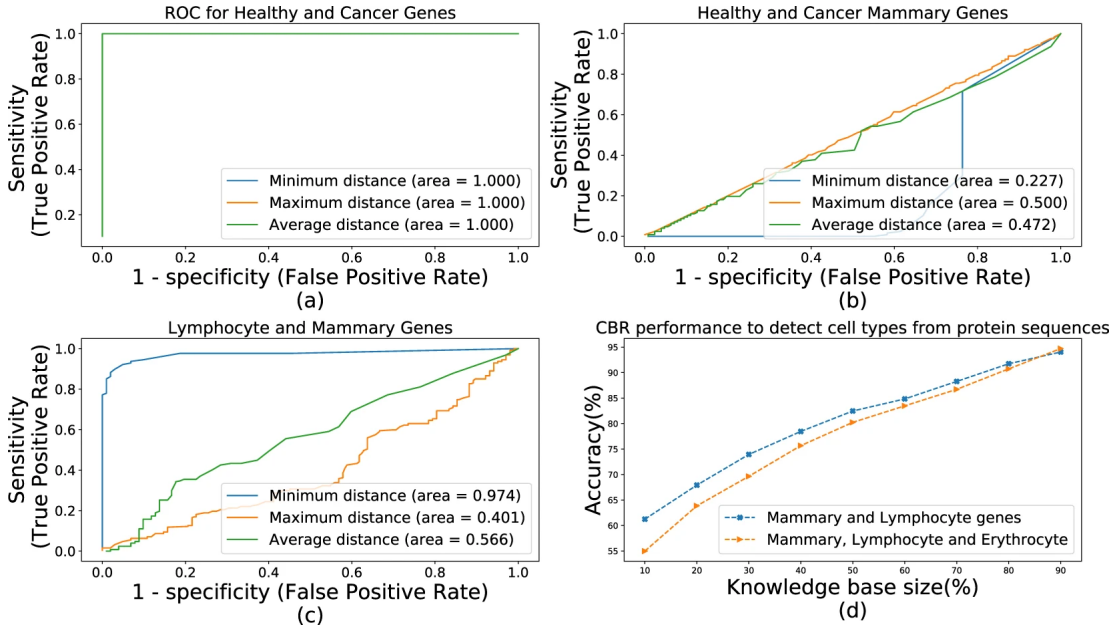


Figure 3.8: DNA similarity, extended study ROC curve for threshold-classification between (a) Healthy/Cancer samples from [24], (b) Healthy/Cancerous mammary and (c) lymphocyte/mammary samples from NCBI; and (d) the use of CBR cell-type classification to identify the payload containing the malicious code [38].

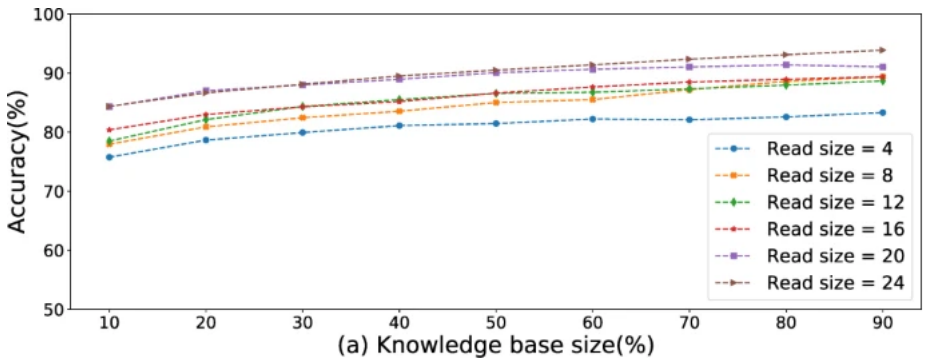


Figure 3.9: CBR-based detection of malicious content in DNA fragments of human mammary, erythrocyte, and lymphocyte DNAs [38].

weakness of the dataset was that it is not large enough. Furthermore, we have observed that the cancer data are clustered so tightly. Therefore, we considered a different dataset to examine the performance of the classification approach. When we have applied the same technique to a different but larger set of data, where we have classified healthy and cancerous data, the results are poor (Figure 3.8(b)). Interestingly, we get good results if we want to distinguish between protein sequences from various kinds of cell types (Figure 3.8(c) and (d)). The main reason is that it requires a small amount of mutations for a healthy sequence to turn cancerous, where the variations between the sequences will be quite high for the two different types. From that assumption, we expect that we will get promising results in detecting malicious payloads as the payload will introduce a larger number of mutations, hence it will be easy to separate them as the distance between natural and mutated sequences will be very high. For that experiment, first we encoded the shell commands (using different host and port addresses) into DNA sequences for exploiting the vulnerabilities and then insert those sequence into random places of the collected long DNA sequences. These sequences are the malicious sequences and the long DNA sequences without any insertion of the encoded shell commands are the clean sequence. To create an experimental dataset, we draw equal number of reads from the clean and malicious data considering a read size of the sequencing machine. In clean data any read will not contain any part of malicious DNA sequence and on the other hand in malicious data every read will contain some part of the encoded command. We create different experimental datasets considering different sequencing machine read sizes. After applying the simple *CBR* technique, we obtained decent results in detecting malicious payloads. The results of the detection accuracies for various read sizes and knowledge base sizes is shown in the Figure 3.9. Further details are available in Appendix B.

3.3.2 Detection of Trigger Encoded DNA for Trojan Attack

For the experimental performance evaluation of the *CNN* based detection technique to detect the DNA sequences with trigger message for the Trojan activation [36], real plasmid DNA sequences were used. The sequences were collected from a repository called *Addgene*. In total, 716 E. Coli plasmid DNA sequences were collected and used for our experiment. To prepare the experimental datasets, 4356 reads (with read size of 1000) were drawn from the collected sequences as clean samples. We randomly select 1000 reads out of the 4356 reads. As we did not make any changes inside the reads, so these are natural DNA. We called these natural DNA as clean sample for our experiments. Then we generate 1000 random hostnames and port addresses as trigger messages. Then these trigger messages were fragmented and

encoded into DNA snippets, and those fragments were put in different places of the clean samples. These are malicious samples for our experiments, which are modified, so unnatural. We repeat these processes for 10 times to create a dataset with 10,000 clean and 10,000 malicious sample. We also created malicious datasets applying encryption and steganography technique considering various encryption keys and retention numbers. A dataset is split into training and test dataset, where 75 percent data is used for training and rest of the 25 percent data is used for the test. We implemented a *CNN* model using Python programming language, and *Tensorflow* and *Keras* machine learning libraries.

After applying the *CNN*, we achieved promising results in detecting the DNA sequences encoded with the trigger payload to activate the Trojan. As described in Figure 3.7(b), *one hot encoding* [55, 104] technique is applied on the DNA sequences to make them suitable for the input of the *CNN* algorithm. We found decent results just by applying a single convolutional layer after optimising the hyper parameters. The performance of the model in terms of accuracy is shown in Figure 3.10 for payloads after applying various fragmentation, encryption (using various keys) and retention numbers. More details are available in Appendix D.

3.3.3 Validation using Wetlab Experiment

In our work [38], we have considered a buffer overflow vulnerability exploit shown in [64]. In [64], the authors only show how they have successfully synthesised a DNA that contains a payload for such an attack. In our case, we have considered an end-to-end evaluation scenario, where the perpetrator want to escape from any kinds of suspicion. We investigated a scenario where the DNA will be inside a bacterial plasmid and can be part of a nanonetwork of an *IoBNT*. These bacteria can be sprayed and spread over various things, such as in a kitchen or in a forensic scene. Let us assume that the food safety department may collect the bacteria from these places and passes it on to a third party to conduct the sequencing. During the process, the sequencing pipeline will be hacked by exploiting the buffer overflow vulnerabilities present in the sequencing pipeline. So, to validate this scenario, it is important to know the recovery rate of the synthesised DNA sequences from the bacteria. Therefore, we have collaborated with a wetlab experimentalist to conduct tests. First the DNA was synthesised and inserted into the plasmid and then into bacteria as shown in Figure 3.11.

Those bacteria were sprayed on various materials, such as lab coat, gloves, as well as work bench. The bacterial samples were then collected for DNA sequencing. The sequences were analysed to examine whether the synthesised sequences for the buffer overflow vulnerability exploit are retained or not. Figure 3.12 shows the recovery

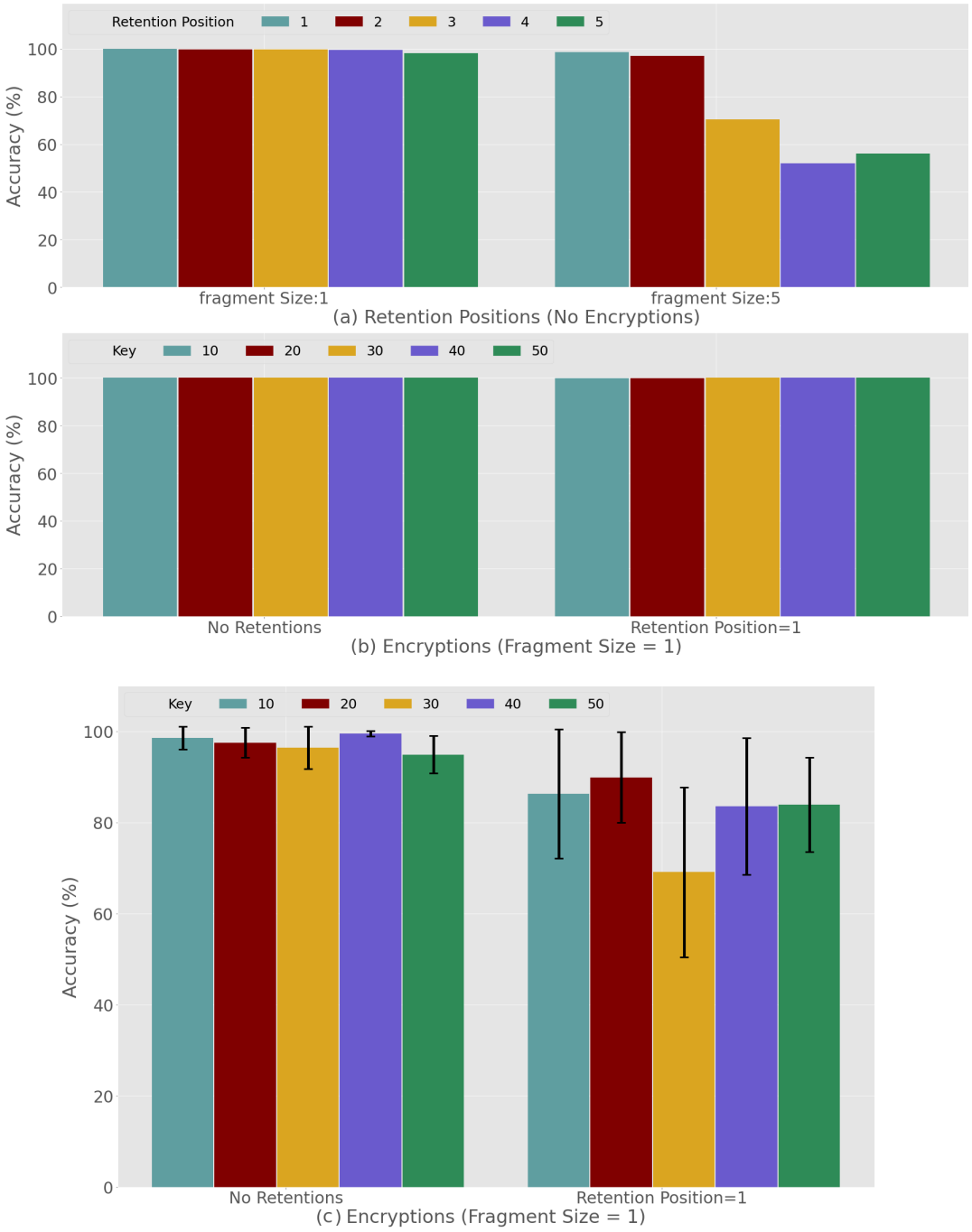


Figure 3.10: CNN detection results: (a) for various retention positions and fragmentation sizes but without encryption, (b) using encryption with prior knowledge of the encryption key, and (c) without prior knowledge of the encryption key [36].

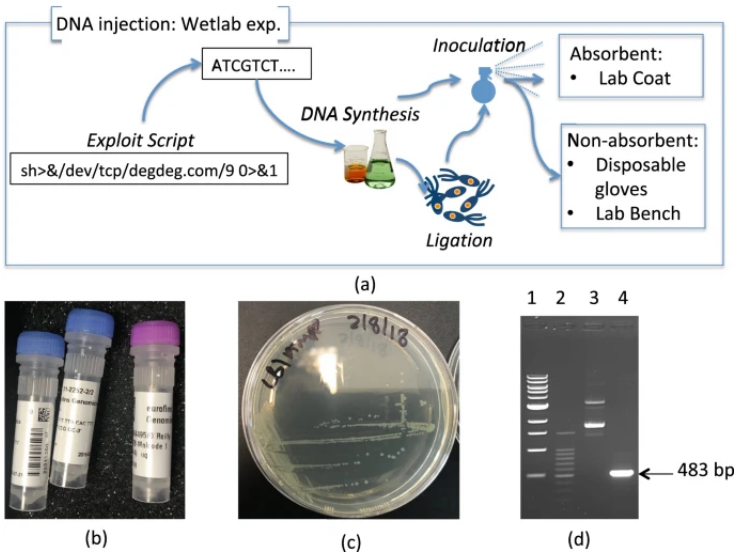


Figure 3.11: Image from wetlab experiment: (a) depicting overall experimental scenario, (b) code is encoded into DNA sequence and then inserted into plasmid, (c) recombinant of plasmid, and (d) agarose gel electrophoresis image verifying the presence of the plasmid [38].

results, which confirms the feasibility of such an attack scenario and indicates the preferred medium for a successful attack.

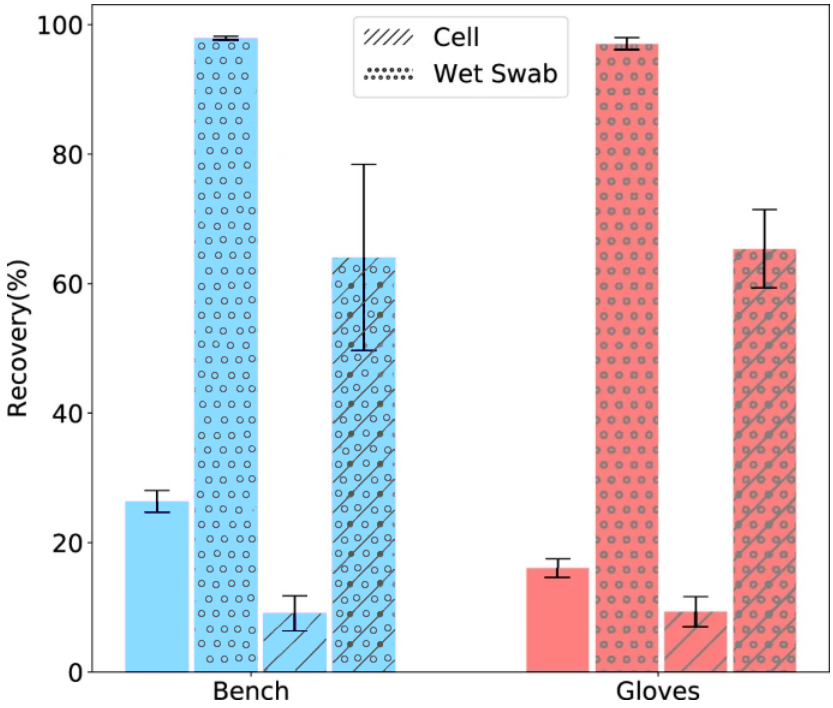


Figure 3.12: Recovery results from wetlab experiment: from two different surfaces, which are (a) bench and (b) gloves, while using dry and wet swabbing [38].

3.4 Contribution

From our above mentioned work, this is clear that our PhD thesis contributes towards the future *IoBNT*, especially more toward the future security challenges in addition to improving the throughput for data transmission in bio-nano communication. The achievements from the PhD thesis while addressing the research questions are listed below.

- **RQ1. Multi-bit Data Transfer using Bacterial Nanonetworks:**
  - **A1.** Recent researches have revealed the opportunities in nano communication by using various bacterial traits, e.g., bacterial motility. In our PhD research, we have shown how we can utilise three properties, which are motility, conjugation, and bio-luminescent to send more than one bit of data at a time. In our work [92], we have shown this novel technique of data communication, where we have outperformed the performance of state-of-the-art techniques. Our approach is based on creating multiple receivers that are arranged spatially, where each receiver represents a number of different bits. This avoids the need for complex engineering bacteria with plasmids that encode the data. The transmission of multiple bacteria to spatial locations that represents different bits has shown significant improvement over super slow bacterial nanonetwork transmission that have been previously proposed.
  - **A2.** Another contribution of the work is that we can send more than one bit of data while avoiding complex processes such as encoding and decoding the message into a DNA sequence.
- **RQ2. Bio-Hacking Security attacks based on DNA Encoded Data:**
  - **A3.** We have conducted the validation of an end to end scenario of buffer overflow vulnerability exploit in the DNA sequencing pipeline [38].
  - **A4.** We have shown how a Trojan attack is possible in a DNA sequencing pipeline. The Trojan attack has existed as a cyber security challenge for a number of years. However, demonstrating how it is a possible cyber-attack in a DNA sequencing pipeline is novel. Instead of hardware Trojan, we have considered a special case, where they will remain dormant and will only be activated with a specially designed DNA used as a trigger [36].
  - **A5.** Our described Trojan attack scenario in the DNA sequencing pipeline offers perpetrators the advantages of using cryptography to hide their

traces as it gives the opportunity to split the trigger message into fragments using techniques like encryption and steganography to make detection harder. Though aggregating the fragments might not be a difficult process, but the final decoding process will be harder if the encryption and steganography are applied. The success rates of this attack considering the read errors in the sequencers have also been examined (Figure 3.5). Furthermore, we consider a technique that is used in bio-informatics to make the overall DNA sequence more natural so that the detection becomes harder. An additional improvement is by constructing the trigger sample for the Trojan activation using the substitution based injection to make the overall DNA sequence more natural.

  - **A6.** We have collaborated with wetlab experimentalists to conduct a lab experiments to synthesis a DNA (with trigger sample in it to activate the Trojan) in order to examine the viability of designing the DNA. This validates the possibility of such an attack and the ability to synthesise the DNA.

- **RQ3. Countermeasure for Bio-Hacking Security Attacks:**

  - **A7.** We developed a countermeasure for an existing buffer overflow vulnerability exploit in DNA sequencing pipeline. Our technique uses real DNA sequences to discover the malicious payload used to exploit the vulnerability. Our results confirm the effectiveness of our proposed method, where we were able to detect 95% malicious DNA. Moreover, rather than using a black box type of machine learning technique, we proposed a simple and easily interpretable technique (as like the explainable AI technique) called *CBR* for the a countermeasures.

  - **A8.** We have demonstrated successfully how the payload for exploiting buffer overflow vulnerability can be placed inside a plasmid, and this is inserted into bacteria through wetlab experiments by means of collaborating with synthetic biologists. The bacteria were sprayed on various things to examine the recovery rate to validate the viability of our scenario.

  - **A9.** For the Trojan attack, we have proposed designing a DNA to encode the trigger message for the activation process. The overall DNA will be more natural in appearance and decoding of the trigger message will be harder as cryptography techniques are applied. As the end-to-end attack scenario is so complex and detection will be harder, therefore we have proposed a countermeasure technique using *CNN* as in the recent past has showed promising results in solving classification problems. The

architecture to describe the end-to-end scenario and where the detection solution should be presented are also described in our work.

Table 3.2 shows the relation between challenges, research questions, achievements, and the corresponding publications for this thesis.

Table 3.2: Research Achievement (with respect to challenges)

Research Q.	**Challenge**	**Achievement**	**Presented in Paper**	**Appx.**
RQ1	C1	A1, A2	[92]	Appx. A
RQ3	C4	A3, A7, A8	[38]	Appx. B
RQ2	C2, C3	A4, A5, A6	[37]	Appx. C
RQ3	C4	A9	[36]	Appx. D

Similarly, Table 3.3 shows the relation between limitations, research questions, achievements, and the corresponding publications for this thesis.

Table 3.3: Research Achievement (with respect to limitations)

Research Q.	**Limitation**	**Achievement**	**Presented in Paper**	**Appx.**
RQ1	L1	A1, A2	[92]	Appx. A
RQ3	L4	A3, A7, A8	[38]	Appx. B
RQ2	L2, L3	A4, A5, A6	[37]	Appx. C
RQ3	L4	A9	[36]	Appx. D

Chapter 4

Conclusion and Future Work

In this chapter we will conclude our PhD thesis and also discuss about the possible future works.

4.1 Conclusion

IoBNT is an idea of connecting body area network for getting data from nano and micro level along with performing actions in such level to bring revolutionary improvements in health care, smart farming, environment control etc., where bacteria will play an important role as they have the potential of being used as bio-compatible nano devices in such scenarios. The auxiliary plasmid DNA, bacterial traits like conjugation and motility have already showed a lot of promise to send large amount of data from a transmitter to receiver by encoding them into the DNA. To avoid the complex processes of DNA Synthesis and DNA sequencing required for reading and writing operations, few alternative techniques were proposed in the past, where the bits are sent based on the bacterial traits like motility and their collecting behaviours such as bioluminescent and quorum sensing. As bacterial motility is a slow and stochastic process, this kind of data transmission is extremely slow (bits per second).

4.1.1 Multi-bits Data Transfer

In our research [92], we have shown a novel way of sending two bits (which can be extended to multiple bits) at a time based on the bioluminescent and quorum sensing traits of bacteria. It is surely a tremendous improvement over the state-of-the-art ON-OFF key-based approach as by using our purposed technique, significantly higher transfer rate can be achieved. In the ON-OFF key-based approach, the presence of the bit depended only on the population of the bacteria in the receiver, and as a result the performance in terms of bit transfer was very poor.

4.1.2 End to End Evaluation of the Attack

From a state-of-the art research [64] it has been realised that other types of data transmission technique using bacteria, where the data will be encoded into DNA, can come under security threats as it is possible to exploit buffer overflow vulnerability in DNA sequencing pipelines. This PhD research considers this as a future impediment towards the progress and applications of *IoBNT*. To avoid being guilty of considering non-functional system requirements like security as a very last task to do, it is necessary to investigate the possibility of such security attacks and address them early. Furthermore, the impact of such work will be very high, as many applications will benefit from it in the near future. Therefore, first, we have come up with an end to end evaluation scenario considering state-of-the-art buffer overflow vulnerability exploitation research [64] and then we have also collaborated with wet lab researchers to validate the feasibility of such scenario. Our considered end to end evaluation experimental scenario demonstrated how an attacker can be disguised and no suspicion will be raised as the designed DNA can be placed in the plasmids of bacteria. Moreover, these bacteria are also sprayed on different materials considering scenarios where third parties can collect the bacterial sample from there to send them to another organisation for the full sequencing. Our experiment [38] has proven the possibility of constructing malicious DNA and how it is possible for the hackers to hide their identity while performing a successful attack.

4.1.3 Novel Trojan Attack Scenario

We further investigate the possibility of other kinds of attacks in similar scenarios and successfully demonstrated a Trojan type of attack scenario in the DNA sequencing pipelines. In the buffer overflow attack, a large portion of the DNA needs to be muted to insert the payload required in the existing natural DNA. Therefore, from a detection point of view, the resulted unnatural DNA sequence offers an advantage as we can assume that with the help of commonly available bioinformatics tools or by applying a simple AI/ML technique, we can separate the unnatural DNA sequences (potential threats) from the natural DNA sequences. That can prompt further processing with serious caution to protect the downstream of the sequencing pipeline from being compromised. That is why the Trojan attack scenario is chosen and designed to explore the possibility of attack in DNA sequencing pipelines, where the DNA will be kept as natural as possible to make the detection harder. To prove such possibility, we consider a trigger based Trojan scenario, where the Trojan software is already implanted in the DNA sequencing pipeline and doing some legitimate activities. It will remain dormant in the pipeline unless it is activated by a specially designed trigger message in the form of a DNA sequence snippet. Another

advantage of considering such scenario is that, we can apply fragmentation and encryption on the trigger message while designing and synthesizing the DNA message required for the exploit. The small fragments of tiny trigger messages can be placed at various places of a large and naturally available DNA sequence to minimise the dissimilarity with the original one. The message is only reassembled and decrypted by the Trojan. The advantage of such technique is that it needs small number of mutations and we can reduce dissimilarity significantly by carefully choosing the location of mutation on the original DNA sequence resulting in reduced the suspicion. Furthermore, the application of encryption and steganography will make the retrieval of the actual message harder for a countermeasure mechanism even after successfully reassembling the fragments, which makes it very difficult to identify the perpetrator. Our research [37] and [36] have successfully demonstrated the idea of the above-mentioned Trojan attack scenario. Furthermore, to validate how realistic it is to synthesis such DNA sequences, we have also performed wetlab experiments [36]. Our research showed that this kind of attack is very possible and realistic. The end to end evaluation was also conducted again for this Trojan type of attack scenario.

4.1.4 Countermeasure to the Attacks

Finally, we propose solutions for these two types of attack. For two end-to-end scenarios, we have considered different detection techniques. As we described earlier that for the buffer overflow exploit, it requires big portion of the existing DNA sequence to be mutated. In the process, the final DNA (for sequencing) becomes quite artificial compared to any existing natural DNA. So, we prefer to go with a simple explainable AI technique to detect clean and malicious DNA. *CBR* is chosen as an explainable AI technique in our research [38]. We considered the DNA sequences as the genomic signals and constructed a case base using the features after applying *FFT*. Our work [38] proves that in the case of buffer overflow exploit, a detection system using a simple technique such as *CBR* is good enough to detect malicious DNA sequences. The performance depends on the size of the case base,i.e. larger case base gives better accuracy. However in the case of Trojan attack, the sequence will be quite similar to its natural form after the necessary minimum mutation, So, a deep learning solution is proposed to detect malicious DNA in such complex Trojan attack scenario. Our work [36] confirms that a deep learning based model can achieve excellent performance in detecting the DNA sequence with a payload of trigger messages. However, accuracy might be reduced if encryption and stenography are applied. Overall, we have achieved very high accuracy in detecting DNA sequences designed and synthesised for both the buffer overflow vulnerability

exploit and the trigger-based Trojan attack.

4.2 Future Work

As *IoBNT* is a very new research frontier and technology is in its infancy, there is a great deal of scope for improvement. Improving data transmission throughput and ensuring security will play a big role in the future of this research area.

4.2.1 Data Transmission Improvement

A future work on our proposed idea of sending multiple bits of data using bacterial quorum sensing and bioluminescence can be performed in a wetlab experiment to validate the idea. We have successfully shown the significant improvement in the bit transfer rate using our proposed novel technique, where the message encoding in DNA is avoided, by our *in silico* experiments. However, it remains to be seen whether the slicing of the sequence required for the production of GFP is possible in various ways. Furthermore, it is also unknown whether the bacteria can reconstruct the complete DNA sequence after conjugation. To validate whether such proposed technique is realistic or not in the future, we need to perform wetlab experiments where we can try to slice the required DNA sequences and insert them in bacteria. Afterwards, we can reconstruct them with the help of conjugation and then finally observe the presence of light due to GFP production. New experiments using different bioluminescent colours can be considered in addition to the existing available colours to improve the data transfer rate in comparison with our proposed technique.

4.2.2 Exploring Other Attacks

The attacks on DNA sequencing pipeline will not be limited to only two types of attacks that are considered for our work. Therefore, all other existing cyber attacks should be considered very seriously to ensure future defences of *IoBNT*. For example, script injection is a common attack available in today's cyber world. Scripting languages like Python and Javascript (NodeJs) are popular for building web applications and we can anticipate that many tools in the DNA sequence pipeline will be developed using these scripting languages. Therefore, any tools in DNA sequencing pipeline built using these scripting languages should be considered for potential vulnerability of script injection attacks. We might argue that input validation might be used to protect web based applications from submitting script as a value of an input fields. But similar to the attack scenarios considered in this thesis, if the script is encoded into DNA and submitted as an input, then it can bypass the input validation.

In the later stage, it can be decoded and can take advantages of the vulnerabilities in the backend of the system developed in Python or JavaScript to perform an script injection attack. So, investigating the possibility of such script injection attacks and also perform an end to evaluation of the attack are very important. Proposing a countermeasure for such attack can also be considered as an important future work. We can consider SQL injection attack as a possibility of another kind of attack. DNA sequence will be stored in various databases and at the same time a DNA itself can be used as a storage device. SQL injection is a common attack where a SQL command can be manipulated to perform malicious activities, e.g., by passing the authentication, and running a block of SQL code. Therefore, The possibility of such attack in DNA sequencing pipeline should also be examined.

Besides these two attacks (Script and SQL injection), there might be other security challenges related to DNA sequencing pipeline and DNA-based storage, which can be considered as future works. We need to analyse and perform end to end evaluations in other possible attack scenarios. The detection of such attacks is also important. If we can come up with solutions for individual attacks, then it will be a new challenge, how we can model a detection system to detect multiple types of attacks in the DNA sequencing pipeline. So, a work on the fusion of various vulnerability detection, i.e. machine learning model fusions can be interesting to explore in future research.

4.2.3 Identifying the Perpetrators

Our research has confirmed that it is very easy to hide identity while performing an attack as the perpetrators will not order a random DNA sequence for synthesise using a web application, rather they may use a third party to collect and send bacterial sample instead. Furthermore, we have successfully demonstrated how to counter such attacks based on the detection technique using machine learning. But the question of tracing the perpetrators is still unanswered. So, proposing a protocol for collecting samples and sequencing to avoid such situations of escaping from being traced can be an interesting future work. Researchers must come up with a protocol to trace the perpetrators while also considering privacy issues.

4.2.4 Possible Detection Improvements

With the continued progress on the process of DNA Synthesis, DNA sequencing and the improvements in the applications using DNA, the use of artificial DNA sequences will be more common, e.g., people will start to store data encoded in DNA. Therefore, a lot of new variation of DNA due to many genetic engineering processes can be created. In our work, we just detect malicious DNA using a binary

classifier, i.e. in a scenario where the DNA sequence will be with sequence snippet for malicious activity or completely clean DNA sequence. One issue with these kinds of binary classifier is that it can only separate malicious and clean payloads. But it will be a difficult to differentiate the clean DNA sequences and unsafe DNA sequences if the DNA sequences have unnatural portion in them. So, we need to consider scenarios, where the detection system can detect the DNA sequences of three types, which are completely clean DNA, with artificial portion with security threats and DNA with artificial portion but no security threat. Finally, another possible future work can be the standardization of security practices with updates and revisions considering possibility of attacks described in our works and mentioned future works.

Chapter 5

List of Research Article

The list of research articles which are published as part of this research work and also included in the thesis is given below. The published research articles are listed in the order of published dates.

Published Articles:

- P1. D. Unluturk, M. S. Islam, S. Balasubramaniam, and S. Ivanov, "Towards Concurrent Data Transmission: Exploiting Plasmid Diversity by Bacterial Conjugation," IEEE Transactions on NanoBioscience, vol. 16, no. 4. Institute of Electrical and Electronics Engineers (IEEE), pp. 287–298, Jun-2017.

- P2. M. S. Islam, S. Ivanov, E. Robson, T. Dooley-Cullinane, L. Coffey, K. Doolin, and S. Balasubramaniam, "Genetic similarity of biological samples to counter bio-hacking of DNA-sequencing functionality," Scientific Reports, 9(1), June 2019.

- P3. M. S. Islam, S. Ivanov, K. Doolin, L. Coffey, T. M. Dooley-Cullinane, D. Berry, and S. Balasubramaniam, "Trojan Bio-Hacking of DNA-Sequencing Pipeline," Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication. ACM, 25-Sep-2019.

- P4. M. S. Islam, S. Ivanov, H. Awan, J. Drohan, S. Balasubramaniam, L. Coffey, S. Kidambi, and W. Sri-saan, "Using Deep Learning to Detect Digitally Encoded DNA Trigger for Trojan Malware in Bio-Cyber Attacks." Scientific Reports, 12(1), June 2022.

Bibliography

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1):39–59, March 1994.

[2] Mobyen Uddin Ahmed and Peter Funk. A Computer Aided System for Post-operative Pain Treatment Combining Knowledge Discovery and Case-Based Reasoning. pages 3–16. Springer Berlin Heidelberg, 2012.

[3] Prema T. Akkasaligar and Sumangala Biradar. Selective medical image encryption using DNA cryptography. *Information Security Journal: A Global Perspective*, 29(2):91–101, February 2020.

[4] I. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy. The internet of bio-nano things. *IEEE Communications Magazine*, 53(3):32–40, March 2015.

[5] Ian Akyildiz and Josep Jornet. The internet of nano-things. *IEEE Wireless Communications*, 17(6):58–63, December 2010.

[6] D. Anastassiou. Genomic signal processing. *IEEE Signal Processing Magazine*, 18(4):8–20, 2001.

[7] Burton W Andrews and Pablo A Iglesias. An information-theoretic characterization of the optimal gradient sensing response of cells. *PLoS computational biology*, 3(8):e153, August 2007.

[8] Sasitharan Balasubramaniam, Sigal Ben-Yehuda, Sophie Pautot, Aldo Jesorka, Pietro Lio', and Yevgeni Koucheryavy. A review of experimental opportunities for molecular communication. *Nano Communication Networks*, 4(2):43–52, June 2013.

[9] Sasitharan Balasubramaniam, Nikita Lyamin, Denis Kleyko, Mikael Skurnik, Alexey Vinel, and Yevgeni Koucheryavy. Exploiting bacterial properties for multi-hop nanonetworks. *IEEE Communications Magazine*, 52(7):184–191, July 2014.

[10] Shaibal Barua, Shahina Begum, and Mobyen Uddin Ahmed. Supervised machine learning algorithms to diagnose stress for vehicle drivers based on physiological sensor signals. *Studies in health technology and informatics*, 211:241–8, 2015.

[11] Jacob Beal, Ting Lu, and Ron Weiss. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PloS one*, 6(8):e22490, jan 2011.

[12] Shahina Begum, Shaibal Barua, Reno Filla, and Mobyen Uddin Ahmed. Classification of physiological signals for wheel loader operators using Multi-scale Entropy analysis and case-based reasoning. *Expert Systems with Applications*, 41(2):295–305, 2014.

[13] Gaymon Bennett, Nils Gilman, Anthony Stavrianakis, and Paul Rabinow. From synthetic biology to biohacking: are we prepared? *Nature Biotechnology*, 27(12):1109–1111, December 2009.

[14] Daniel Berman, Anna Buczak, Jeffrey Chavis, and Cherita Corbett. A survey of deep learning methods for cyber security. *Information*, 10(4):122, April 2019.

[15] Cornelia Caragea, Adrian Silvescu, and Prasenjit Mitra. Protein sequence classification using feature hashing. *Proteome Science*, 10(Suppl 1):S14, 2012.

[16] Jianjing Cui, Jun Long, Erxue Min, Qiang Liu, and Qian Li. Comparative study of CNN and RNN for deep learning based intrusion detection system. In *Cloud Computing and Security*, pages 159–170. Springer International Publishing, 2018.

[17] Nina F de Groot, Britta C van Beers, and Gerben Meynen. Commercial DNA tests and police investigations: a broad bioethical perspective. *Journal of Medical Ethics*, 47(12):788–795, September 2021.

[18] Yiming Dong, Fajia Sun, Zhi Ping, Qi Ouyang, and Long Qian. DNA storage: research landscape and future prospects. *National Science Review*, 7(6):1092–1107, January 2020.

[19] Zhihua Du, Xiangdong Xiao, and Vladimir N. Uversky. Classification of chromosomal DNA sequences using hybrid deep learning architectures. *Current Bioinformatics*, 15(10):1130–1136, February 2021.

[20] Brian D. Earp, Joanna Demaree-Cotton, Michael Dunn, Vilius Dranseika, Jim A. C. Everett, Adam Feltz, Gail Geller, Ivar R. Hannikainen, Lynn A. Jansen,

Joshua Knobe, Julia Kolak, Stephen Latham, Adam Lerner, Joshua May, Mark Mercurio, Emilian Mihailov, David Rodríguez-Arias, Blanca Rodríguez López, Julian Savulescu, Mark Sheehan, Nina Strohminger, Jeremy Sugarman, Kathryn Tabb, and Kevin Tobia. Experimental philosophical bioethics. *AJOB Empirical Bioethics*, 11(1):30–33, January 2020.

[21] Shaker El-Sappagh, Mohammed Elmogy, and A.M. Riad. A fuzzy-ontology-oriented case-based reasoning framework for semantic diabetes diagnosis. *Artificial Intelligence in Medicine*, 65(3):179–208, 2015.

[22] K. Virgil English, Islam Obaidat, and Meera Sridhar. Exploiting memory corruption vulnerabilities in connman for IoT devices. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, jun 2019.

[23] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, January 2019.

[24] Antara Ghosh and Soma Barman. Application of euclidean distance measurement and principal component analysis for gene identification. *Gene*, 583(2):112–120, 2016.

[25] Luis Uriel Gonzalez-Avila, Juan Manuel Vega-López, Leda Ivonne Pelcastre-Rodríguez, Omar Alejandro Cabrero-Martínez, Cecilia Hernández-Cortez, and Graciela Castro-Escarpulli. The challenge of CRISPR-cas toward bioethics. *Frontiers in Microbiology*, 12, May 2021.

[26] Thomas E. Gorochowski, Antoni Matyjaszkiewicz, Thomas Todd, Neeraj Oak, Kira Kowalska, Stephen Reid, Krasimira T. Tsaneva-Atanasova, Nigel J. Savery, Claire S. Grierson, and Mario di Bernardo. BSim: An agent-based tool for modeling bacterial populations in systems and synthetic biology. *PLoS ONE*, 7(8):e42790, August 2012.

[27] Hemalatha Gunasekaran, K. Ramalakshmi, A. Rex Macedo Arokiaraj, S. Deepa Kanmani, Chandran Venkatesan, and C. Suresh Gnana Dhas. Analysis of DNA sequence classification using CNN and hybrid models. *Computational and Mathematical Methods in Medicine*, 2021:1–12, July 2021.

[28] Shize Guo, Jian Wang, Zhe Chen, Yubai Li, and Zhonghai Lu. Securing IoT space via hardware trojan detection. *IEEE Internet of Things Journal*, 7(11):11115–11122, nov 2020.

[29] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, nov 2020.

[30] Antoine L. Harfouche and Farid Nakhle. Creating bioethics distance learning through virtual reality. *Trends in Biotechnology*, 38(11):1187–1192, November 2020.

[31] Monowar Hasan, Ekram Hossain, Sasitharan Balasubramaniam, and Yevgeni Koucheryavy. Social Behavior in Bacterial Nanonetworks: Challenges and Opportunities. November 2014.

[32] Joseph M. Hatfield. Social engineering in cybersecurity: The evolution of a concept. *Computers & Security*, 73:102–113, March 2018.

[33] Siquan Hu, Ruixiong Ma, and Haiou Wang. An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences. *PLOS ONE*, 14(11):e0225317, November 2019.

[34] Xin Hu, Bo Xia, Martin Skitmore, and Qing Chen. The application of case-based reasoning in construction management research: An overview. *Automation in Construction*, 2016.

[35] Fahad Hussain, Umair Saeed, Ghulam Muhammad, Noman Islam, and Ghazala Shafi Sheikh. Classifying cancer patients based on DNA sequences using machine learning. *Journal of Medical Imaging and Health Informatics*, 9(3):436–443, March 2019.

[36] M. S. Islam, S. Ivanov, H. Awan, J. Drohan, S. Balasubramaniam, L. Coffey, S. Kidambi, and W. Sri-saan. Using deep learning to detect digitally encoded DNA trigger for trojan malware in bio-cyber attacks. *Scientific Reports*, 12(1), June 2022.

[37] M. S. Islam, S. Ivanov, K. Doolin, L. Coffey, T. M. Dooley-Cullinane, D. Berry, and S. Balasubramaniam. Trojan bio-hacking of DNA-sequencing pipeline. In *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication*. ACM, September 2019.

[38] Mohd Siblee Islam, Stepan Ivanov, Eric Robson, Tríona Dooley-Cullinane, Lee Coffey, Kevin Doolin, and Sasitharan Balasubramaniam. Genetic similarity of biological samples to counter bio-hacking of DNA-sequencing functionality. *Scientific Reports*, 9(1), June 2019.

[39] Miten Jain, Hugh E. Olsen, Benedict Paten, and Mark Akeson. The oxford nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biology*, 17(1), November 2016.

[40] Sara B. Jordan, Samantha L. Fenn, and Benjamin B. Shannon. Transparency as threat at the intersection of artificial intelligence and cyberbiosecurity. *Computer*, 53(10):59–68, October 2020.

[41] Curtis Steward Jr., Luay A. Wahsheh, Aftab Ahmad, Jonathan M. Graham, Cheryl V. Hinds, Aurelia T. Williams, and Sandra J. DeLoatch. Software security: The dangerous afterthought. In *2012 Ninth International Conference on Information Technology - New Generations*. IEEE, April 2012.

[42] Shruti Kalsi, Harleen Kaur, and Victor Chang. DNA cryptography and deep learning using genetic algorithm with NW algorithm for key generation. *Journal of Medical Systems*, 42(1), December 2017.

[43] Firoz Khan, Cornelius Ncube, Lakshmana Kumar Ramasamy, Seifedine Kadry, and Yunyoung Nam. A digital DNA sequencing engine for ransomware detection using machine learning. *IEEE Access*, 8:119710–119719, 2020.

[44] Jeffrey C. Kimmel, Andrew D. Mcdole, Mahmoud Abdelsalam, Maanak Gupta, and Ravi Sandhu. Recurrent neural networks based online behavioural malware detection techniques for cloud infrastructure. *IEEE Access*, 9:68066–68080, 2021.

[45] Hideki Kobayashi, Mads Kaern, Michihiro Araki, Kristy Chung, Timothy S Gardner, Charles R Cantor, and James J Collins. Programmable cells: interfacing natural and engineered gene networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(22):8414–9, jun 2004.

[46] Vasileios Kouliaridis, Georgios Kambourakis, and Tao Peng. Feature importance in android malware detection. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, dec 2020.

[47] Bhuvana Krishnaswamy, Caitlin M. Austin, J. Patrick Bardill, Daniel Russakow, Gregory L. Holst, Brian K. Hammer, Craig R. Forest, and Raghupathy Sivakumar. Time-elapse communication: Bacterial communication on a microfluidic chip. *IEEE Transactions on Communications*, 61(12):5139–5151, December 2013.

[48] Jean-Baptiste Lamy, Boomadevi Sekar, Gilles Guezennec, Jacques Bouaud, and Brigitte Séroussi. Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. *Artificial Intelligence in Medicine*, 94:42–53, March 2019.

[49] T. Laver, J. Harrison, P.A. O'Neill, K. Moore, A. Farbos, K. Paszkiewicz, and D.J. Studholme. Assessing the performance of the oxford nanopore technologies MinION. *Biomolecular Detection and Quantification*, 3:1–8, March 2015.

[50] David B. Leake and David C. Wilson. *When Experience is Wrong: Examining CBR for Changing Tasks and Environments*, pages 218–232. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[51] Liangzhi Li, Kaoru Ota, and Mianxiong Dong. Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Transactions on Industrial Informatics*, 14(10):4665–4673, October 2018.

[52] Shancang Li, Li Da Xu, and Shanshan Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, April 2014.

[53] Zihao Liu, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. Deepn-jpeg: A deep neural network favorable jpeg-based image compression framework. In *Proceedings of the 55th Annual Design Automation Conference*. ACM, June 2018.

[54] Ali Louati, Sabeur Elkosantini, Saber Darmoul, and Lamjed Ben Said. A Case-Based Reasoning System to Control Traffic at Signalized Intersections. *IFAC-PapersOnLine*, 49(5):149–154, 2016.

[55] Zhibin Lv, Hui Ding, Lei Wang, and Quan Zou. A convolutional neural network using dinucleotide one-hot encoder for identifying dna n6-methyladenine sites in the rice genome. *Neurocomputing*, 422:214–221, 2021.

[56] Yangdi Lyu and Prabhat Mishra. Automated test generation for trojan detection using delay-based side channel analysis. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, mar 2020.

[57] Abhishek Majumdar, Arpita Biswas, Atanu Majumder, Sandeep Kumar Sood, and Krishna Lal Baishnab. A novel DNA-inspired encryption strategy for concealing cloud storage. *Frontiers of Computer Science*, 15(3), December 2020.

[58] Sivappriya Manivannan, Lakshmi Kuppusamy, and N. Sarat Chandra Babu. TRAP-GATE: A probabilistic approach to enhance hardware trojan detection and its game theoretic analysis. *Journal of Electronic Testing*, 36(5):607–616, oct 2020.

[59] Jennifer L. Mantle, Jayan Rammohan, Eugenia F. Romantseva, Joel T. Welch, Leah R. Kauffman, Jim McCarthy, John Schiel, Jeffrey C. Baker, Elizabeth A. Strychalski, Kelley C. Rogers, and Kelvin H. Lee. Cyberbiosecurity for biopharmaceutical products. *Frontiers in Bioengineering and Biotechnology*, 7, May 2019.

[60] Andrew McDole, Mahmoud Abdelsalam, Maanak Gupta, and Sudip Mittal. Analyzing CNN based behavioural malware detection techniques on cloud IaaS. In *Lecture Notes in Computer Science*, pages 64–79. Springer International Publishing, 2020.

[61] M. Nathaniel Mead. Nutrigenomics: The genome–food interface. *Environmental Health Perspectives*, 115(12), December 2007.

[62] Mirjam Minor, Mohd. Siblee Islam, and Pol Schumacher. Confidence in Workflow Adaptation. pages 255–268. Springer Berlin Heidelberg, 2012.

[63] Randall S. Murch, William K. So, Wallace G. Buchholz, Sanjay Raman, and Jean Peccoud. Cyberbiosecurity: An emerging new discipline to help safeguard the bioeconomy. *Frontiers in Bioengineering and Biotechnology*, 6, April 2018.

[64] Peter Ney, Karl Koscher, Lee Organick, Luis Ceze, and Tadayoshi Kohno. Computer security, privacy, and DNA sequencing: Compromising computers with synthesized DNA, privacy leaks, and more. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 765–779, Vancouver, BC, August 2017. USENIX Association.

[65] H.H. Nguyen, J. Park, S. Park, C.-S. Lee, S. Hwang, Y.-B. Shin, T. Ha, and M. Kim. Long-term stability and integrity of plasmid-based dna data storage. *Polymers*, 10(28):1–10, 2018.

[66] Yutaka Okaie, Tadashi Nakano, Takahiro Hara, Senior Member, Takuya Obuchi, Kazufumi Hosoda, Yasushi Hiraoka, and Shojiro Nishio. Cooperative Target Tracking by a Mobile Bionanosensor Network. *IEEE transactions on nanobioscience*, 13(3):267–277, September 2014.

[67] Anne O. Oyewole, Lucy Barrass, Emily G. Robertson, James Woltmann, Hannah O'Keefe, Harsimran Sarpal, Kim Dangova, Catherine Richmond, and

Dawn Craig. COVID-19 impact on diagnostic innovations: Emerging trends and implications. *Diagnostics*, 11(2):182, January 2021.

[68] Fatih Ozsolak and Patrice M. Milos. RNA sequencing: advances, challenges and opportunities. *Nature Reviews Genetics*, 12(2):87–98, December 2010.

[69] Malathi P., M. Manoaj, R. Manoj, V. Raghavan, and R.E. Vinodhini. Highly Improved DNA Based Steganography. *Procedia Computer Science*, 115:651–659, 2017.

[70] Zhixin Pan and Prabhat Mishra. Automated test generation for hardware trojan detection using reinforcement learning. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*. ACM, jan 2021.

[71] Byung-Wook Park, Jiang Zhuang, Oncay Yasa, and Metin Sitti. Multifunctional bacteria-driven microswimmers for targeted active drug delivery. *ACS Nano*, 11(9):8910–8923, September 2017.

[72] Gabriela Pavarini, Robyn McMillan, Abigail Robinson, and Ilina Singh. Design bioethics: A theoretical framework and argument for innovation in bioethics research. *The American Journal of Bioethics*, 21(6):37–50, January 2021.

[73] Cristiana Pavlidis, George P. Patrinos, and Theodora Katsila. Nutrigenomics: A controversy. *Applied & Translational Genomics*, 4:50–53, March 2015.

[74] Michael Pedersen and Andrew Phillips. Towards programming languages for genetic engineering of living cells. *Journal of the Royal Society, Interface / the Royal Society*, 6 Suppl 4:S437–50, aug 2009.

[75] Xiao-Ou Ping, Yi-Ju Tseng, Yan-Po Lin, Hsiang-Ju Chiu, Feipei Lai, Ja-Der Liang, Guan-Tarn Huang, and Pei-Ming Yang. A multiple measurements case-based reasoning method for predicting recurrent status of liver cancer patients. *Computers in Industry*, 69:12–21, 2015.

[76] Joshua R Porter, Burton W Andrews, and Pablo A Iglesias. A framework for designing and analyzing binary decision-making strategies in cellular systems. *Integrative biology : quantitative biosciences from nano to macro*, 4(3):310–7, March 2012.

[77] Rami Puzis, Dor Farbiash, Oleg Brodt, Yuval Elovici, and Dov Greenbaum. Increased cyber-biosecurity for DNA synthesis. *Nature Biotechnology*, 38(12):1379–1381, November 2020.

[78] Dima Rabadi and Sin G. Teo. Advanced windows methods on malware detection and classification. In *Annual Computer Security Applications Conference*. ACM, dec 2020.

[79] Yoel Raban and Aharon Hauptman. Foresight of cyber security threat drivers and affecting technologies. *foresight*, 20(4):353–363, August 2018.

[80] Heena Rathore, Abdulla Khalid Al-Ali, Amr Mohamed, Xiaojiang Du, and Mohsen Guizani. A novel deep learning strategy for classifying different attack patterns for deep brain implants. *IEEE Access*, 7:24154–24164, 2019.

[81] Jiadong Ren, Zhangqi Zheng, Qian Liu, Zhiyao Wei, and Huaizhi Yan. A buffer overflow prediction approach based on software metrics and machine learning. *Security and Communication Networks*, 2019:1–13, mar 2019.

[82] Love Kumar Sah, Sheikh Ariful Islam, and Srinivas Katkoori. Variable record table: A run-time solution for mitigating buffer overflow attack. In *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, aug 2019.

[83] Kevin Santoso, Suk-Hwan Lee, Won-Joo Hwang, and Ki-Ryong Kwon. Sector-based DNA information hiding method. *Security and Communication Networks*, 9(17):4210–4226, nov 2016.

[84] Richa Sharma, Vijaypal Singh Rathor, G.K. Sharma, and Manisha Pattanaik. A new hardware trojan detection technique using deep convolutional neural network. *Integration*, 79:1–11, jul 2021.

[85] Seth L. Shipman, Jeff Nivala, Jeffrey D. Macklis, and George M. Church. CRISPR–cas encoding of a digital movie into the genomes of a population of living bacteria. *Nature*, 547(7663):345–349, July 2017.

[86] Ankit Singh, Aditi Sharma, Nikhil Sharma, Ila Kaushik, and Bharat Bhushan. Taxonomy of attacks on web based applications. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. IEEE, jul 2019.

[87] Prasanthi Sreekumari. Malware detection techniques based on deep learning. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, may 2020.

[88] Sarvesh Kumar Srivastava and Vikramaditya G. Yadav. Bionic manufacturing: Towards cyborg cells and sentient microbots. *Trends in Biotechnology*, 36(5):483–487, May 2018.

[89] Abdulkadir Tasdelen and Baha Sen. A hybrid CNN-LSTM model for pre-miRNA classification. *Scientific Reports*, 11(1), July 2021.

[90] Vrizlynn L. L. Thing. IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, March 2017.

[91] Elena Toader, Oana Eva, Andrei Olteanu, and Sorin Anton. Application of biomedical technologies - issues in modern bioethics. In *2017 E-Health and Bioengineering Conference (EHB)*. IEEE, June 2017.

[92] Bige D. Unluturk, M. Siblee Islam, Sasitharan Balasubramaniam, and Stepan Ivanov. Towards concurrent data transmission: Exploiting plasmid diversity by bacterial conjugation. *IEEE Transactions on NanoBioscience*, 16(4):287–298, 2017.

[93] Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. Argumentation and explainable artificial intelligence: a survey. *The Knowledge Engineering Review*, 36, 2021.

[94] Abdul Wahab, Hilal Tayara, Zhenyu Xuan, and Kil To Chong. Dna sequences performs as natural language processing by exploiting deep learning algorithm for the identification of n4-methylcytosine. *Scientific Reports*, 11(1), 2021.

[95] Daniel Weimer, Bernd Scholz-Reiter, and Moshe Shpitalni. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals*, 65(1):417–420, 2016.

[96] Ray A. Wickenheiser. A crosswalk from medical bioethics to forensic bioethics. *Forensic Science International: Synergy*, 1:35–44, 2019.

[97] Luhang Xu, Weixi Jia, Wei Dong, and Yongjun Li. Automatic exploit generation for buffer overflow vulnerabilities. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, jul 2018.

[98] Kim Sangwoo Yang In Seok. Analysis of whole transcriptome sequencing data: Workflow and software. *Genomics Inform*, 13(4):119–125, 2015.

[99] Rozhin Yasaei, Shih-Yuan Yu, and Mohammad Abdullah Al Faruque. GNN4tj: Graph neural networks for hardware trojan detection at register transfer level. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, feb 2021.

[100] Koichiro Yasaka and Osamu Abe. Deep learning and artificial intelligence in radiology: Current applications and future directions. *PLOS Medicine*, 15(11):e1002707, November 2018.

[101] Ali K. Yetisen. Biohacking. *Trends in Biotechnology*, 36(8):744–747, August 2018.

[102] B.Pharm. Yolanda Smith. Amino acids and protein sequences, 2021.

[103] Da Zhang and Mansur R. Kabuka. Protein family classification from scratch: A CNN based deep learning approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(5):1996–2007, September 2021.

[104] Yongqing Zhang, Shaojie Qiao, Shengjie Ji, and Yizhou Li. DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding. *International Journal of Machine Learning and Cybernetics*, 11(4):841–851, July 2019.

[105] Jian Zhao, Jiasong Wang, and Hongmei Jiang. Detecting periodicities in eukaryotic genomes by ramanujan fourier transform. *Journal of Computational Biology*, 25(9):963–975, 2018.

Appendices

Appendix A

Distributed Modulation using Bacterial Nanonetworks

Journal Title	IEEE Transactions on NanoBioscience
Article Type	Regular
Complete Author List	Bige D. Unluturk, Mohd Siblee Islam, Stepan Ivanov and Sasitharan Balasubramaniam
Status	Date of Publication: 23 May 2017
Contribution	Mr. Islam is the second author of the above mentioned article. Mr. Islam has equal contribution to generate and formulate the idea behind the paper. His main contribution is setting up the simulation to simulate the above mentioned scenario and estimate the propagation delay of sending bacteria from the receiver to transmitter for various parameters e.g. distance between transmitter and the receiver, volume of the transmitter and receiver, viscosity and thermal condition of the liquid environment, velocity dragging and swimming of the bacteria, model of the tumble angle of the bacteria etc. The contribution is toward our research question 1. We can take and improve the work to make communication possible using bio-compatible device like bacteria inside body area network for *IoBNT*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

Towards Concurrent Data Transmission: Exploiting Plasmid Diversity by Bacterial Conjugation

Bige D. Unluturk, M. Siblee Islam, Sasitharan Balasubramaniam, and Stepan Ivanov

***Abstract*—The progress of molecular communication is tightly connected to the progress of nanomachine design. State-of-the-art states that nanomachines can be built either from novel nanomaterials by the help of nanotechnology or they can be built from living cells which are modified to function as intended by synthetic biology. With the growing need of biomedical applications of MC, we focus on developing bio-compatible communication systems by engineering the cells to become MC nanomachines. Since this approach relies on modifying cellular functions, the improvements in the performance can only be achieved by integrating new biological properties. A previously proposed model for molecular communication is using bacteria as information carriers between transmitters and receivers, also known as *bacterial nanonetworks*. This approach has suggested encoding information into the plasmids inserted into the bacteria which leads to extra overhead for the receivers to decode and analyze the plasmids to obtain the encoded information. Another scheme, which is proposed in this paper, is to determine the digital information transmitted based on the quantity of bacteria emitted. While this scheme has its simplicity, the major drawback is the low data rate resulting from the long propagation of the bacteria. To improve the performance, this paper proposes a *Distributed modulation* scheme utilizing three bacterial properties, namely, engineering of plasmids, conjugation, and bacterial motility. In particular, genetic engineering allows us to engineer different combinations of genes representing different series of bits. When compared to *Binary Density modulation* and the *M-ary Density modulation*, it is shown that the Distributed modulation scheme outperforms the other two approaches in terms of bit error probability as well as the achievable rate for varying quantity of bacteria transmitted, distances, as well as time slot length.**

***Index Terms*—Molecular communication, bacterial nanonetworks, bit error probability, achievable rate**

Bige D. Unluturk is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA.

M. Siblee Islam and Stepan Ivanov is with the Telecommunication Software and Systems Group, Waterford Institute of Technology, Ireland.

S. Balasubramaniam is with the Nano Communications Center (NCC), Department of Electronics and Communication Engineering at Tampere University of Technology (TUT), Finland.

E-mail: bigedeniz@ece.gatech.edu, sislam@tssg.org, sasi.bala@tut.fi, sivanov@tssg.org.

This work is supported by the Academy of Finland FiDiPro programme for the project "Nanocommunications Networks" 2012 - 2016, and the Finnish Academy Research Fellow programme under Project no. 284531. It is also partly funded by Science Foundation Ireland via the CONNECT research centre (grant no. 13/RC/2077), which is co-funded by the European Regional Development Fund (ERDF).

I. Introduction

THE field of *Molecular Communication* [1], [2] aims to create nanoscale networks whose nodes are communicating by the exchange of molecules to accomplish sensing and actuating tasks in macroscale where electro-magnetic communication fails to operate properly. Environments where EM waves have trouble properly propagating, cause detrimental effects to the environment or cannot be feasibly maintained form the niche of MC applications such as infrastructure monitoring in air ducts where EM waves irrecoverably attenuate [3] or intra-body applications where EM waves cause health hazards [11].

In this study, we focus on biomedical applications of MC where biological components and systems are utilized to create artificial communication systems. This new paradigm for developing communication networks could pave the way for new forms of healthcare monitoring solutions, where artificial communication systems are developed from biological components and are integrated with the human body [4], [8], [9], [11]. This could lead to an in-body network system that provides fine granular sensing and early detection of diseases.

Numerous models for molecular communication have been proposed, including diffusion based systems where molecules that represent information are diffused into the environment [12], as well as FRET [13], and calcium signaling [14]. Besides these models, another approach is utilizing organisms as information carrier, and specifically bacteria. Bacteria have a number of properties that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

have been used to create molecular communication. In [22], the quorum sensing process was utilized to transfer information. In quorum sensing, the bacteria coordinate and signal each other by producing the molecules known as AHL (acyl homoserine lactone) according to their local population density. A bacteria population will produce AHL molecules that diffuse and travel through the microfluidic channels to a receiver which is another population of bacteria. Through the quorum sensing process, the receiver can sense the density of the population transmitting the signal.

In [20], programmed bacteria that emit attractants and repellants are used to localize and track targets. Through their cooperative communication process, the bacteria can search the environment in a timely and efficient manner. This form of searching process can provide new solutions towards localization of diseased cells that are malignant. Another technique that has been proposed is utilizing the motility properties of bacteria and their ability to hold plasmids, which can potentially be used to encode and store information [16]. The motility is usually achieved through the flagella that extend from the bacteria body, enabling the bacteria to swim in a fluidic medium. Based on these properties, the bacteria nanonetwork is established by having the bacteria pick up plasmid with encoded information from a transmitter nanomachine, and swimming towards a receiver to unload the plasmid [18]. However, an issue with this form of information delivery is the process required to encode the information into the plasmids, and engineering the receiver to decode this information by first removing the plasmid from the bacteria, and searching through the DNA to find the genes that hold the encoded information. It requires a mechanism that can read the DNA which is not an easy task. A simpler approach is to use bioluminescence to decode the information by modulating the quantity of bacteria rather than the genes carried by the bacteria. The intensity of the bioluminescence indicates the modulation of the bacteria. Furthermore, information transfer in bacterial nanonetworks create long delays. To mitigate these delays we create parallel transmission mechanisms by introducing distributed receivers which are spatially separated. Using engineering plasmids, information carrying bioluminescence genes are distributed among bacteria groups to create **plasmid diversity**. We use multiple transmitter-receiver pairs, each bound to a different combination of the distributed genes, which are distinguished by the spatial separation, i.e., the receiver of each pair illuminate at a different location which leads us to create distinguishable parallel paths. Hence, the information transfer rate can be improved by sending information simultaneously from these parallel paths, i.e., distributed receivers.

The proposed modulation technique is achieved through different combinations of genes carried by the bacteria that can lead to bioluminescence. In this paper, we focus on four different combinations of the genes on the plasmids, leading to M distributed receivers. According to the information that are to be transmitted the corresponding bacteria will be released from the transmitter. Bacteria will swim towards one of M receivers to bind and conjugate with the non-motile bacteria that are stationary. Upon successful binding, the genes that are transferred and combined in the receiver bacteria will enable bioluminescence. We refer to this form of modulation as *Distributed modulation for bacterial nanonetworks* (For the rest of the article we will only refer to *Distributed modulation*).

In this paper, we first simulated the bacteria propagation behavior in 3D to determine the probability distribution for the first passage time of bacteria which is modeled as an Inverse Gaussian Function. Then, we introduce *Binary Density Modulation*, *M-ary Density Modulation*, and *Distributed Modulation* schemes. We compare these schemes by evaluating the performance metrics such as the bit error probability as well as the achievable rate, where we vary the distances between the transmitter and receivers, as well as the average transmit power which corresponds to the quantity of bacteria released from the transmitter. The results from our analysis show that the Distributed Modulation scheme outperforms the other two schemes due to the minimization of ISI that can result from bacteria emitted during previous time slots. This in turn leads to higher achievable rates. The results also found that the achievable rate changes with the time slot length, since distinct bacteria for different symbols can be concurrently emitted from the transmitter, leading to smaller time slots required for each symbol transmission.

The contributions of this paper can be listed as

- We determined of the first hitting time parameters of Brownian Motion by simulations

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

conducted with BSim based on the physical parameters of system.
- We introduced the plasmid diversity and distributed receivers concepts to create diversity in bacterial nanonetworks.
- We proposed three modulation schemes and derived the corresponding probability of errors and achievable rates where distributed modulation outperforms the others and stands out as a reliable candidate for modulation.

The paper is organized as follows: Section III introduces the system model for bacterial nanonetworks by presenting the background information on the genes programmed into the plasmid leading to bioluminescence. In Section II, an extensive literature review is given. In Section III-B the propagation model of the bacteria is presented. In Section III-C the bioluminescence occurring upon the reception of bacteria at the receiver is described. Section IV presents the detailed model of the modulation schemes, while Section V presents the performance evaluation comparison between the three different schemes. Lastly, Section VI concludes the paper.

II. Related Work

Bacterial nanonetworks are studied in the literature from many different perspectives. In [5], the fundamentals of bacterial networks are discussed. The encoding and decoding of information on bacterial plasmid by conjugation are defined in communications engineering perspective. Furthermore, the motion of bacteria carrying plasmid messages inside various environments is defined as the propagation of the information. In [6], a simulation model is developed to study the channel capacity in bacterial nanonetworks. In both of these studies, bacteria is considered to move following run-and-tumble cycles as in our work, however, the motion is not analytically modeled. [5] only simulates the propagation channel but does not calculate any other communication metric whereas [6] does not consider the loss due to random motion of bacteria but incorporates it as a term in delay.

Another perspective to bacterial nanonetworks is presented in [10] where a simulation is performed to characterize the dynamics of bacterial nanonetworks. The BNSim is tool developed which takes into account chemotactic movement of bacteria, genetic circuits and intercellular interactions among bacteria for drug delivery applications. In [9], a mathematical model for capturing the dynamics of bacteria populations are derived for biological applications. In [8], a statistical physics model is proposed to study the dynamics of dense networks of bacteria coupled with intercellular communication of bacteria. These studies focus mostly on swarming of the bacteria and how the bacteria population is distributed into the environment and whether they accumulate on the target.

Furthermore, in [43] a non-equilibrium statistical physics inspired model is proposed to study biological communication defined in many levels such as inside cell, intercellular, and interkingdom levels. [43] proposes new metrics for information theory where there is no definition of individual transmitter or receiver but each cell performs both functions. The mutual information is here defined between the concentration of an intracellular entity such as quorum sensing molecules and the physical behavior of bacteria such as bioluminescence.

Another perspective considered in [44], presents the information spreading with opportunistic communications in bacterial nanonetoworks using an epidemic approach similar to Delay Tolerant Networks and model analytically the number of bacteria receiving the plasmid carrying the information in a complex bacterial nanonetwork.

Despite all the previous efforts in the literature, there are still many problems in determining how to use the bacteria and their swarming capabilities in order to create efficient biological communication networks. Prior body of work concentrates on modelling and simulating the organization of bacterial populations and their motion with respect to environmental cues.

Our approach in this paper combines different elements from state-of-the-art to move one step closer to realizing bacterial nanonetworks. Our work analyzes the performance of communication systems that can be build on top of these elements in terms of the achievable rate. Furthermore, we devise the novel concept of plasmid diversity and distributed receivers which improves the information transfer rate.

III. System Model

Although previous works have proposed various modulation schemes for molecular communications,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

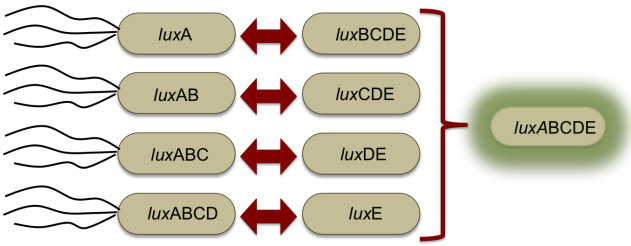


Fig. 1. Illustration of genes *luxA*, *luxB*, *luxC*, *luxD*, and *luxE* distributed between the bacteria. The collection of all five genes will lead to bioluminescence.

the majority of these works were focused on diffusion based systems [41], [24], [25]. The objective of our proposed approach is to develop a modulation scheme improving the data rate using bacterial properties. We utilize in total three different properties, namely, engineering plasmids, motility, as well as the conjugation process.

Engineering plasmids: Besides the chromosome, bacteria also have a circular DNA molecule that is known as plasmid. In Synthetic Biology, plasmids are usually engineered with different combination of genes that provides the bacteria new traits. One of these traits is engineering bioluminescence in the bacteria to emit visible light. We assume that the transmitter and receiver bacteria acquired a combination of *lux* genes by the engineering of plasmids before being deployed in the environment. Currently, it is very common and easy to modify the genetic material of bacteria using techniques like CRISPR [34]. Furthermore, the genes that we chose for this study, namely, *lux* genes which encodes bioluminescence proteins are very thoroughly studied in the literature and it is well-known how to create plasmids comprising of *lux* genes [35] since bioluminescence is frequently used as a reporting mechanism of the genomic level events [36].Bacterial cells produce light if they have all of the following five genes, namely, *luxA*, *luxB*, *luxC*, *luxD*, and *luxE* [28]. In the event that any of these five genes are missing, no light will be produced. However, the bacteria may be able to pick any of these genes from plasmids of other organism in order to have the full collection that will lead to light emission. This is illustrated in Figure 1. Bioluminescence is very common in marine bacteria such as *Vibrio fishieri*, but the gene sequence responsible for luminescence can easily be transferred to other bacteria such as *E. coli* which is the bacteria considered in this study.

Motility: Bacteria are able to mobilize by utilizing flagella, which are hair like structures that extend from the body. In order to achieve motility, the flagella will form a single body that will act as a propeller to enable the bacteria to mobilize between different locations [19].

Conjugation: Bacteria are able to transfer and pass plasmids between each other. This process is known as conjugation. During conjugation, the bacteria will come together and form a physical connection through the **pilus** that allows copies of plasmids to be transferred [15]. Bacterial conjugation is a natural DNA transfer mechanism for bacteria [37], [38] which creates significantly dynamic genomes where lots of genes can be deleted or inserted easily. When two bacteria come close to each other they make a physical connection by joining their pili. Then, the plasmid of the donor bacteria gets nicked and a single strand DNA is transferred to the recipient cell. Both cells synthesize complementary DNA strands and both plasmids become circular again. Conjugation may happen between both the same species of bacteria or different species however the plasmid transfer rate is higher between similar strains. The plasmid transfer rate changes between 10^{-6} to 10^{-3} [39], [40].

These three different properties allow us to create molecular communication links for bacterial nanonetworks, which are illustrated in Figure 2. The information is coded through the genes *luxA*, *luxB*, *luxC*, *luxD*, and *luxE* that are inserted into the plasmid of the motile bacteria contained in the transmitter. The receiver consists of non-motile bacteria (i.e, the flagella have been removed) which is located at a distance d apart. We considered that all pairs are parallel to each other as shown in Figure 2, so that no pair has advantage over the others. However, in a more elaborate situation where the a priori probabilities of each symbol are known, the transmitter-receiver distance of the pair transferring the most probable symbol may be smaller to increase the rate.

We assume that the time is slotted and the transmitter releases N_0 genetically encoded bacteria at the beginning of each time slot which lasts T_s sec. Furthermore, we assume that the transmitters and receivers are perfectly synchronized. The synchronization may be established using quorum sensing which activates certain intracellular mechanisms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

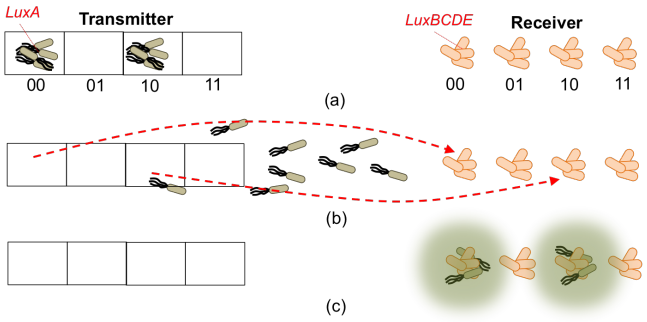


Fig. 2. Illustration of modulation using bacterial nanonetworks with distributed receivers. The transmitter contains motile bacteria, while the receiver contains non-motile population. In this example, the digital bits "0010" is to be sent from the transmitter to the receiver. Each population of bacteria at the transmitter and receiver have combination of genes that will lead to bioluminescence (e.g. for digital bits "00", the transmitter bacteria contain *luxA*, while the receiver non-motile bacteria contain *luxBCDE*). (a) the motile bacteria are initially stored within the transmitter, (b) the bacteria are released from the transmitter, (c) the conjugation process at the receiver between the motile and non-motile bacteria.

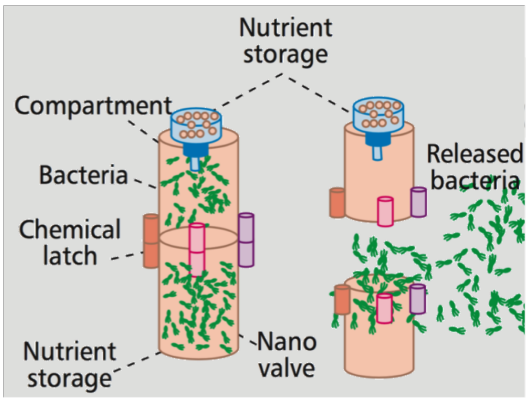


Fig. 3. The transmitter model [45]

only when the bacteria population reaches a threshold [7], or using the extracellular noise common to all cells which induces collective dynamics [31], or a blind synchronization algorithm which implements the non-decision directed Maximum Likelihood (ML) criterion for the estimation of channel delay [32].

A. Transmission Model

The transmitter contains compartments storing the bacteria with different gene combinations representing two digital bits, as illustrated in Figure 2. The transmitter can be modeled as a container with chemical latches opening and closing to release bacteria as shown in Figure 3. The opening and closing process of the latch can be stimulated chemically. In order to have a reusable transmitter, a nutrient harvesting process can be mounted into the nutrient storage [45]. In Figure 2, the bits "00" consist of the transmitter motile bacteria plasmids having *luxA*, while the non-motile bacteria are the receiver having *luxBCDE* genes in their plasmid. Therefore, the combination of genes for each pair of bits between the transmitter and receiver is unique. At the beginning of each time slot, the transmitter releases these bacteria into the medium where they propagate randomly.

Since the bacteria follow a Brownian motion, bacteria will be dispersed in the environment and only a portion of the released bacteria will be able to reach the receiver. In the following section we will introduce a propagation model for the motile transmitter bacteria.

B. Propagation Model

Bacteria can follow either a random or directed motion. When there is no specific source of attraction, bacteria move in the environment randomly following a Brownian motion model. When there is a source of attraction such as nutrition, light or magnetic field, bacteria move towards it following a chemotactic movement model [29]. In this study, we assume that there is no specific source of attraction in the environment.

The Brownian motion of bacteria is governed by a sequence of run-and-tumble process. This means that they run straight at constant speed v for a random time duration t_r, then tumbles without changing position for t_t, and choose a new direction with a random angle θ, and this is followed by the run phase. Repeating this sequence, the bacteria move randomly in the environment. To characterize this movement, we ran 3D simulations in a confined environment to obtain the properties of the first passage time of bacteria released from the transmitter reaching the receiver. In particular, we performed 3D discrete time simulation of bacteria using run-and-tumble model for Δt time intervals.

The simulation is conducted using *BSim* [21] simulator, an agent-based computational tool to model the dynamics of bacterial population moving in a 3D environment. For the simulation, a 3D container of $1\ mm^3$ size is considered, where the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

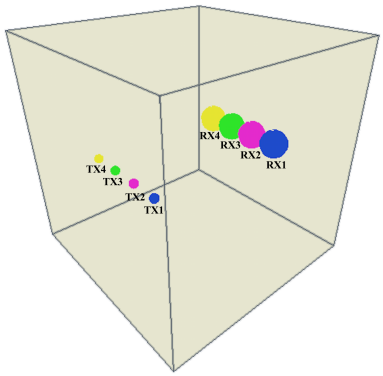


Fig. 4. Initial state of simulation environment.

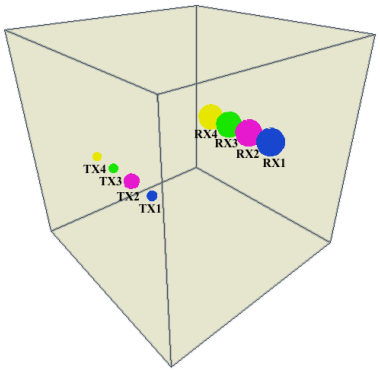


Fig. 5. Bacteria are released from the sender (green coloured).

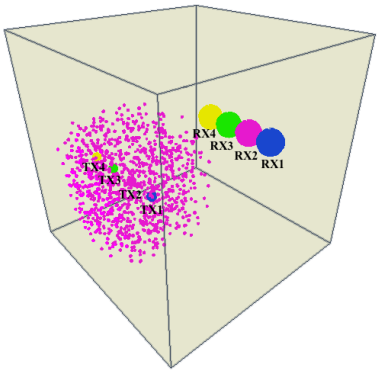


Fig. 6. Bacteria are moving away from the sender.

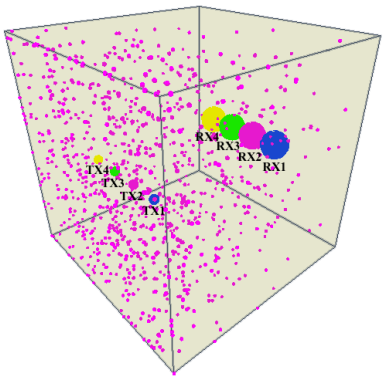


Fig. 7. Some of the bacteria reached to different receivers.

surfaces of the container are solid. As the size of the container is quite big considering the size of the bacteria and the velocity of the bacterial movement, the collisions and reflections between the bacteria and container walls can be ignored for our total simulation time. The receivers and transmitters are set at opposite sides from the center of the container at equal distances. Four transmitters and four corresponding receivers are placed for various distances. Both the transmitters and receivers are circular in shape with a very small radius. The bacteria are considered to successfully reach the target once it collides with that receiver. In order to determine the impact of successfully reaching the target, the radius of the receivers are varied for various runs. We assume that the population of bacteria moving inside the container will remain the same during the transmission period due to sufficient supply of nutrients [22]. A sample of the transmitter and receivers locations, their initial state, the bacteria release and movement, as well as their propagation towards the receivers are illustrated in Figure 4, 5, 6 and 7 respectively. For our simulation, we have considered three different distances between the transmitters and corresponding receivers and 3 different receiver volumes. There are two states of bacterial movement, which are running when the flagella are rotating counter clockwise and tumbling when the flagella are rotating clockwise. The tumbling angles are random values and follow a gamma distribution. The maximum tumbling angle is set as 180 degree. Other parameters for the simulation is listed into Table I.

Since it is known that the first passage time of the random walk is represented by an inverse probability distribution function [23], we compared our simulation results with an inverse Gaussian pdf. The distribution obtained from the simulation for the first passage time of a bacterium at the receiver is very similar to an inverse Gaussian distribution, as shown in Figure 8, which is expressed as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

TABLE I
SIMULATION PARAMETERS

Parameter Name	Value
Temperature	305 K
Viscosity	2.7e-3 Pa s
Radius of the bacteria	1 μm
Flagella force	1 pN
Mean time to end a run	0.86 sec
Mean time of end a tumble	0.14 sec
The maximum tumbling angle	180 degree
Boltzman constant	1.38e-23
Number of Bacteria	10000
Max. Bacteria lifetime	6 hours
Distances	500, 1000, 1500 μm
Receiver radius	100, 200, 300 μm
Simulation duration	6 hours
Timestamp	0.01 seconds

TABLE II
FITTED INVERSE GAUSSIAN PARAMETERS

Distance (μm)	RX Volume (μm^3)	ν	λ
500	100	2993.37	1044.080
500	200	2971.459	1092.047
500	300	3033.642	1113.672
1000	100	5880.463	4594.112
1000	200	5726.056	4651.549
1000	300	5742.775	2532.088
1500	100	8083.802	11887.945
1500	200	8017.303	11568.980
1500	300	8126.618	11760.664

$$f(t) = \left[\frac{\lambda}{2\pi t^3}\right]^{1/2} \exp\left(-\frac{\lambda(t-\nu)^2}{2\mu^2 t}\right), \quad (1)$$

where the coefficients λ and ν depend on the run-and-tumble parameters of the bacteria, the distance between transmitter and receiver, and the receiver volume.

By curve-fitting we compute λ and ν from our simulation setup for a range of transmitter-receiver distance and receiver volume (Table II). This characterization of the bacterial motion enables us to model the propagation of bacteria between different locations.

The probability of a bacterium arriving to the receiver in a time slot T_s is [30]

$$q = \int_0^{T_s} f(t)dt. \quad (2)$$

If N_0 bacteria are released from the transmitter at the beginning of the time slot, we can compute the number of bacteria arriving to the receiver, N_a, with the following binomial distribution

$$N_a(N_0) \sim Binomial(N_0, q). \quad (3)$$

When the number of bacteria released from the transmitter, N_0 is large, this binomial distribution can be approximated by a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ where the mean and variance are [41]

$$\mu = N_0 q, \qquad \sigma^2 = N_0 q(1-q). \quad (4)$$

According to Figure 7, it is observed that the probability distribution function for the first passage time has a long tail. Therefore, there will be bacteria arriving to the receiver after the intended time slot causing inter-symbol interference between each symbol sent in consecutive time slots. Hence, the total number of bacteria arriving in the current time slot, i.e., for the current symbol, is reformulated by adding the bacteria released in this time slot and the remaining bacteria released in the previous time slots causing the inter-symbol interference. If the time slot length is very large, the ISI effects will be lower but data rate will be slowed down too since there is more time between consecutive symbols. Hence, we choose the time slot length as short as possible after which the pdf of arrival of bacteria becomes flat. In other words, we choose the time slot length as the point where the pdf drops below 0.00005. In this study, for distances $d = 500, 1000, 1500$ μm with a receiver volume of 100 μm^3, the time slot lengths are chosen as $T_s = 1774, 4690, 7383$ $seconds$, respectively.

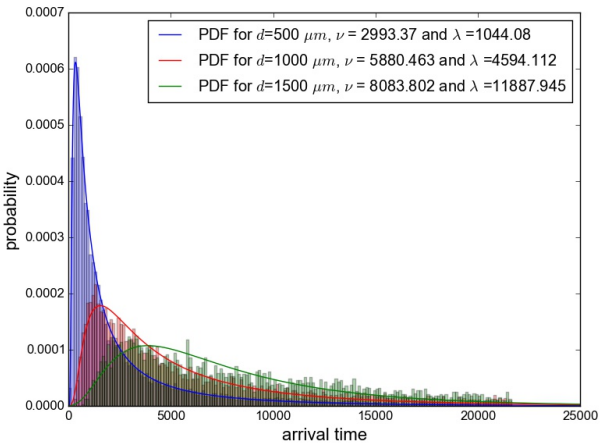


Fig. 8. Probability Distribution Functions (PDF) of the arrival time of the bacteria for various distances (receiver volume = 100 μm^3).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

When we calculate the probabilities that bacteria released in previous slots arrive in the current time slot, it is observed that the largest contribution comes from the immediate previous time slot. For example, for $d = 500\mu m$ and receiver volume of $100\mu m^3$, the probability that a bacteria arrives in the current time slot q is 0.6, whereas the probability from the immediate previous slot $q_p = 0.17$, and the probability for the third and forth slots are 0.07 and 0.03, respectively. Hence, we can assume that only the previous time slot contributes to the inter-symbol interference. This leads us to define the total number of bacteria arriving to the receiver as follows

$$N_T(s_c, s_p) = N_a(n_{s_c}) + N_p(n_{s_p}), \tag{5}$$

where n_{s_c} is the number of bacteria released in the current time slot and n_{s_p} is the number of bacteria released in the previous time slot, and the number of bacteria released in the previous slot but arriving in the current time slot is denoted as $N_p(n_{s_p})$. The s_c represents the symbol sent in the current time slot and s_p represents the symbol sent in the previous time slot.

$N_p(n_{s_p})$ can similarly be approximated by a Gaussian distribution $\mathcal{N}(\mu_p, \sigma_p^2)$ where the mean and the variance are

$$\mu_p = n_{s_p} q_p \qquad \sigma_p^2 = n_{s_p} q_p (1 - q_p), \tag{6}$$

where q_p is probability of bacteria which was released in the previous time slot arriving in the current time slot. This probability q_p is expressed as

$$q_p = \int_{T_s}^{2T_s} f(t)dt. \tag{7}$$

Since N_a and N_p are independent Gaussian random variables, the probability distribution of $N_T(s_c, s_p)$ becomes also Gaussian distributed $\mathcal{N}(\mu_T, \sigma_T^2)$ with mean and variance

$$\mu_T = \mu + \mu_p \qquad \sigma_T^2 = \sigma^2 + \sigma_p^2. \tag{8}$$

The N_T bacteria who arrived to the receiver will conjugate and transfer their message to the receiver bacteria according to the reception model discussed in the next section.

C. Reception Model

When the motile bacteria from the transmitter reach the receiver, they conjugate with the non-motile bacteria in the receiver to transfer the plasmids, in order to create the full set of genes required for bioluminescence. The number of conjugated receiver bacteria N_r increases with every incoming motile bacteria from the transmitter. Hence, the intensity of light due to bioluminescence increases with the number of incoming bacteria at the receiver. In this section, we denote the measured light intensity as $L(s_c, s_p)$ and we express it in terms of the total number of bacteria released from the transmitter arriving to the receiver $N_T(s_c, s_p)$.

Since the conjugation process takes a couple of minutes [15] which is very short compared to the propagation time in the order of hours [16], we can neglect the time required for conjugation.

In a bacteria population, conjugation does not take place between all the bacteria. Only a certain ratio of motile bacteria from the transmitter will conjugate with the bacteria at the receiver. We call the ratio of conjugated bacteria to the organisms released from the transmitter as transfer frequency and it is denoted by α_c. Hence, the number of receiver bacteria which are conjugated with the transmitter bacteria is found as $N_r = \alpha_c N_T$. The α_c is a parameter that relates to the bacterial species as well as environmental genetic factors [17].

When the density of conjugated bacteria reaches a critical density, the receiver starts shining light significantly. This phenomenon is called quorum sensing where bacterial cells produce a small autoinducer molecule which diffuses in and out of the cell whose concentration increases with the increasing bacterial cell density. The bioluminescence genes are controlled by this autoinducer molecule. When the autoinducer concentration increases above a threshold level, the bioluminescence genes become active and the light becomes observable.

The relation between the bacterial cell density and the autoinducer concentration is found by [26]

$$\frac{dA}{dt} = v_A N_r - d_A A, \tag{9}$$

where A is the autoinducer concentration, v_A is autoinducer production rate and d_A is the autoinducer degradation rate.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

The autoinducer forms a complex with bacterial cell receptors with a probability of $\rho(t)$. The dynamics of this probability is described by [27]

$$\frac{d\rho}{dt} = -\kappa\rho + A\gamma(1-\rho), \tag{10}$$

where κ is the dissociation rate and γ is the complex formation rate, and ρ represents the probability that autoinducer forms a complex with cell receptors.

When cell receptors are bound with the autoinducer, it activates the *lux* genes associated with bioluminescence. Although the detailed biochemistry of bioluminescence is unknown, the gene expression can be approximated by a two-step process for the production of the bioluminescent proteins and the light as

$$\begin{aligned} \frac{dS}{dt} &= (b_0\rho + a_0) - b_1 S \\ \frac{dL}{dt} &= a_1 S - b_2 L, \end{aligned} \tag{11}$$

where L is the amount of light, S is a post-transcriptional messenger, and b_0, a_0, a_1, b_1, b_2 are constants [27]. a_0 represents the basal production of bioluminescence proteins in the absence of autoinducer [28], b_0 represents the production rate of the post-transctriptional messenger in the presence of autoinducer, a_1 is the rate of light production, b_1 denotes the decay rate of the post-transcriptional messenger and b_2 denotes the decay rate of light.

At the end of the time slot, the system will come to a steady-state where the autoinducer concentration is

$$A_s = \frac{v_A N_r}{d_A}. \tag{12}$$

Accordingly, the probability of forming a complex at steady-state is

$$\rho_s = \frac{A_s\gamma}{A_s\gamma + \kappa}. \tag{13}$$

Then, the light intensity at the steady-state is expressed as

$$L_s = \frac{a_1(b_0\rho_s + a_0)}{b_1 b_2}. \tag{14}$$

In our study, the receiver is a population of bacteria whose members are noisy inherently. However, the effect of the discrepancies and uncertainties between each bacterium is less significant when the response of the population is studied instead of the response of each individual bacterium. Hence, we assume that the noise in the reception process is negligible compared to the noise resulting from the propagation process which is the main source of noise in this study.

In Section III-B, we have found that the number of bacteria arriving to the receiver N_T has a Gaussian distribution with mean μ_T and variance σ_T^2. Thus, the number of conjugated bacteria N_r can be easily described also by a Gaussian distribution with mean $\alpha_c\mu_T$ and variance ${\alpha_c}^2\sigma_T^2$.

Since the steady-state autoinducer concentration A is a linear function of the number of conjugated bacteria N_r, the probability of the autoinducer concentration also follows a Gaussian distribution with mean $\mu_A = (v_A/d_A)\alpha_c\mu_T$ and mean $\sigma_A^2 = (v_A/d_A){\alpha_c}^2\sigma_T^2$.

According to (13), the probability distribution of ρ is changing nonlinearly with autoinducer concentration A_s. Hence, the pdf of ρ can be described as follows

$$f_{\rho_s} = \frac{f_{A_s}\left(\frac{\rho_s\kappa}{\gamma(1-\rho_s)}\right)}{\gamma(1-\rho_s)^2/\kappa}, \tag{15}$$

where f_{A_s} is the Gaussian probability distribution of A_s.

Similarly, the probability distribution for the light can be found by

$$f_{L_s} = \frac{f_{\rho_s}\left(L_s b_1 b_2/(a_1 b_0)\right)}{a_1 b_0/(b_1 b_2)}. \tag{16}$$

IV. Modulation Schemes

In Section III, we described the bacterial propagation model as well as the reception model that will indicate a successful transfer of plasmids at the receiver. We investigated the propagation of bacteria from transmitter to the receiver and the reception by bacteria located in the receiver. Based on our simulation, as well as previous works, we have found that the propagation of bacteria suffers very long delays, which in turn will affect the end-to-end data rate of the communication system. In order to increase the rate of the information transfer, we suggest two modulation schemes exploiting the engineering plasmid property that allows us to program different combination of genes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

A. Modulation with a Single Receiver

In this modulation scheme, a single transmitter and receiver pair is considered.

1) Binary Density Modulation: When the information to be transmitted for the time slot is the symbol 0, no bacterium is sent from the transmitter. Since there is no transmitter bacterium arriving to the receiver, no bioluminescence is observed. However, when the information is the symbol 1, N_0 bacteria is released from the transmitter at the beginning of the time slot and when they deliver the message to the receiver bacteria by conjugation, the receiver bacteria produce visible light. To detect the information sent, the light intensity is compared to a threshold above which symbol 1 is decoded and below which symbol 0 is decoded. This modulation scheme resembles to an *ON-OFF* Keying modulation for conventional communication system.

We assume that the symbols for binary density modulation s_i can be either 0 or 1. Also, we assume that all symbols are equiprobable and independent of each other. For binary density modulation, the total probability of error can be calculated by

$$P_e = \sum_{s_c=0}^{1} P(s_c) P(\hat{s}_c \neq s_c | s_c) \tag{17}$$

where $P(s_c)$ is the a priori probability of transmitting symbol s_c and $P(\hat{s}_c \neq s_c|s_c)$ denotes the probability of incorrect decoding given the current symbol, where $\hat{s}_c$ is the current received symbol.

Since there are ISI effects, it is necessary to take into account the interference of the previous symbol on the current symbol. Thus, the incorrect decoding probabilities are expressed in terms of previous symbol as follows

$$P(\hat{s}_c \neq s_c|s_c) = \sum_{s_p=0}^{1} P(s_p) P(\hat{s}_c \neq s_c|s_c, s_p), \tag{18}$$

since the transmitted symbols are independent and the incorrect decoding probability $P(\hat{s}_c \neq s_c|s_c, s_p)$ depends on the current and previous symbols.

If we set the threshold for light intensity to τ_L, the incorrect decoding probabilities become

$$P(\hat{s}_c = 1|s_c = 0, s_p) = P(L_s(s_c, s_p) > \tau_L | s_c = 0, s_p) \tag{19}$$

$$P(\hat{s}_c = 0|s_c = 1, s_p) = P(L_s(s_c, s_p) < \tau_L | s_c = 1, s_p) \tag{20}$$

The light intensity $L_s(s_c, s_p)$ is found by replacing the number of released bacteria for the current and previous symbols n_{s_c} and n_{s_p} for $N_T(s_c, s_p)$ in (5) with

$$n_{s_i} = \begin{cases} N_0, & \text{if } s_i = 1 \\ 0, & \text{if } s_i = 0 \end{cases} \tag{21}$$

The threshold depends on the camera system used to measure the light and might change with the experimental setup, i.e., the sensitivity of the camera, ambient light, growth conditions of the bacteria and the bacteria species.

Since all the symbols are equally likely the error probability can be calculated by

$$P_e = \\ = \frac{1}{4}\left(\sum_{s_p=0}^{1} F_{L_s(1,s_p)}(\tau_L) + \sum_{s_p=0}^{1} (1 - F_{L_s(0,s_p)}(\tau_L)) \right). \tag{22}$$

where $F_{L_s(s_c,s_p)}$ is the cumulative density function of the pdf of $L_s(s_c, s_p)$ derived in (16).

2) M-ary Density Modulation: In *M-ary* density modulation, instead of using two symbols, we can introduce M symbols, i.e., M levels of bacterial density representing $log_2(M)$ bits. By thresholding the bioluminescence intensity at the receiver with $M - 1$ thresholds, one of these M levels can be decoded. This modulation scheme resembles to an *Amplitude Shift Keying* (*ASK*) modulation from conventional communication system.

We consider the case $M = 4$ representing 2 bits of information where transmitted symbols are $\{0, 1, 2, 3\}$ corresponding to $\{0, N_0/3, 2N_0/3, N_0\}$ released bacteria at the transmitter, respectively. For the rest of the text, M-ary density modulation refers to the modulation with $M = 4$. We further assume that all symbols are equally likely and independent from each other.

For M-ary density modulation, the total probability of error can be calculated by

$$P_e = \sum_{s_c=0}^{3} P(s_c) P(\hat{s}_c \neq s_c|s_c) \tag{23}$$

where $P(s_c)$ is the a priori probability of transmitting symbol s_c and $P(\hat{s}_c \neq s_c|s_c)$ denotes the incorrect decoding probability given s_c, where $\hat{s}_c$ is the current received symbol.

To take into account the effect of ISI, we further elaborate (23) by conditioning it with the previous

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

symbol s_p. We express the incorrect decoding probabilities given s_c when $s_p \in \{0,1,2,3\}$ as

$$P(\hat{s}_c \neq s_c|s_c) = \sum_{s_p=0}^{3} P(s_p)P(\hat{s}_c \neq s_c|s_c, s_p), \quad (24)$$

since the current and previous symbols are independent of each other.

To detect the 4 transmitted levels, 3 thresholds are needed. If we set the thresholds for light intensity to $\tau_{L_0}, \tau_{L_1}, \tau_{L_2}$, where τ_{L_i} differentiates between $\hat{s}_c = i$ and $\hat{s}_c = i+1$, the incorrect decoding probabilities become

$$P(\hat{s}_c \neq s_c|s_c = 0, s_p) = P(L_s(s_c,s_p) > \tau_{L_0}|s_c = 0, s_p) \quad (25)$$

$$P(\hat{s}_c \neq s_c|s_c = 1, s_p) = P(L_s(s_c,s_p) < \tau_{L_0} \text{ or } L_s(s_c, s_p) > \tau_{L_1}|s_c = 1, s_p) \quad (26)$$

$$P(\hat{s}_c \neq s_c|s_c = 2, s_p) = P(L_s(s_c,s_p) < \tau_{L_1} \text{ or } L_s(s_c, s_p) > \tau_{L_2}|s_c = 2, s_p) \quad (27)$$

$$P(\hat{s}_c \neq s_c|s_c = 3, s_p) = P(L_s(s_c,s_p) < \tau_{L_2}|s_c = 3, s_p) \quad (28)$$

The light intensity $L_s(s_c, s_p)$ is found by replacing the number of bacteria released for current and previous symbols n_{s_c} and n_{s_p} in $N_T(s_c, s_p)$ in (5) with

$$n_{s_i} = \begin{cases} N_0, & \text{if } s_i = 3 \\ 2N_0/3, & \text{if } s_i = 2 \\ N_0/3, & \text{if } s_i = 1 \\ 0, & \text{if } s_i = 0 \end{cases} \quad (29)$$

The incorrect decoding probabilities given in (25), (26), (27), (28) can be found by using the pdf of L_s given in (16).

If all the symbols are equally likely, the probability of error can be expressed as

$$\begin{aligned} P_e = \frac{1}{16} \Bigg(& \sum_{s_p=0}^{3} (1 - F_{L_s(0,s_p)}(\tau_{L_0})) \\ & + \sum_{s_p=0}^{3} (F_{L_s(1,s_p)}(\tau_{L_1}) + 1 - F_{L_s(1,s_p)}(\tau_{L_0})) \\ & + \sum_{s_p=0}^{3} (F_{L_s(2,s_p)}(\tau_{L_2}) + 1 - F_{L_s(2,s_p)}(\tau_{L_1})) \\ & + \sum_{s_p=0}^{3} F_{L_s(3,s_p)}(\tau_{L_2}) \Bigg) \end{aligned} \quad (30)$$

where $F_{L_s(s_c,s_p)}$ is the cumulative density function of the pdf of $L_s(s_c, s_p)$ derived in (16).

B. Modulation with Multiple Receivers

In this section, we introduce a novel modulation scheme called distributed modulation using bacterial nanonetworks where multiple transmitter and receiver pairs are used. Since the detection of the light is realized by a camera, the light intensity at different location in an image can be measured. Hence if we place M receivers in the environment spatially separated from each other, we can create a new modulation scheme where the transmitter between each pair no longer represents one bit but $log_2(M)$ bits as shown in Figure 2. In this study, we consider that $M = 4$, i.e., but the analysis can easily be extended to include more receivers. When an information 00 is to be sent, the associated transmitter releases bacteria which propagate in the environment reaching the complementary receiver.

Each transmitter-receiver pair is associated with one of the following symbols $\{0,1,2,3\}$ and only one transmitter-receiver pair is active at each time slot. Since we assume that the a priori probabilities of symbols are not known, we place the pairs such that transmitter-receiver distance is the same for all pairs. If it is known that a symbol has a higher probability, then the transmitter and the receiver of the corresponding pair can be positioned closer to increase the rate. Since we have 4 different transmitters and receivers, there are 4 light intensities to measure corresponding to each receiver which we denote as $L_{s,i}$ for i^{th} receiver. Each receiver is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

assumed to have the same threshold τ_L. If the light intensity of a receiver is above τ_L, the corresponding symbol will be received. We assume that all symbol are equally likely and independent of each other.

The total probability of error for this modulation scheme can be calculated by

$$P_e = \sum_{s_c=0}^{3} P(s_c) P(\hat{s}_c \neq s_c | s_c) \tag{31}$$

where $P_{s_c}(s_c)$ is the a priori probability of transmitting symbol s_c and $P(\hat{s}_c \neq s_c|s_c)$ denotes the incorrect decoding probability given s_c, where $\hat{s}_c$ is the current received symbol. Since all 4 transmitter-receiver pairs are parallel $P(\hat{s}_c \neq s_c|s_c)$ is the same for all symbols, i.e, for all tx-rx pairs. Hence, (31) becomes

$$P_e = P(\hat{s}_c \neq s_c|s_c). \tag{32}$$

To incorporate the effects of ISI, we condition the incorrect decoding probability with the previous symbol s_p and express it as

$$\begin{aligned} P(\hat{s}_c \neq s_c|s_c) = \\ = \sum_{s_p \neq s_c} P(s_p) P(\hat{s}_c \neq s_c | s_c, s_p, s_p \neq s_c) \\ + P(s_p = s_c|s_c) P(\hat{s}_c \neq s_c | s_c, s_p, s_p = s_c), \end{aligned} \tag{33}$$

where the first term corresponds to the case where the previous symbol is not equal to the current symbol, i.e., the previous symbol was sent from a different transmitter. One source of error in this case is that remaining bacteria from the previous symbol activating the receivers for the other symbols than the current one. The other source of error is that there is not enough bacteria released from the transmitter of the current symbol to activate the corresponding receiver. The second term of (33) represents the case where the previous symbol is equal to the current symbol. In this case, the number of bacteria from the previous symbol is added to the number of bacteria for the current symbol. The only source of error is that there is not enough bacteria to activate the intended receiver. There is no possibility that the other receivers will be activated since there is no remaining bacteria from the previous slot for them.

If we set the thresholds for the light intensity to τ_L for each pair, the incorrect decoding probabilities are expressed as

$$\begin{aligned} P(\hat{s}_c \neq s_c | s_c, s_p, s_p \neq s_c) = \\ = P(L_{s,s_p}^{(s_p \neq s_c)} > \tau_L | s_c, s_p, s_p \neq s_c) \\ + P(L_{s,s_c}^{(s_p \neq s_c)} < \tau_L \ \& \ L_{s,s_p}^{(s_p \neq s_c)} < \tau_L | s_c, s_p, s_p \neq s_c), \end{aligned} \tag{34}$$

$$\begin{aligned} P(\hat{s}_c \neq s_c | s_c, s_p, s_p = s_c) = \\ = P(L_{s,s_c}^{(s_p = s_c)} < \tau_L | s_c, s_p, s_c = s_p). \end{aligned} \tag{35}$$

$L_{s,s_c}^{(s_c,s_p)}$ is found by replacing n_{s_c} and n_{s_p} in $N_T(s_c, s_p)$ in (5) with $n_{s_c} = N_0$ and with

$$n_{s_p} = \begin{cases} N_0, & \text{if } s_c = s_p \\ 0, & \text{if } s_c \neq s_p \end{cases} \tag{36}$$

$L_{s,s_p}^{(s_c,s_p)}$ is found by replacing n_{s_c} and n_{s_p} in $N_T(s_c, s_p)$ in (5) with $n_{s_c} = 0$ and with

$$n_{s_p} = \begin{cases} 0, & \text{if } s_c = s_p \\ N_0, & \text{if } s_c \neq s_p \end{cases} \tag{37}$$

The incorrect decoding probabilities (34, 35) can be found by using the probability distribution of L_s given in (16).

If all the symbols are equally likely, the probability of error can be expressed as

$$\begin{aligned} P_e = \frac{3}{4} \left(1 - F_{L_{s,s_p}^{(s_c \neq s_p)}}(\tau_L) + F_{L_{s,s_c}^{(s_c \neq s_p)}}(\tau_L) F_{L_{s,s_p}^{(s_c \neq s_p)}}(\tau_L) \right) \\ + \frac{1}{4} F_{L_{s,s_c}^{(s_c = s_p)}}(\tau_L); \end{aligned} \tag{38}$$

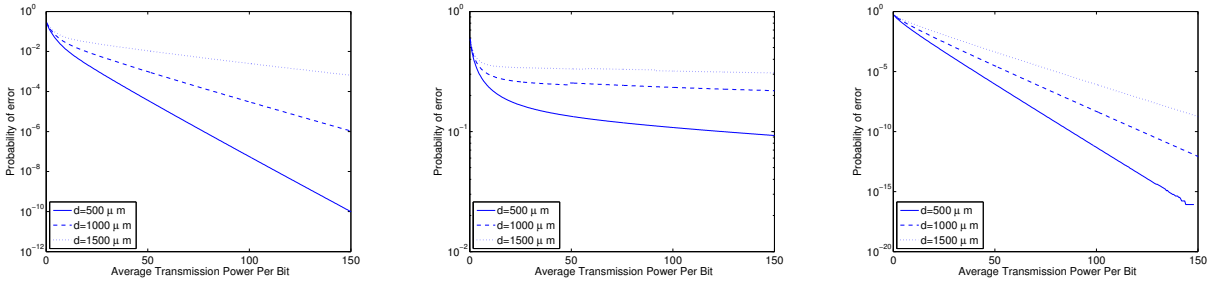
where $F_{L_{s,i}(s_c,s_p)}$ is the cumulative density function of the pdf of $L_s(s_c, s_p)$ derived in (16).

C. Achievable Rate

We define the achievable rate R that maximizes the mutual information between the transmitted symbol and the received symbol as follows

$$\begin{aligned} R = \\ = \max_{\tau_L} \ I(X;Y) \\ = \max_{\tau_L} \sum_X \sum_Y P(X,Y) log_2 \left(\frac{P(X,Y)}{P(X)P(Y)} \right). \end{aligned} \tag{39}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience



(a) (b) (c)

Fig. 9. Probability of error for different modulation schemes for varying distances (a) Binary Density Modulation. (b) M-ary Density Modulation. (c) Modulation with Distributed Receivers.

V. Performance Evaluation

In this section, to compare the performance of the three different modulation schemes we evaluate the probability of bit error and achievable rates against the average transmission power per bit for each type of the modulation that we proposed in Section IV. In this paper, the transmission power is defined as the number of bacteria released from the transmitter. For fair comparison between modulation schemes, the average transmission power per bit, i.e, average number of bacteria released per bit is used. Firstly, we use the simulations from Section III-B to observe the arrival times of bacteria for varying transmitter-receiver distances. The inverse Gaussian model that we fitted to the simulation for the arrival time shows that the arrival probability does not change significantly after a certain time due to its long flat tail. Hence, we chose $d = 500, 1000, 1500$ μm with a receiver volume of 100 μm^2, the time slot lengths are chosen as $T_s = 2557, 6159, 9095$ $seconds$, respectively as discussed in Section III-B.

Using these T_s values, the probability of error and achievable rate of the three different modulation schemes are evaluated for optimum thresholds values which are found by minimizing the probability of errors for varying transmission powers. In Figure 9, the probability of errors for each modulation scheme are shown for varying transmitter-receiver distances. For all modulation schemes, probability of error is decreasing significantly with the increasing average transmission power per bit, i.e., the number of bacteria released from the transmitter which is expected. When the number of bacteria released from the transmitter increases, the number of bacteria arriving the receiver in- creases in turn increasing the correct detection probability. This decrease in probability of error is less significant for *M-ary* density modulation since increasing transmission power increases the separation between symbol levels while also contributing to the ISI where the residual bacteria from the previous symbol cause errors. Furthermore, as seen from Figure 9, the distance has a considerable effect on probability of error for all three modulation schemes. Since the bacteria propagate randomly, it is harder to reach the receiver when they have to travel longer distances which results in an increase in probability or error.

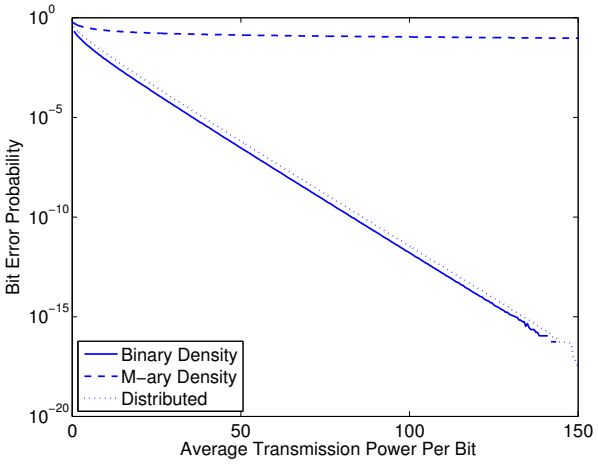


Fig. 10. Bit Error Probability comparison between Binary Density, M-ary Density, and Distributed Modulation schemes.

To compare the probability of errors of the proposed modulation schemes, the probability of errors are converted to bit error probability for fair comparison. In Figure 10, bit error probabilities for $d = 500\mu m$ are plotted versus the average transmission power per bit. Distributed modulation and the binary density modulation perform very

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

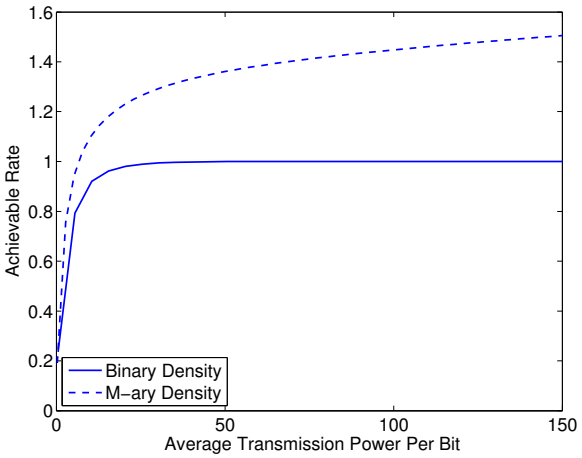


Fig. 11. Achievable rate for binary density and m-ary density modulation.

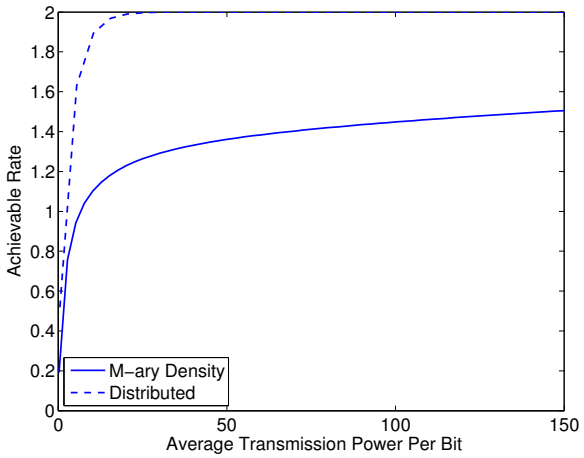


Fig. 12. Achievable rate for m-ary density and distributed modulation.

similarly, the probability of error decreases with the increasing transmission power per bit. However, for M-ary density modulation, increasing the transmission power does not ameliorate the bit error probability.

In this study, we considered *M-ary* modulation with $M = 4$, transmitting $log_2(M) = 2$ bits per symbol. The achievable rates comparison between binary and *M-ary* modulation with $M = 4$ is illustrated in Figure 11. For *M-ary* modulation, the achievable rate increases very quickly with increasing transmission power whereas for binary modulation, the achievable rate requires a lot of power for a small increase in rate.

Similarly, the achievable rates of *M-ary* density modulation and distributed modulation schemes are compared in Figure 12. Asymptotically, both schemes reach the rate of 2 bits per/slot as expected. However, to attain the same rate, distributed modulation requires larger quantity of less transmission power than the *M-ary* density modulation. Considering the distributed modulation's lower error probability and higher rate, it can be considered as an efficient modulation scheme.

Another factor influencing the achievable rate is the length of time slot. To examine its effect, the achievable rate versus time slot length is presented in Figure 13 for various transmission powers. When the duration of the time slot increases, there are more bacteria reaching the receiver and delivering its message. Hence, the rate per time slot increases.

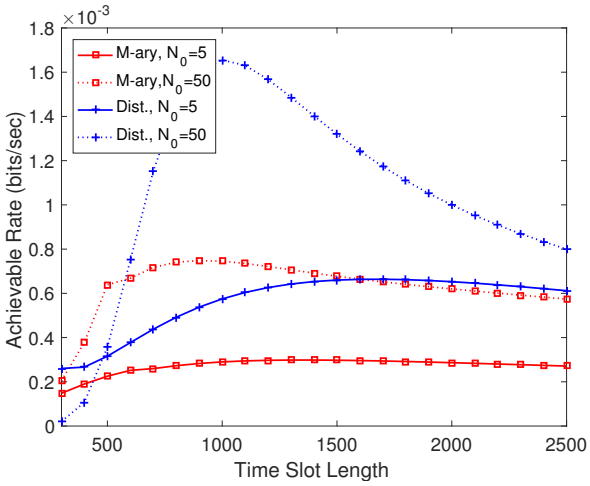


Fig. 13. Achievable rate for m-ary density and distributed modulation.

However, when the time slot becomes too large, this rate is divided by the large time slot length which in turn decreases the rate per second.

VI. Conclusion

Bacterial nanonetworks is one proposed model for molecular communication that utilizes bacteria as information carriers between the transmitter and receiver. While this bio-compatible approach can allow information to be transmitted up to millimeter distances, there are numerous complexities in developing encoding techniques of the plasmids at the transmitter, as well as decoding at the receiver.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNB.2017.2706757, IEEE Transactions on NanoBioscience

A simpler approach could be achieved through ON-OFF keying where the population of the bacteria represents the digital bits that are to be transmitted. However, the long propagation period of the bacteria leads to low data rate. In order to improve the performance, this paper proposed incorporating another property which is the encoding of different combination of genes into the plasmid, where the different combinations can represent a series of bits. The transmitter motile bacteria will swim towards the non-motile bacteria at the receiver to conjugate and transfer the plasmids with the encoded genes. This will lead to the non-motile bacteria at the receiver to receive a full set of genes that will lead to bioluminescence. Through these different combination of genes, parallel transmission of bits can be achieved, and this in turn will lead to lower bit error probability as well as higher achievable data rate. The performance evaluation compared the distributed modulation scheme presented in this paper with the Binary Density Modulation as well as the M-ary Density Modulation scheme, and found that the performance improvement can be established for varying distances, quantity of bacteria emitted, as well as time slots. This proposed approach has shown how incorporating other known cellular functions, such as engineering different combination of genes into the plasmids, can be incorporated into bacterial nanonetworks to further improve their performance. This would, therefore, lay the foundation for incorporating other functionalities and properties in the future to further improve the performance and open up new opportunities for novel healthcare applications.

REFERENCES

[1] I. F. Akyildiz, F. Brunetti, and C. Blazquez, "Nanonetworks: A New Communication Paradigm," *Computer Networks,* June 2008.

[2] T. Nakano, M. Moore, F. Wei, A. T. Vasilakos, and J. W. Shuai, "Molecular Communication and Networking: Opportunities and Challenges," *IEEE Transactions on NanoBioscience,* vol. 11, no. 2, pp. 135-148, June 2012.

[3] N. Farsad, et al., "A comprehensive survey of recent advancements in molecular communication," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 3, pp. 1887-1919, 2014.

[4] M. Sitti, "Miniature devices: Voyage of the microrobots," *Nature,* vol. 458, no. 7242, pp. 1121-1122, 2009.

[5] M. Gregori, et al. "Physical channel characterization for medium-range nanonetworks using flagellated bacteria." Computer Networks 55.3 (2011): 779-791.

[6] L. C. Cobo, and I. F. Akyildiz, "Bacteria-based communication in nanonetworks," *Nano Communication Networks,* vol. 1, no. 4, pp. 244-256, 2010.

[7] S. Abadal and I. F. Akyildiz, "Bio-inspired synchronization for nanocommunication networks," in *Proc. 2011 Global Telecommunications Conference - Wireless Networking Symposium,* pp. 15.

[8] G. Wei, P. Bogdan and R. Marculescu, Bumpy Rides: Modeling the Dynamics of Chemotactic Interacting Bacteria, in IEEE Journal on Selected Areas in Communications, vol. 31, no. 12, pp. 879-890, December 2013.

[9] P. Bogdan, G. Wei and R. Marculescu, Modeling Populations of Micro-robots for Medical Appli- cations, in Proc. of the 2nd IEEE International Workshop on Molecular and Nanoscale Communications, Ottawa, Canada, June, 2012.

[10] G. Wei, P. Bogdan, and R. Marculescu, Efficient Modeling and Simulation of Bacteria-based Nanonetworks with BNSim, IEEE Journal on Selected Areas in Communications - 2013 Special Issue on Emerging Technologies in Communications, 2013.

[11] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy, "Internet of BioNanoThings," *IEEE Communications Magazine,* vol. 53, no. 3, pp. 32-40, March 2015.

[12] M. Pierobon, and I. F. Akyildiz, "A Physical End-to-End Model for Molecular Communication in Nanonetworks," *IEEE Journal of Selected Areas in Communications,* vol. 28, no. 4, pp. 602-611, May 2010.

[13] M. Kuscu, and O. B. Akan, "A Communication Theoretical Analysis of FRET-Based Mobile Ad Hoc Molecular Nanonetworks," *IEEE Transactions on Nanobioscience,* vol. 13, no. 3, pp. 255-266, September 2014.

[14] M. Barros, S. Balasubramaniam, B. Jennings, and Y. Koucheryavy, "Transmission Protocols for Calcium-Signaling-based Molecular Communications in Deformable Cellular Tissue," *IEEE Transactions on Nanotechnology,* vol. 13, no. 4, pp. 779-788, 2014.

[15] F. Bonhoeffer, "DNA transfer and DNA synthesis during bacterial conjugation," *Zeitschrift fr Vererbungslehre,* vol. 98, no. 2, pp. 141-149, 1966.

[16] S. Balasubramaniam, and P. Lio, "Multi-hop Conjugation based Bacteria Nanonetworks," *IEEE Transactions on NanoBioscience,* vol. 12, no. 1, pp.47-59, March 2013.

[17] P. Trieu-Cuot, et al., "Plasmid transfer by conjugation from Escherichia coli to Gram-positive bacteria," *FEMS Microbiology Letters,* vol. 48, no. 1-2, pp. 289-294, 1987.

[18] V. Petrov, D. Moltchanov, S. Balasubramaniam, and Y. Koucheryavy, "Incorporating Bacterial Properties for Plasmid Delivery in Nano Sensor Networks," *IEEE Transactions on Nanotechnology,* vol. 14, no. 4, 2015.

[19] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems,* vol. 22, no. 3, pp. 52-67, 2002.

[20] Y. Okaie, T. Nakano, T. Hara, K. Hosoda, Y. Hiraoka, and S. Nishio, "Cooperative Target Tracking by a Mobile Bionanosensor Network," *IEEE Transactions on Nanobioscience,* vol. 13, no. 3, pp. 267 - 277, Sept. 2014.

[21] T. E. Gorochowski, et al., "BSim: an agent-based tool for modeling bacterial populations in systems and synthetic biology," *PloS one,* vol. 7, no. 8, e42790, 2012.

[22] B. Krishnaswamy, et al., "When bacteria talk: Time elapse communication for super-slow networks," *Communications (ICC), 2013 IEEE International Conference on,* 2013.

[23] J. L. Folks, and R. S. Chhikara, "The inverse Gaussian distribution and its statistical application–a review," *Journal of the Royal Statistical Society Series B (Methodological),* pp. 263-289, 1978.

[24] M.U. Mahfuz, D. Makrakis, and H.T. Mouftah, "On the characterization of binary concentration-encoded molecular communication in nanonetworks," *Nano Communication Networks Journal, Elsevier Science,* vol. 1, pp. 289-300, 2010.

[25] P. C. Yeh, et. al., "A new frontier of wireless communication theory: diffusion-based molecular communications," *IEEE Wireless Communications,* vol. 19, no. 5, 2012.

[26] L. You, et al. "Programmed population control by cell-cell communication and regulated killing," *Nature,* vol. 428, no. 6985, pp. 868-871, 2004.

[27] J. Mller, C. Kuttler, and B. A. Hense, "Sensitivity of the quorum sensing system is achieved by low pass filtering," *Biosystems,* vol. 92, no. 1, pp. 76-81, 2008.

[28] E. A. Meighen, "Genetics of bacterial bioluminescence," *Annual review of genetics,* vol. 28, no. 1, pp. 117-139, 1994.

[29] A. Guney, B. Atakan, and O. B. Akan, "Mobile ad hoc nanonetworks with collision-based molecular communication," *IEEE Transactions on Mobile Computing,* vol. 11, no. 3, pp. 353-366, 2012.

[30] T. Nakano, et al., "Channel model and capacity analysis of molecular communication with brownian motion," *IEEE communications letters,* vol. 16, no. 6, pp. 797-800, 2012.

[31] T. Zhou, L. Chen, and K. Aihara, "Molecular communication through stochastic synchronization induced by extracellular fluctuations," in *Physical Review Lett.,* Oct. 2005.

[32] H. ShahMohammadian, G. G. Messier, and S. Magierowski, "Blind synchronization in diffusion-based molecular communication channels," *IEEE communications letters,* vol. 17, no. 11, pp. 2156-2159, 2013.

[33] N. C. Darnton, et al., "On torque and tumbling in swimming Escherichia coli," *Journal of bacteriology,* vol. 189, no. 5, pp. 1756-1764, 2007.

[34] L. Cong, et al., "Multiplex genome engineering using CRISPR/Cas systems," *Science,* vol. 339, no. 6121, pp. 819-823, 2013.

[35] M. K. Winson, et al., "Construction and analysis of luxCDABE-based plasmid sensors for investigating N-acyl homoserine lactone-mediated quorum sensing," *FEMS microbiology letters,* vol. 163, no. 2, pp. 185-192, 1998.

[36] E. A. Meighen, "Bacterial bioluminescence: organization, regulation, and application of the lux genes," *The FASEB journal,* vol. 7, no. 11, pp. 1016-1022, 1993.

[37] D. B. Clewell, ed., *Bacterial conjugation,* Springer Science & Business Media, 2013.

[38] S. J. Sorensen, et al., "Studying plasmid horizontal transfer in situ: a critical review," *Nature Reviews Microbiology,* vol. 3, no. 9, pp. 700-710, 2005.

[39] L. Simonsen, et al., "Estimating the rate of plasmid transfer: an end-point method," *Microbiology,* vol. 136, no. 11, pp. 2319-2325, 1990.

[40] D. M. Gordon, "Rate of plasmid transfer among Escherichia coli strains isolated from natural populations," *Microbiology,* vol. 138, no. 1, pp. 17-21, 1992.

[41] M. S. Kuran, et al., "Modulation techniques for communication via diffusion in nanonetworks," *Communications (ICC), 2011 IEEE International Conference on,* 2011.

[42] E. A. Codling, M. J. Plank, and S. Benhamou. "Random walk models in biology." Journal of the Royal Society Interface 5.25 (2008): 813-834.

[43] N. A. Ruhi and P. Bogdan, Multiscale modeling of biological communication, 2015 IEEE International Conference on Communications (ICC), London, 2015, pp. 1140-1145.

[44] G. Castorina, L. Galluccio, and S. Palazzo, "On Modeling Information Spreading in Bacterial Nano-Networks Based on Plasmid Conjugation," *IEEE Transactions on NanoBioscience,* vol. 15, no. 6, pp. 567-575, 2016.

[45] S. Balasubramaniam, et al. "Exploiting bacterial properties for multi-hop nanonetworks." IEEE Communications Magazine 52.7 (2014): 184-191.

[46] Gorochowski, Thomas E., et al. "BSim: an agent-based tool for modeling bacterial populations in systems and synthetic biology." PloS one 7.8 (2012): e42790.

Appendix B

Genetic similarity of biological samples to counter bio-hacking of DNA sequencing functionality

Journal Title	Scientific Reports, Nature
Article Type	Regular
Complete Author List	Mohd Siblee Islam, Stepan Ivanov, Eric Robson, Tríona DooleyCullinane, Lee Coffey, Kevin Doolin and Sasitharan Balasubramaniam

SCIENTIFIC REPORTS

OPEN

Genetic similarity of biological samples to counter bio-hacking of DNA-sequencing functionality

Received: 23 November 2018
Accepted: 29 May 2019
Published online: 18 June 2019

Mohd Siblee Islam[1], Stepan Ivanov[2], Eric Robson[2], Tríona Dooley-Cullinane[3], Lee Coffey[3], Kevin Doolin[2] & Sasitharan Balasubramaniam[2,4]

We present the work towards strengthening the security of DNA-sequencing functionality of future bioinformatics systems against bio-computing attacks. Recent research has shown how using common tools, a perpetrator can synthesize biological material, which upon DNA-analysis opens a cyber-backdoor for the perpetrator to hijack control of a computational resource from the DNA-sequencing pipeline. As DNA analysis finds its way into practical everyday applications, the threat of bio-hacking increases. Our wetlab experiments establish that malicious DNA can be synthesized and inserted into *E. coli*, a common contaminant. Based on that, we propose a new attack, where a hacker to reach the target hides the DNA with malicious code on common surfaces (e.g., lab coat, bench, rubber glove). We demonstrated that the threat of bio-hacking can be mitigated using dedicated input control techniques similar to those used to counter conventional injection attacks. This article proposes to use genetic similarity of biological samples to identify material that has been generated for bio-hacking. We considered freely available genetic data from 506 mammary, lymphocyte and erythrocyte samples that have a bio-hacking code inserted. During the evaluation we were able to detect up to 95% of malicious DNAs confirming suitability of our method.

In recent years, the field of bioinformatics has undergone a noticeable transformation due to the advancements in both genomics and DNA-sequencing equipment. On one hand, current knowledge of DNA structures contributes immensely to a variety of biological and medical applications from disease screening[1] to plant[2] and animal[3] breeding, to forensics[4]. On the other hand, radical new approaches such as DNA-based data storage[5] have extended the use of DNA-related technologies. The arrival of the MinION sequencer[6] and its associated technologies has also significantly increased the accessibility of DNA-analysis to the general public, and in the future will become an essential hardware for a range of industrial applications.

A recent study[7] has demonstrated a new form of vulnerability that DNA-sequencing can be susceptible to. The study shows how an adversary can insert a malicious payload from a computer script into a DNA sequence of a biological sample. The inserted payload takes advantage of a specific binary vulnerability of software used in the DNA-sequencing pipeline. The pipeline assembles the DNA-structure of a sample from the output of a DNA-sequencing instrument (i.e. FASTQ files). Then, the payload creates and opens a reverse shell to a remote address and port for the adversary to seize control of computational resources hosting the affected software. Though hosted separately from the sequencing instrument, the pipeline is an essential part of the DNA-sequencing process. Control of the pipeline will allow the attacker to eavesdrop on and even sabotage future DNA analyses. This may lead to consequences including misdiagnosis of illnesses, use of wrong DNAs for criminal forensics investigations, or suboptimal animal and plant breeding. In this paper we consider (i) a new scenario of attack on DNA sequencing pipeline, and (ii) input-control for detecting the DNA with encoded malicious code that is used for hacking. The following sub-sections will provide a brief introduction to each of these contributions.

[1]McAfee Ireland Ltd., Building 2000, City Gate, Mahon, Cork, Ireland. [2]Telecommunications Software and Systems Group, Waterford Institute of Technology, Waterford, Ireland. [3]Pharmaceutical and Molecular Biotechnology Research Centre, Waterford Institute of Technology, Waterford, Ireland. [4]Faculty of Information and Communication Sciences, Tampere University, Tampere, Finland. Correspondence and requests for materials should be addressed to S.I. (email: sivanov@tssg.org)

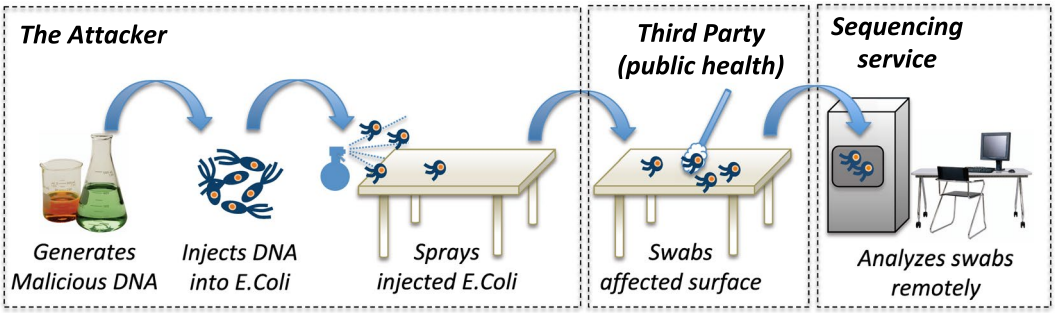


Figure 1. Synthesis of DNA with encoded malicious code, physical transport of the malicious DNA to the targeted remote sequencer.

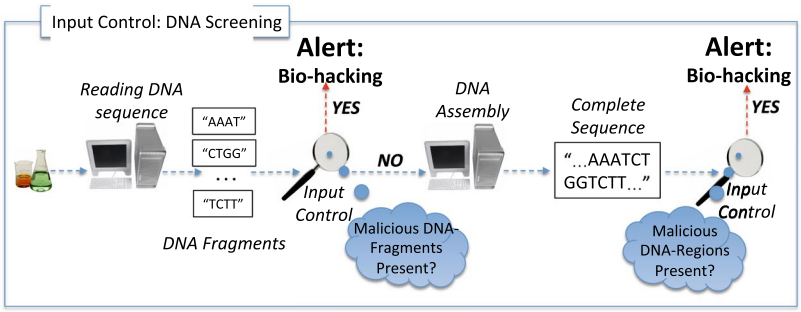


Figure 2. Input control to detect DNA-injections.

Physical Transport of Malicious DNA

In this article we consider an end-to-end execution of the attack on the DNA-sequencing pipeline, starting from generation of malicious DNA, to its delivery to the intended target and finally sequencing. The delivery takes the form of spraying the DNA containing malicious code onto different materials, such as a lab coat, glove, or a lab bench. Figure 1 presents an example of our attack scenario where the malicious DNA is injected into *E. Coli plasmids*, which in their raw form or injected into *E. Coli* bacteria are sprayed on a common surface (e.g. in a restaurant kitchen). The surface is swabbed by a third-party (e.g. during a routine health-and-safety inspection) and the swabs are sent for analysis to an external DNA-sequencing service (e.g. to detect the exact *E. Coli* strains present). The DNA-sequencing service is the intended target for the attack. Such scenarios will become more and more prevalent in the future. Due to advances in Cyber-Security, it will become increasingly difficult to gain control over a remote service using software-only vulnerabilities. Therefore, hackers will resort to more sophisticated approaches, such as the attacks we consider in this paper, where malicious code is delivered via DNA samples. These attacks represent a biological version of the *injection* practices used by hackers today.

Input-Control for Detecting Bio-Hacking

Computer Science has been dealing with injection attacks for some time. Thus, Ron *et al.*[8] present an overview of injection attacks on NoSQL data-storage systems. Similarly, Tsoutsous and Maniatakos[9] review the attacks on Embedded Systems. To neutralize the threat identified in[7], computer science offers a number of solutions. As demonstrated in[10], certain hardware functionality (e.g. Intel Memory Protection Expansion) may be successfully used to address some of the underlying memory-access issues. However, such solutions are naturally hardware-specific and, therefore, cannot be applied across-the-board. Alternatively, memory-access can be tightened at the Operating System's level[11]. While this loosens hardware-dependency of such solutions, their applicability is still limited. Finally, at the application level, injection attacks are successfully countered by the input-control techniques. For example[12], uses input-control as a countermeasure to an injection-based attack on a system managing an electrical grid[13]. Runtime Application Self-Protection described in[8] is an input-control technique proposed for NoSQL systems. Though built for a particular application, these techniques are compatible with multiple configurations of hardware, middleware, and operating systems. This property is particularly important for protecting a DNA-sequencing pipeline that may consist of a number of diverse computational resources.

In this paper, we evaluate the suitability of using input-control techniques against malicious DNA-injections (Fig. 2). As the main purpose of this article is to establish suitability of input-control, the evaluation is done only for a limited variety of samples. While this is sufficient for the problem at hand, the techniques can be readily extended to account for other samples (see the last paragraph in *Section 2.2*). We propose an input-control technique that tightly aligns with the typical stages of the DNA-sequencing process, namely *Reading DNA-Fragments* and *DNA-assembly*[14]. During *the first stage* a complete DNA sample is divided into multiple fragments that are

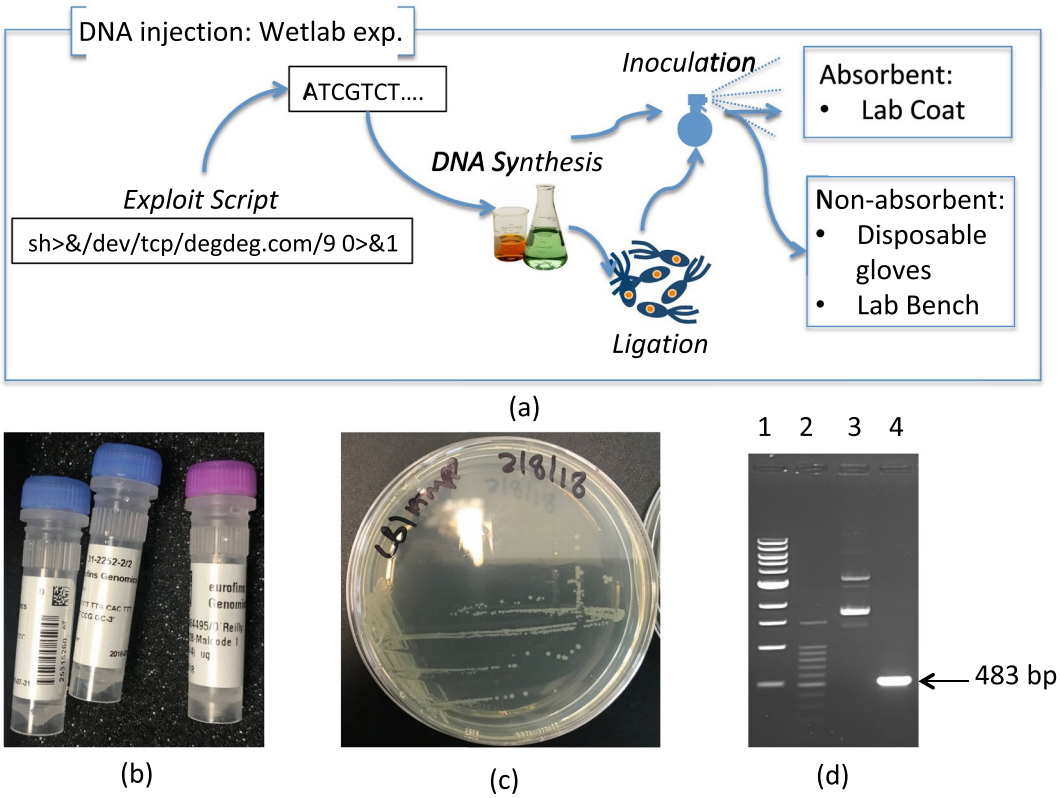


Figure 3. Wetlab Experimental Setup. Gel-Lane content on subfigure (c) Lanes: 1 - NEB 1Kb DNA-ladder; 2 - Promega 100 bp DNA-ladder; 3 - pMal1, complete plasmid; 4 - malicious segment isolated.

sequenced individually to speed up the overall process. As the result of this step, the chemical structure of each fragment is represented as a string of the "A", "C", "T", "G" letters representing DNA's corresponding nucleic acids. During *the second stage* the fragments are assembled into the complete DNA-sequence that describes the structure of the entire DNA sample. The two stages present two distinct points where an injection of malicious DNA-content can be detected. The detection algorithm will stop any further processing of the malicious sample, and this is one of the contributions of this paper.

Methodology

Methodologically, the study presented in this article can be separated into two stages. During the first stage we demonstrate the reality of the threat of bio-hacking. We conducted a series of wetlab experiments to prove that *E. coli* bacteria can be successfully used as a viable carrier for malicious code that is encoded into its DNA plasmid. *E. coli* bacteria can be used to host, multiply and preserve the code within the environment and/or transport the code to its intended target as part of the physical transport. During the second stage we showed that insertion of malicious code into a DNA sequence reduces its genetic similarity with other organisms from the same biological specie. We used this fact to propose a screening algorithm to detect malicious code in DNA.

Biohacking as a valid threat. We conducted a series of wetlab experiments to prove the ability of *E. coli* bacteria to serve as malicious code carriers. Specifically, we considered the stability of *E. Coli* plasmid DNAs. It has been shown that certain *E. Coli* plasmids can be used as the media for long-term data storage (e.g. up to 20 days in Accelerated Aging Conditions with temperature of 65C[15]). However, *E. Coli* plasmids with encoded malicious code do not portray the typical characteristics seen in bacterial hosts nor has their stability been confirmed over multiple repetition studies. Furthermore, even minor sequencing errors (e.g. incorrect base-calling) have the potential to alter the malicious-code and render it non-functional (e.g. introduce spelling mistakes into the shell code). These errors may occur due to reasons stretching from exposure to UV, heat, chemicals[16] to phasing effect and other sequencing problems[17]. Therefore, the stability of malicious payload recovery from the manipulated *E. Coli* plasmids requires additional confirmation. To do so we reproduced possible steps of a hacker trying to attack a DNA-sequencing service. We first produced *E. Coli* plasmids with malicious code integrated into their DNA. Next we evaluated the recovery of malicious DNA from various services sprayed by the hacker as part of the physical transport mechanism.

Figure 3(a) shows a high-level overview of the experimental design. We successfully inserted via ligation the code from[7] as a DNA sequence into the plasmid (pEX-A128, see Fig. 3(b)) and designated the final recombinant plasmid as pMal1. This DNA sequence was synthesised by Eurofins Genomics, Germany. The plasmid DNA material was then successfully transformed into a population of *E. coli* cells (Novablue strain (Novagen)) as shown on Fig. 3(c). Plasmid DNA (prepared using the Monarch Plasmid MiniPrep kit (New England Biolabs)) and

recombinant *E. coli* containing the plasmids were separately inoculated onto three surfaces: wooden lab bench, nitrile disposable glove (both non-absorbent) and cloth/labcoat (absorbent). The study aimed to establish if the malicious DNA material could be recovered through swabbing from surfaces several hours past spraying if a hacker was to transport the DNA. In doing so, we model physical malicious code delivery by either live *E. coli* or residual plasmids of non-viable bacteria that remained on various surfaces. Controls were carried out to ensure integrity of the wet lab experiments. Negative controls consisted of sterile ultrapure water being added to identical surfaces. That was done to ensure that only DNA material introduced during the experiments would be detectable by our methods. Standard *E. Coli* plasmid DNA samples were used as positive controls to confirm credibility of the DNA recovery through swabbing. During controls and core experiments of the study, samples were left on the surface for 24 hours and dried completely before swabbing.

To each surface, 500 ng of plasmid DNA or 1 µl of cell suspension @ $O.D._{600} = 12.5$ (approximately ten million cells calculated using Agilent Genomics BioCalculator) were inoculated to a 1 cm^3 area. Both dry and wet swabs (swabs pre-moistened with sterile TE buffer) were used to recover plasmid DNA and *E. coli* bacteria from each surface using the cross hatch swabbing technique. The swabs were resuspended in TE buffer and all DNA preparations were quantified using the Qubit3™ Fluorometer and Qubit™ dsDNA BR kit (Thermo Scientific), with cell suspensions measured using a NanoDrop ND-1000™ at an optical density of 600 nm. The isolated material was also subjected to PCR to detect the presence of the DNA material injected with malicious code i.e. pMal1. Oligonucleotides (pEX-For and pEX-Rev) were supplied by Eurofins, MWG operon, Germany. PCR conditions used to amplify Malcode 1 were as follows; each 15 µL PCR reaction mixture contained 7.5 µL Q5® High-Fidelity DNA Polymerase Master Mix (NEB), 15 pmol of each primer and 15 ng pMal1 plasmid DNA. PCR conditions include: 1 cycle of 95 °C for 5 min, 30 cycles of 95 °C for 1 min, 66 °C for 1 min, 72 °C for 30 s, 1 final extension stage of 8 minutes. PCR amplification limits were tested by adding decreasing amounts of pMal1 plasmid DNA to reactions via serial dilution of template. PCR products were analyzed by agarose gel electrophoresis. Figure 2(d) presents complete pMal1 plasmid DNA (lane 3) and PCR amplification of a segment from pMal1 DNA containing the malicious code sequence (lane 4). Sequencing of plasmid insert was carried out in triplicate using the vector-specific primers pEX-For and pEX-Rev, with all sequencing carried out by GATC Biotech, Germany.

Genetic similarity as a counter-measure. During the genetic similarity analysis we made extensive use of Genetic Signal Processing (GSP) techniques. For an organism (e.g. *E. coli* bacterium), GSP works with the string representation of the DNA structure. The structure is presented as a sequence of "A", "C", "T" and "G" symbols corresponding to the 4 DNA nucleic acids. The sequence is transformed into a continuous signal (often referred to as *Genetic Signal*) that is then analysed using various Signal Processing techniques. This research used the Voss transformation[18] to obtain Genetic Signals from DNA strings. The transformation had previously proven efficient in multiple studies on similarities of DNA within biological types, classes and families[19]. Results of the Voss transformation were subjected to Discrete Fourier Transform (DFT). Results of DFT formed the features (specifically Energy Values of DFT's Frequency Spectrums obtained for DNA Voss transform as presented in the Appendices) that we used to establish dissimilarity between the original and injected DNAs (proven technique, same as in[20]). See Appendix A of the Supplementary Material. The overall transformation represents an arbitrary DNA sequence (may be rather large in length) as a tuple of 20 floating-point numbers. This representation significantly reduces complexity of distance-wise comparisons between DNA sequences. Consider a distance between two DNA sequences whose lengths are m and n. Then, complexity of traditional methods such as Needleman-Wunsch (part of popular BLAST framework) estimated as $\boldsymbol{O(nm)}$, while complexity of calculating Euclidean distance in $\mathbf{R^{20}}$ (used in this article) is only $\boldsymbol{O(1)}$. This reduction is particularly important when such comparisons may consider multitude of species and their variations.

DNA Robustness to malicious injections. Similarity of Voss genetic signals was used to evaluate the effect of malicious code injection of various DNA structures. Thus, initially, the robustness of DNA from mammary, erythrocyte and lymphocyte cells of humans was considered. The three DNA-types are significantly larger in comparison to that of *E. coli* plasmids and, therefore, have a larger potential to camouflage the injected malicious code DNA sequences. Therefore, the ratio between the original and malicious DNA codes is lower for human cells, making it harder to identify the injected DNA sequences. This is particularly relevant as our analysis heavily relied on Fourier Spectral Energy values, which tend to overlook smaller fluctuations in signals. Subsequently, results of those analyses were confirmed for *E. coli* plasmids.

When investigating robustness of DNA to code injections, we first obtain string representations of real DNAs for each type that we analyze: DNA of *E. coli* plasmids as well as mammary, erythrocyte and lymphocyte DNAs of humans. Thus, for the human cells we used 254 mammary, 104 lymphocyte and 48 erythrocyte DNAs obtained from individuals, stored and made publicly available by NCBI database (see Appendix B of the Supplementary Material). For each of the cell-types the DNAs included equal numbers of cancerous mutations and their cancer-free counterparts. For each cell types its DNA strings were modelled with injection of malicious code, where sub-sequence of the original DNA were substituted with malicious DNA sub-sequences similar to that proposed in[7]. Each malicious sub-sequence was generated for a particular shell-command designed to hijack control of the DNA-sequencer. Full list of the commands can be found in the Supplementary Material, Appendix C of the article. Each command was first represented by the binary code of its characters. From the binary code the malicious DNA sub-sequence was then derived by using a simple coding technique, where '00' is substituted with 'A', '01' with 'C', '10' with 'T', and '11' with 'G'. The ACTG encodings were inserted into the existing DNAs. In each of the DNA a partition of the same size as the encoding is randomly selected, the partition is substituted with the encoding. All of the original DNA and their injected counterparts were amalgamated into a single DNA-pool.

To establish robustness in identifying malicious code injections of DNAs, Case Based Reasoning (CBR), a renowned Data Mining technique was applied. CBR is a technique that mimics the decision-making process of

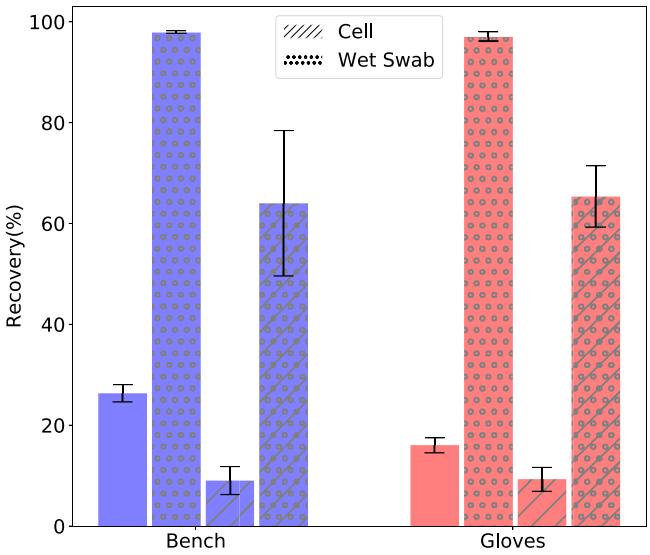


Figure 4. Recovery Rates for Malicious DNA material from non-absorbent surfaces using wet and dry swabbing.

humans such that when an individual needs to make a decision they will first intuitively try to call upon their previous similar experiences[21], which is continually extended depending on the results caused by each decision. In the exact same way, CBR maintains and updates a collection of previous cases that have been put to it. For genetic similarity assessment, we considered two DNA classes: *Original* and *Injected*. For each class and each DNA-type, we randomly selected a subset of original and injected DNAs from the DNA-pool. The subset would serve as CBR's previous experience. Euclidean closeness to one of the previous experience cases was used as the basis for classification. Performance of the CBR classification was evaluated for the available DNAs other than those used as previous experience. While this study is only concerned with suitability of input-control techniques, successful use of any classifier is sufficient. However, some of the specific features of CBR are particularly attractive in the context of classification of an arbitrary DNA. CBR does not have an explicit training stage but solely relies on its previously known cases. Therefore, introduction of additional species will only require the cases to be appended with the species' representative samples. For detailed description of CBR and its use in this study see Appendix E.

Results and Discussion

Following the methodological stages of this research, the two subsections below present results that we obtained while validating *Biohacking as a Valid Threat* and using *Genetic Similarity as a Counter-Measure*.

Biohacking as a valid threat. To prove that *E. coli* may be used during physical transport of malicious DNA, 500 ng of *E. coli* plasmid DNA and approximately 10 million *E. coli* bacteria were inoculated to the three surfaces under investigation (i.e. lab bench, glove and lab coat), as described in the previous section. When using wet-swabs to recover dried plasmid DNA from non-absorbent surfaces, 97% of inoculated material was recoverable (see Fig. 3). Using dry swabs, 26% and 16% of dried plasmid DNA was recovered from the bench and glove sites, respectively. When using wet-swabs to recover dried *E. coli* bacteria from on-absorbent surfaces ~65% of cells were recoverable. Using dry swabs, 9% of the dried *E. coli* population was recovered from the lab bench and glove sites. DNA sequencing of all of the recovered material allowed correct reconstruction of the manipulated DNA. No errors were encountered during sequencing. Neither *E. coli* plasmid DNAs nor *E. coli* bacteria were recoverable from the absorbent surfaces (i.e. labcoat) to quantifiable levels, regardless of dry or wet swabbing. The detection limits using standard quantification for plasmid DNA and bacteria cells were 1 ng/µl or $O.D._{600} = 0.001$ respectively.

While the results in Fig. 4 paint a picture of relative safety offered by absorbent surfaces, lowering the PCR detection level of the swabbing results did yield some DNA material from the injected *E. coli* plasmids and bacteria. Even though, in this case the level of contamination was much lower than what's required for standard quantification, the amount of manipulated DNA allowed for error-free sequencing and was sufficient to contaminate and compromise a DNA sequencer. To prove that we used a series of *E. coli* plasmid dilutions, which were subjected to PCR amplification, as little as 0.1 pg of plasmid DNA as template per 15 µl PCR reaction yielded amplification. As the plasmid used in this study (pEX-A128) is 2,450 bp in size and contains a malicious insert (297 bp), it can be calculated that 0.1 pg of plasmid DNA contains 3.5×10^4 plasmid molecules. Using a conservative estimate of 200 plasmid molecules per cell, only 1,750 cells from the 10 million inoculated on surfaces are needed to yield amplifiable amounts of the malicious DNA. Therefore, it can be concluded that both *E. coli* plasmid DNA and *E. coli* bacteria possess sufficient capabilities to contaminate and compromise DNA sequencing equipment regardless of the surface type or swabbing method. The persistence, and therefore, stability of cells as a malicious DNA carrier/source of infection would be further augmented if spore-forming cells were used[22].

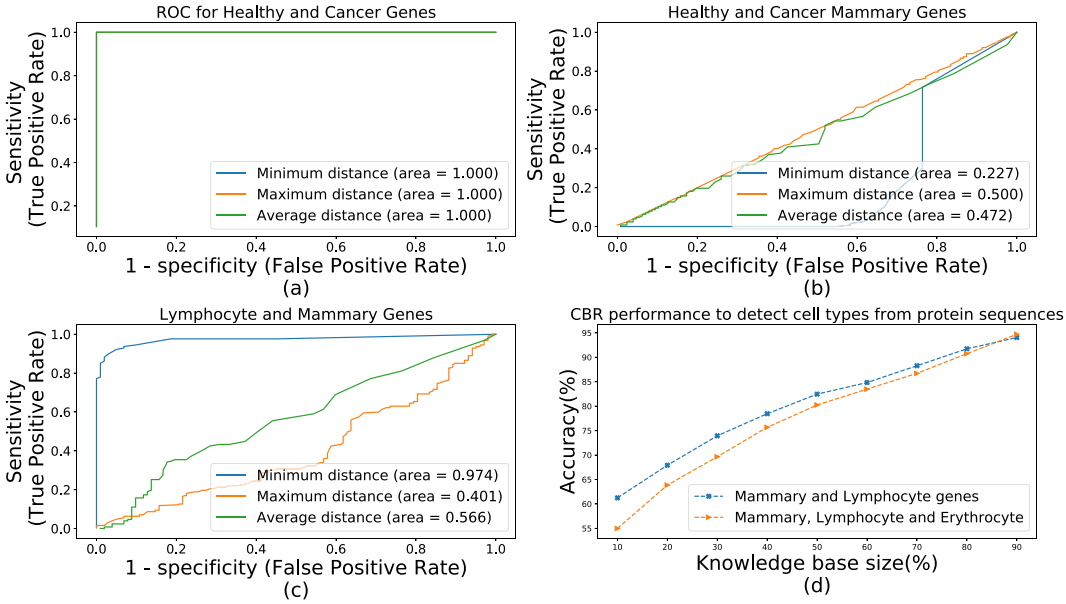


Figure 5. DNA similarity, extended study ROC curve for threshold-classification between (**a**) Healthy/Cancer samples from[20], (**b**) Healthy/Cancerous mammary and (**c**) lymphocyte/mammary samples from NCBI; (**d**) CBR cell-type classification.

Genetic similarity as a counter-measure. Results of previous studies have shown the use of Fourier Spectrum Energy representation of Voss Transformation for DNA classification. Thus[20], reports on a particular arrangement for DNA sequences of Healthy and Cancerous samples of human mammary tissue (see *Appendix D* Fig. 9), where all Healthy DNAs belong to a single tight cluster that all Cancerous DNAs lay strictly outside of. However, as results of[20] are based on a small selection of DNAs, conclusions of[20] require further validation. Figure 5(a–c) evaluate (via ROC-analysis) threshold-based separation between Healthy and Cancerous DNA using Average, Minimal or Maximal distance to the remainder of Cancerous/Healthy samples. The arrangement reported in[20] results in exceptional ROC curves with AUC equal 1 (Fig. 5(a)). This does not quite hold for a larger set of DNAs obtained from the NCBI database (Fig. 5(b,c)). Both figures noticeably differ from the results of[20] in cases of Average and Maximal distances. Both distances express a fairly low predictive capacity (AUC close to 0.5 due to chance), which contradicts one cluster arrangement of Healthy mammary DNAs. However, good predictive capacity of the Minimal Distance (i.e. AUC close to 1 or 0) is an indication of some cluster-like structure amongst the DNAs. This was further confirmed by sufficiently high CBR-classification accuracy obtained for the NCBI's DNA samples (Fig. 5(d)).

The structure re-established by Fig. 5 further re-affirms the validity of CBR for identification of DNAs with malicious code injections. As cancerous mutations (which typically affect very small DNA-partitions) are sufficient to distinguish Cancerous from Healthy DNAs, much larger code injections should be detectable using the same methods. To confirm that, we first conducted a series of experiments trying to determine if the original malicious code proposed in[7] could be detected via CBR. Figure 6 presents results for the malicious code injections in mammary, lymphocyte and erythrocyte DNAs. Similar to other figures, Fig. 6(a,b) evaluates the predictive capacity of the three distances. This evaluation closely aligns with the conclusions from Fig. 5. Subsequently, Fig. 6(c) shows results of the CBR classification showing the levels of correct detection increasing up to approximately 90%. While this detection rate may seem low, it is anticipated that higher detection rate will be achieved in a practical setting. The presented results were obtained for various sizes of previously known genetic material exploited by the CBR. To protect a real DNA sequencer, the most complete CBR knowledge (close to 100%) will be used. In our experiments up to 90%-complete CBR knowledge was used, as a portion of known DNAs were required to evaluate the detection itself.

Figures 5(d) and 6(c) provide additional insight in relation to the impact of the set of biological species within which the malicious DNA is detected. The CBR-classification is better for the lesser specie-sets (i.e. *"Mammary and Lymphocyte"* set in Fig. 5(d) and *"Lymphocyte"* set in Fig. 6(c)) when compared to their extended versions (i.e. *"Mammary, Lymphocyte and Erythrocyte"* set on both figures). Due to the increased DNA-variability of the extended sets CBR knowledge of a particular size is likely to capture more of the DNA-variability of a lesser specie-set. At the same time, for any specie-set, larger CBR knowledge captures more of the DNA-variability and thus shows better detection accuracy. This can be also seen in Figs 5(d) and 6(c) where all of the detection accuracies increase with increase of knowledge size. However, as the accuracy-growth is naturally bounded, the difference in detection accuracy between lesser and extended specie-sets diminishes with the increase of CBR knowledge size. This will impact on the practical use of the proposed CBR-based detection of malicious DNAs. To ensure adequate detection-accuracy, applications dealing with heterogeneous DNA-sources will require larger CBR knowledge compared to their specialized counter-parts (e.g. Human Lymphocyte-only samples).

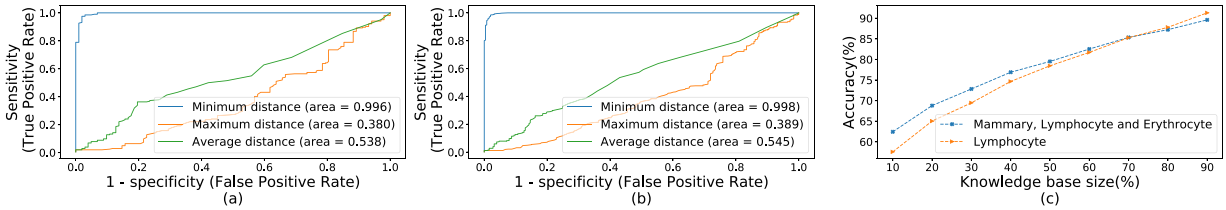


Figure 6. DNA similarity, injection of malicious code from[7]: ROC curves for threshold-classification of injected DNA for (**a**) lymphocyte and (**b**) mammary, lymphocyte and erythrocyte cells; (**c**) CBR classification.

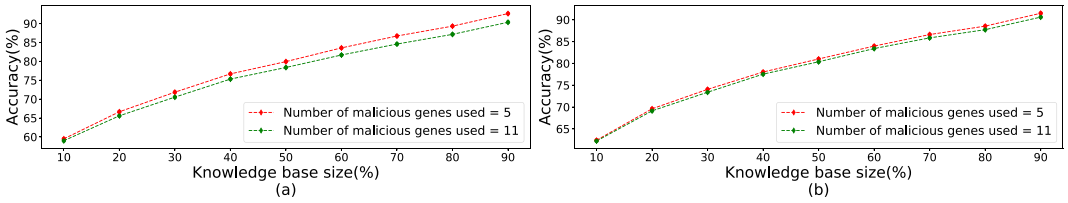


Figure 7. CBR-based detection of diverse malicious DNA in (**a**) lymphocyte and (**b**) lymphocyte, mammary and erythrocyte cells.

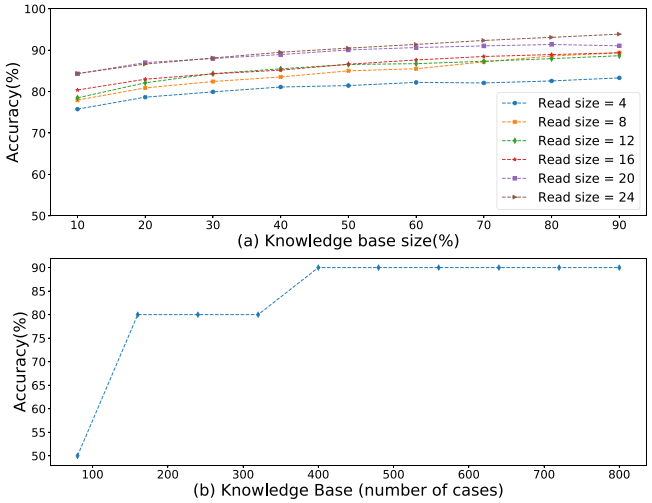


Figure 8. CBR-based detection of malicious content: (**a**) in DNA fragments of human mammary, erythrocyte and lymphocyte DNAs; (**b**) in *E. Coli* Plasmid DNA.

Figure 7 summarizes the impact of variability in the malicious code on the detection accuracy. 5 to 11 (see Supplementary Materials) different malicious code samples were injected into DNAs of lymphocyte, erythrocyte and mammary cells. The shape and values on Fig. 7 are noticeably close to those on Fig. 6, which leads us to believe that code variability had little or no impact on the CBR performance. This supports the hypothesis originated from the analysis of Fig. 5. Malicious code exceeds cancerous mutations in size and should be detectable by CBR.

Figure 8(a) investigates the use of CBR for injection classification based on DNA fragments that would be typically formed at the end of the first stage of DNA sequencing. Figure 8(a) looks at how accurate a single DNA-fragment can be identified as coming from a genuine or injected DNA. Note that depending on the DNA sequencer and associated chemistry (e.g., Sanger versus Illumina sequencing or amplicon versus genome sequencing), its user can specify or predict the size of DNA fragments (also called reads, amplicons or fragments to be sequenced) that the original DNA will be assembled into the complete sequence[23]. From the information-security viewpoint, detecting malicious code injections from fragments presents a stronger counter-measure for DNA hacking. Detecting a malicious injection from fragments will preclude any further analysis of the fragments, guarding DNA assembly code from its own binary vulnerabilities. However, the quality of such detection will depend on the size of the DNA fragments. On one hand, for smaller fragments the length ratio between the original and injected components is expected to be higher (compared to larger fragments or full DNA), potentially simplifying the detection. On the other hand, from the perspective of combinatorics, we know that the numbers of all possible "ACTG" string-representations is fewer for shorter fragments. This complicates the detection. This

is confirmed in Fig. 8(a), where general classification accuracy increases as the fragment length (or read size) increases. Yet the increase is bounded. Even for fragments of 4 bp, the detection accuracy is higher than that of a full DNA (Figs 6–8). Therefore, the optimal read size is limited, but should not be below 24 bp.

Finally, we investigated the use of CBR to detect malicious code injections into the DNA plasmid material of *E. coli* bacteria. Figure 8(b) shows the growth of the detection accuracy with the increase of the knowledge size. As it can be seen from the figure, the growth is more rapid for *E. coli* rather than humans. This can be explained by the lower complexity of *E. coli* DNA, leading to a lesser representation required to describe the underlying structure of genuine and injected DNAs.

Conclusions

In this article we have investigated the threat of bio-hacking for modern DNA-sequencers. We have shown that simple organisms such as *E. coli* bacteria can be used to transport malicious DNA code to its destination. To protect against this threat, we have proposed a counter measure. We have shown that it is possible to use the existing DNA-similarity of biological species to identify the presence of malicious code within DNA of a particular sample. With the example of lymphocyte, erythrocyte and mammary DNA of humans, we have shown how Voss Transformation and CBR can be used for that identification. The accuracy of the identification increases as more DNA structure information becomes available to the CBR model. Code injections appear to be substantially different to natural mutations of DNA, and variability of the malicious code has lower impact on its identification accuracy. It is beneficial to use DNA fragments to increase the accuracy of identification, where there is an optimal fragment-length to be used. We have also demonstrated a new form of transport for the DNA with encoded malicious code, and that is through various types of materials (e.g., lab coat, glove, or lab bench). This demonstrates that hackers in the future can transport the DNA with the encoded malicious code and swab the samples once they get close to the DNA sequencer, or spread the samples in an environment that will be potentially be swabbed. Experiments have shown that recovery was achievable for materials such as glove and lab bench, but was not very good for cloth.

Data Availability

All data used during the study presented in the manuscript is freely available in the public domain. The manuscript's Methodology (e.g. Section 2) outlines the origin of the data. Furthermore, for the convenience of the reader we provide supplementary material that (Appendixes B and C) lists the DNAs and injection codes used.

References

1. Sun, W. *et al.* Common Genetic Polymorphisms Influence Blood Biomarker Measurements in COPD. *PLOS Genetics* **12**(8), 1–33 (2016).
2. Varshney, R. K. *et al.* Analytical and Decision Support Tools for Genomics-Assisted Breeding. *Trends in Plant Science, Elsevier* **21**(4), 354–363 (2016).
3. Wallén, S. E., Lillehammer, M. & Meuwissen, T. H. E. Strategies for implementing genomic selection for feed efficiency in dairy cattle breeding schemes. *Journal of Dairy Science, Elsevier* **100**(8), 6327–6336 (2017).
4. Paoletti, D. R., Krane, D. E., Raymer, M. L. & Doom, T. E. Inferring the Number of Contributors to Mixed DNA Profiles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.* **9**(1), 113–122 (2012).
5. Jain, S., Farnoud Hassanzadeh, F., Schwartz, M. & Bruck, J. Duplication-Correcting Codes for Data Storage in the DNA of Living Organisms. *IEEE Transactions on Information Theory* **63**(8), 4996–5010 (2017).
6. MinION DNA-sequencer, accessed online on 15.10.2018 at, https://nanoporetech.com/products/minion.
7. Ney, P., Koscher, K., Organick, L., Ceze, L. & Kohno, T. Computer security, privacy, and DNA sequencing: Compromising computers with synthesized DNA, privacy leaks, and more, Proceeding of the 26th USENIX Security Symposium (USENIX Security 17), USENIX Association, Vancouver, BC, pp. 765–779 (2017).
8. Ron, A., Shulman-Peleg, A. & Puzanov, A. Analysis and Mitigation of NoSQL Injections. *IEEE Security & Privacy* **14**(2), 30–39 (2016).
9. Tsoutsos, N. G. & Maniatakos, M. Anatomy of Memory Corruption Attacks and Mitigations in Embedded Systems, *in IEEE Embedded Systems Letters*, **10**(3), 95–98 (Sept, 2018).
10. Jin, H., Liu, B., Du, Y. & Zou, D. BoundShield: Comprehensive Mitigation for Memory Disclosure Attacks via Secret Region Isolation. *IEEE Access* **6**, 36341–36353 (2018).
11. Ahn, D. & Lee. G. A Memory-Access Validation Scheme against Payload Injection Attacks, *IEEE Transactions on Dependable and Secure Computing*, **12**(4), 387–399 (1 July-Aug. 2015).
12. Yu, J. J. Q., Hou, Y. & Li, V. O. K. Online False Data Injection Attack Detection With Wavelet Transform and Deep Neural Networks. *IEEE Transactions on Industrial Informatics* **14**(7), 3271–3280 (2018).
13. Liu, X., Li, Z., Liu, X. & Li, Z. Masking Transmission Line Outages via False Data Injection Attacks. *IEEE Transactions on Information Forensics and Security* **11**(7), 1592–1602 (2016).
14. Motahari, A. S., Bresler, G. & Tse, D. N. C. Information Theory of DNA Shotgun Sequencing. *IEEE Transactions on Information Theory* **59**(10), 6273–6289 (2013).
15. Nguyen, H. H. *et al.* Long-Term Stability and Integrity of Plasmid-Based DNA Data Storage. *Polymers* **10**(28), 1–10 (2018).
16. Fan, L. *et al.* Development of a screening system for DNA damage and repair of potential carcinogens based on dual luciferase assay in human HepG2 cell. *Mutagenesis, Oxford University Press* **28**(5), 515–524 (2013).
17. Pfeiffer, F. *et al.* Systematic Evaluation of error rates and causes in sort samples in next-generation sequencing, Scientific Reports. *Nature* **8**(1), 1–14 (2018).
18. Meng, T. *et al.* Wavelet Analysis in Current Cancer Genome Research: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **10**(6), 1442–1459 (2013).
19. Mendizabal-Ruiz, G., Román-Godínez, I., Torres-Ramos, S., Salido-Ruiz, R. A. & Morales, J. A. On DNA numerical representations for genomic similarity computation, *Plos One*, **12**(3) (2017).
20. Ghosh, A. & Barman, S. Application of Euclidean distance measurement and principal component analysis for gene identification. *Gene, Elsevier* **583**(2), 112–120 (2016).
21. Chen, Q. *et al.* An ontology-driven, case-based clinical decision support model for removable partial denture design. *Scientific Reports, Nature* **6**(1), 1–8 (2016).
22. Willerslev, E. *et al.* Long-term persistence of bacterial DNA. *Current Biology, Cell Press* **14**(1), 9–10 (2004).
23. Schuster, S. C. Next-generation sequencing transforms today's biology. *Nature Methods* **5**(1), 16–18 (2008).

Acknowledgements

This work is supported by the Finnish Academy Research Fellow programme [284531], as well as Science Foundation Ireland (SFI) via the Precision Dairy project [13/IA/1977], VistaMilk [16/RC/3835] and CONNECT [13/RC/2077] research centres.

Author Contributions

The following description outlines in detail contribution made by the individual authors of the manuscript. *Mr. Mohd Siblee Islam* is the primary author of the article. Mr. Islam specializes in CBR analysis and Information Security. He is the primary author, and he directly contributed to at stages of the research presented in the manuscript. Mr. Islam was the researcher who carried out the CBR-based DNA-analysis presented in this paper. Mr. Islam was responsible for collating the results from wetlab and CBR experiments, and writing the paper. Mr. Islam is the primary author of the paper, he was the primary writer of the content of the article. *Dr. Stepan Ivanov* directed the CBR-based analysis conducted by Mr. Islam. As part of these activities, Dr. Ivanov assisted Mr. Islam in selection of the human DNAs to be used as part of the CBR-analysis, selection of the CBR-experiments that were carried out by Mr. Islam. Input of Dr. Ivanov was vital in understanding the potential cluster-like structure of the DNA-material (e.g. similar to [20]) and linking it to the performance of the CBR method. Dr. Ivanov took part in collating results from wet-lab and CBR-analysis. Dr. Ivanov took part in writing the text of the manuscript (Sections 2.2, 3.2). *Mr. Eric Robson* specializes in the Area of Data Mining and Artificial Intelligence. Mr. Robson took part in understanding the cluster-like structure of the DNA-material. It was Mr. Robson's proposition to use threshold-based analysis and Area Under the Curve metric to explore the geometry of the feature space selected to represent the DNA sequences. The suggestion and subsequent involvement and guidance from Mr. Robson proved crucial in assessing the impact of different distance-based measurements (e.g. minimum distance to the set of cancerous DNAs) on the cluster-like structure of DNA data. *Ms. Trion Dooley-Cullinane* and *Dr. Lee Coffey* specialize in bio-chemistry and were the team who conducted the wetlab and PCR experiments as part of "Relevance of Biohacking" stage of the analysis presented in the manuscript. Ms. Dooley-Cullinane and Dr. Coffey wrote sections 2.1 and 3.1 of the manuscript. *Mr. Kevin Doolin* has a background in the Architecture of Computer Systems. Mr. Doolin's input was vital in understanding the practical use of the proposed detection technique from the computer systems perspective. It is Mr. Doolin who drew a parallel between the attack and the injection techniques that are commonly used against modern ICT systems. This understanding and further direction and guidance from Mr. Doolin helped identify input control as the approach to counter the attack. Subsequently, Mr. Doolin identified the use of the technique as part of a system (e.g. detection from DNA fragments rather than complete sequence). *Dr. Sasitharan Balasubramaniam* was the mastermind and the main scientific driver behind the experiments presented in the article. Due to his multi-disciplinary background, Dr. Balasubramaniam identified the possibility for *E. Coli* bacteria to be used as carriers of malicious DNA on-purposed engineered as part of an attack. In essence that was the starting point for the research presented in the article. Subsequently, Dr. Balasubramaniam directed and oversaw both wetlab and DNA experiments conducted in the presented research.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-019-44995-6.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix C

Trojan Bio-Hacking of DNA-Sequencing Pipeline

Conference Title	6^{th} ACM Conference on Nanoscale Computing and Communication (ACM NanoCom)
Article Type	Regular
Complete Author List	Mohd Siblee Islam, Stepan Ivanov, Kevin Doolin, Lee Coffey, Triona Marie Dooley-Cullinane, Donagh Berry and Sasitharan Balasubramaniam

Trojan Bio-Hacking of DNA-Sequencing Pipeline

M.S. Islam, S. Ivanov, K. Doolin, L. Coffey, T.M. Dooley-Cullinane, D. Berry, S. Balasubramaniam

ABSTRACT

The article focuses on the information security risks that arise from the use of dubious software as part of a DNA-sequencing pipeline. We show how the perpetrator can use a biologically engineered sample that contains the remote machine's IP address and port number to trigger Trojan spyware previously dormant, and create a connection to the remote machine. The spyware is then used to either steal sensitive data processed by the pipeline (e.g. DNA-sample of crime suspect) or manipulate its control-flow (e.g. via opening a backdoor). To avoid detection the spyware can accept and expect required payload in fragments, which are also hidden inside the sample in a distributed manner. We show how the adversary can use cryptographic tools such as encryption and steganography to make such detection even harder while limiting the footprint that either identifies the attacker or makes the trigger-sample substantially different from its biological species. Therefore, we prove the viability of the attack and further stress the need to account for attacks being launched from the physical, rather than cyber-world. Furthermore, DNA sequencing error can hinder the successful delivery of a payload, hence the success of such attacks. We estimate the success rates for different sequencing error rates, where the calculated results are also verified with corresponding results from simulations.

CCS CONCEPTS

• **Security and privacy** → *Cryptanalysis and other attacks*; *Domain-specific security and privacy architectures*; Information flow control.

KEYWORDS

Bio-Hacking, DNA-Sequencing Pipeline, Steganography, Encryption

ACM Reference Format:

M.S. Islam, S. Ivanov, K. Doolin, L. Coffey, T.M. Dooley-Cullinane, D. Berry, S. Balasubramaniam. 2019. Trojan Bio-Hacking of DNA-Sequencing Pipeline. In *The Sixth Annual ACM International Conference on Nanoscale Computing and Communication (NANOCOM '19), September 25–27, 2019, Dublin, Ireland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3345312.3345474

1 INTRODUCTION

Nowadays one can hardly overstate the importance of DNA-analysis in our society. Once a matter from the domains of science-fiction and blue-sky research, today applications of DNA-analysis include disease diagnostics [9], animal [15] and plant breeding [14], criminal forensics [7] and even data-storage [3]. Meanwhile, this rather substantial application range is set to increase even further. The increase is mainly attributed to the improved accessibility of DNA-sequencing due to the arrival of miniature DNA-Sequencing technologies such as MinION [11]. The miniature size and moderate costs of the MinION device allow for the DNA-sequencing to be performed by any individual and in any location. This is a significant improvement of the conventional methodology where sequencing of a biological sample has to be done by a trusted third-party on their premises.

As DNA-analysis is used more and more and becoming commercially available, cyber-security issues of the typical software used as part of the analysis come to light. Thus, a recent study [13] uncovers a binary vulnerability of the typical DNA-sequencing pipeline, an essential part of the DNA-sequencing process. The study shows an attack where a biological sample is genetically engineered in such a way that its sequencing allows the perpetrator to hijack the control-flow of the pipeline's operation, and subsequently either steal confidential DNA-data and/or alter results of the DNA-sequencing. To achieve this the DNA sequence of the engineered sample includes a malicious payload, which is a computer script that opens a reverse shell in a form of a backdoor for the perpetrator to gain control of the pipeline. Depending on the context, the attack can have severe consequences, such as manipulation of disease diagnosis or using false DNA-forensics in criminal investigations. Memory corruption, the underlying issue that allows execution of the payload for the attack, is a well-known and studied software vulnerability that can be addressed by a number of existing techniques. For example, [4] describes the use of hardware strengthening of memory access (i.e., Intel MPX technology) [1], which utilizes a technique that operates at the Operating System level. Routine inclusion of these techniques in software/hardware configuration of DNA-sequencing pipeline will diminish the threat posed by the attack. The attack demonstrated in [13] demonstrates the first kind of **bio-cybernetic** attack.

In this article we propose an alternative bio-cybernetic attack on the DNA sequencing pipeline. The attack is executed by a dormant Trojan spyware installed by a user unaware of its intended purpose. The spyware may be supplied together or as part of a wider bio-informatics software package incorporating a variety of bio-informatics tools, some of which may be genuine. Fooled by the appearance of the toolbox with its genuine components, as well as targeted social engineering efforts (e.g., positive recommendations from colleagues and other trusted members of the society), the user installs the entire package including the spyware. Immediately after installation the spyware remains dormant for a period of time, and this allows the spyware to (1) avoid security counter measures such as sandboxing, and (2) continue building user-confidence in the toolbox up to the point of the spyware activation. The trigger for the spyware, however, will come remote attacker's address (port and IP address) that is encoded into a DNA. Once the DNA is sequenced,

NANOCOM '19, September 25–27, 2019, Dublin, Ireland

ACM ISBN 978-1-4503-6897-1/19/09…$15.00
https://doi.org/10.1145/3345312.3345474

the spyware will search for the encoded address in the DNA and create a remote connection to the attacker's computer. The uniqueness of our proposed attack is the hybrid combination of the spyware as well as a genetically engineered DNA trigger-sample.

The paper is organized as follows. Section 2 describe the envisaged scenario of bio-cybernetic attack. Section 3 describes the technique for generating of the trigger sample for the attack. In Section 4 and Section 5 our experiments and results are discussed. Finally the paper is concluded in Section 6.

2 THE ATTACK SCENARIO

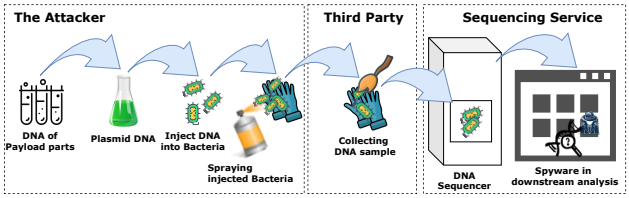


Figure 1: Spyware activation; Synthesis, Delivery and Sequencing of the Trigger Sample

In this section, we present our view of how a Trojan based bio-cybernetic attack on a DNA sequencing pipeline can be carried out. As described in the Introduction, the Trojan spyware is already embedded into the pipeline, and remains dormant while awaiting for the corresponding trigger which will come through the DNA samples. We skip the details of how this can be achieved, as it is anticipated that the attacker will use some of the already existing techniques proven for the conventional spyware. Instead, we deliberately focus on the novel hybrid bio-cybernetic component of the attack, namely spyware activation.

To activate the dormant spyware, the attacker will execute a number of steps presented in Fig. 1. The attacker begins by engineering the trigger-sample. The sample presents a genetic modification of an existing baseline specie. While the attacker may choose the baseline specie from a wide range of organisms, in this article we consider the use of *E.Coli* bacteria. A common contaminant frequently found in various environments is the *E.Coli* bacteria and it is used extensively in DNA-based synthetic biology experiments. Easily manipulated and then sequenced, the *E.Coli* are frequently used in range of applications including DNA data storage [5] and bacterial communications [12]. Thus, it is with relative ease that the attacker may engineer an *E.Coli* bacterial sample that would include a particular payload, the information necessary for the Trojan's activation and its subsequent operation. The information may vary depending on the purpose and functionality of the Trojan. In Fig. 1 the attacker inputs the payload into the *E.Coli* plasmid DNA (as described in Section 3) that is then inserted via ligation into the *E.Coli* bacteria themselves. In this article we consider a spyware that creates a connection to a remote machine that belongs to the hacker. This communication will require a network address and a port number to communicate the data. Naturally, for this scenario the payload will comprise of the address and the port number.

Once the trigger sample is generated the attacker needs to deliver it to the intended target, a specialist DNA sequencing facility, laboratory, medical practice or other. During the analysis of the sample, the payload will be extracted and passed on to the spyware. To deliver the trigger sample, we envision that the attacker would spray it at various locations that would be then swabbed by a third-party and subsequently sent for analysis. With *E.Coli* being a common and well known contamination agent, it is anticipated that presence of the bacteria is unlikely to be treated as suspicious by either the third-party of the sequencing service. This will mask the attack, while the use of a third-party will protect the attacker's physical identity.

3 TRIGGER SAMPLE: PAYLOAD INJECTION

With the attacker's physical identity shielded by the third-party, the payload information injected into the DNA of the trigger sample will become the only forensic link to the attacker. Therefore, to avoid possible repercussions it is in the attacker's best interest to inject the payload in a way that significantly complicates its discovery by anybody apart from the attacker or the Trojan spyware. This motivates us to utilize a technique known as *Steganography*, which allows a user to hide (i.e. inject to prevent unintended discovery) information in a wide variety of data. For example, [10] describes a technique to hide text messages inside other non-related text-data. Over the years, a number of techniques have been proposed to hide information in DNA data [8], [6], and in this article we build on top of the existing DNA-steganography techniques to ensure secrecy of the payload injection.

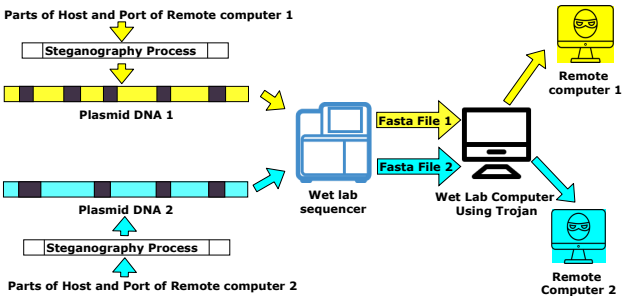


Figure 2: Payload injection by means of DNA-steganography

Specifically, we adapt the state-of-the art DNA-steganography technique developed in [6] to inject the payload into the plasmid DNA of *E.Coli* bacteria. Given a generic text, the technique transforms it into a sequence of 'ACTG' nucleotides that is then inserted into a DNA plamid, also known as the carrier DNA. The transformation consists of two phases, where first the text is padded with specialist symbols (in this article we use '#' and '_') marking the beginning and the end of the payload. The padded text is then represented as binary version of its ASCII code. Each binary pair from the code is then mapped to 'ACTG' nucleotides, with '00', '01', '10' and '11' being replaced by 'A', 'C', 'G' and 'T', correspondingly. As this encoding scheme is rather common (e.g. used in [13]) and, hence, may be easily uncovered, [6] may also include an element of encryption as the second phase of the DNA-steganography technique. We provide a more detailed scenario-based description of the operation below. However, while the two steps of the technique provide an 'ACTG'

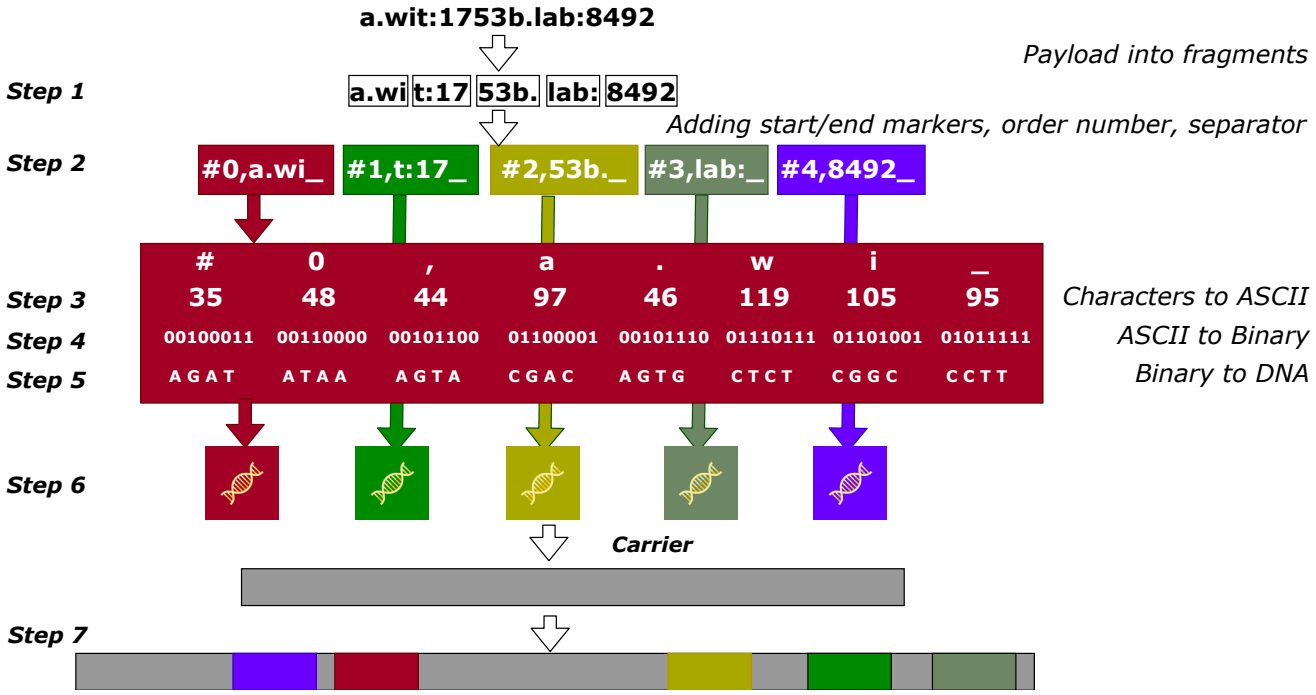


Figure 3: Fragment-Based payload injection: first phase

sequence that is cryptographically protected, the sequence may not follow structural patterns typical to the carrier DNA's structure. Large payloads may completely change the DNA-appearance of the trigger sample, making it unnatural and suspicious. To avoid this, we propose splitting the payload into fragments and insert into the plasmid DNA, as illustrated in Fig. 2.

3.1 Fragment-Based Payload Injection

To illustrate the mechanics of the proposed algorithm, consider an attack where the attacker intends to make a remote connection using networking addresses **a.wit** and **b.lab** via ports 1753 and 8492, respectively. The attacker needs to engineer an *E.Coli* plasmid DNA with **a.wit:1753b.lab:8492** inserted into the sequence. Fig. 3 and the description below presents the steps that the attacker takes to produce the plasmid with the insert payload as the trigger sample.

Step 1: Payload will be divided into fragments of size $Size_{fr}$. In Fig. 3, $Size_{fr}$ is 4.
Step 2: Each fragment is appended with its sequence number (used by the spyware to re-assemble a payload from the fragments) that is separated from the fragment's text by ','. The result is padded with the start ('#') and the end ('_') markers.
Step 3: Each padded fragment is represented as sequence of ASCII values of its symbols.
Step 4: Each integer value for the ASCII is converted into binary.
Step 5: Binary value are encoded into nucleotide bases, where '00', '01', '10' and '11' are converted into A', 'C', 'G' and 'T', respectively.
Step 6: Steps 2 to Step 5 are repeated for each of the fragments.
Step 7: 'ACTG' encoding for each of the fragments is inserted into a carrier DNA.

As part of the second phase, Step 4 can be extended with a two-key encryption process. In this case, a symmetric primary key ($key1$) is used for encryption/decryption of the padded fragments, while a secondary key ($key2$) is used to introduce additional bits and thus hide the $key1$-encrypted fragments. Fig. 4 depicts the additional steps and their descriptions is as follows:

Step 4a: XOR operation is performed first between the binary value of $key1$ and the binary of the least significant character of the fragment. In Fig. 4, the $key1$ is 60. The output binary is used for next the XOR operation with the binary value of the next least significant character. The process continues until all characters are encoded.
Step 4b: A cover DNA string is considered next, and it can be any arbitrary DNA string of a sufficient length. The number of cover bits is determined by the secondary key ($key2$). In Fig. 4, $key2$ is 3, i.e. one bit from Step 4a is going to be covered by 3 bit from the Cover DNA.
Step 4c: Cover DNA gets converted into binary format.
Step 4d: Bits from the cover DNA are embedded with bits from Step 4a. The output is then used as the input for Step 5 that is described above.

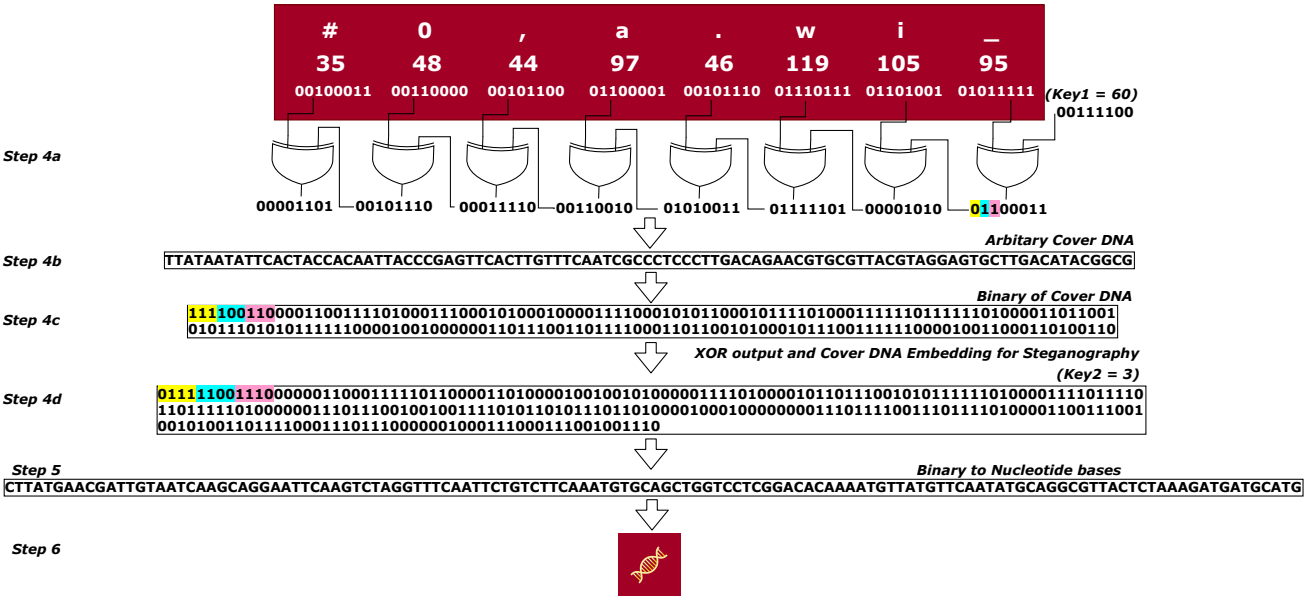


Figure 4: Fragment-Based payload injection: second phase

3.2 Payload Inflation due to Fragment Padding

While payload fragmentation allows the attacker to avoid prolonged sequence injections into the carrier DNA, and thus makes the injected DNA appear more natural, fragmentation also results in negative effect. As each fragment is padded individually with its sequence number and a separator symbol, fragmentation increases the overall length of the payload that needs to be injected into the carrier DNA. This is because the additional padding will result in the overall length of all fragments exceeding the length of the payload. Thus, consider a payload of $Size_p$ characters divided in fragments of length not greater than $Size_{fr}$. In this case the number of fragments generated will be:

$$N_{fr} = ceil(Size_p / Size_{fr}), \tag{1}$$

where *ceil* is the function that returns the least integer value exceeding the argument. As each of the fragments is required to be padded with the fragment start and end markers, its sequence number, and the separator symbol, the combined length of all fragments equates to:

$$N_{ch} = Size_p + N_{fr} \cdot (S_{stm} + S_{edm} + S_{sep} + S_{o}*), \tag{2}$$

where S_{stm}, S_{edm} denotes the length of the fragment start and end markers; $S_{o}*$ and S_{sep} denotes the character-length of the fragment sequence number and its separator. As the length of each ASCII-character is 1-byte (8 bits), its nucleotide encoding will require 4 nucleotide bases (each symbol being either A', 'C', 'G' and 'T' encoding 2 bits), leading to the overall length of injected-DNA being:

$$N_p = 4 \cdot N_{ch}, \tag{3}$$

which, if the second phase is also used, will be extended to:

$$N'_p = (8 \cdot N_{ch} + 8 \cdot N_{ch} \cdot key2)/2. \tag{4}$$

The (1)-(4) helps us gain a numerical insight into the payload inflation. Thus, using fragments of smaller size $Size_{fr}$ will increase their number N_{fr} (see (1)), which, due to padding (second term in (2)), will require larger number of nucleotides for the encoding regardless if the first (see (3)) or both phases (see (4)) are used as part of the payload injection.

3.3 Sequencing Errors and Payload Retrieval

While DNA-sequencing has become rather advanced, and modern techniques typically provide stable and accurate detection, errors in DNA-sequencing may still occur. Poor and contaminated samples may result in low-quality output of the DNA-Sequencing Instrument leading to Incorrect Base Calling (i.e. errors in nucleotide structure of the DNA). Therefore, there is an inherent probability $prob_e$ for each nucleotide of the trigger sample to be detected incorrectly during the sequencing process. If these errors are found in the payload, will result in profound effects on the performance of the Trojan spyware, which will either fail to recognise particular fragments (e.g. due to fragment start marker missing) or read the network address/port incorrectly. Meanwhile, given $prob_e$, the probability $prob_r$ of correct retrieval of payload size N_p can be calculated using the formula for bit error rate calculation in data communication, and is represented as follows:

$$prob_r = (1 - prob_e)^{N_p} \tag{5}$$

In the case of steganography, the nucleotides appended for the $key2$ should be ignored, except for the least significant nucleotide as it is related to the least significant bit of the payload. To further clarify this point, let us assume that our payload contains a nucleotide *'C'*, which is **01**, and let's consider the $key2$ is 3 and the arbitrary bits

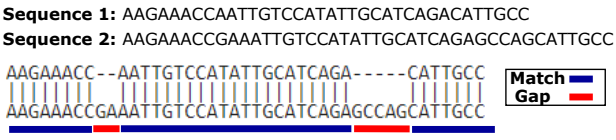


Figure 5: Match and Gap to compute NW_{score}

using $key2$ are *101* and *111*. This will result in the values *101***0** and *111***1**, after applying steganography. From these *101***0** and *111***1**, **0** and **1** is important and this will create two nucleotides *'T'* and *'G'*. Therefore, in the case of encryption, one nucleotide directly related to payload will be split into two nucleotides. However, the error in that nucleotide will not directly affect the retrieval as we can see that the least significant bit is important in **0** and **1**. Therefore, if *'T'* is read with an error as *'A'*, then the least significant bit will be same but will be different for *'C'* and *'G'*. This means that the mutation favouring the retrieval is 1/3 and the remaining 2/3 is only important for the calculation. So (5) can be modified to the following:

$$prob'_r = (1 - prob_e)^{\frac{2}{3}x}, \text{Where } x = \frac{2 \cdot N_p}{key2 + 1}. \quad (6)$$

Similar to the previous section, (5) and (6) helps us gain further insight into the effect of fragment padding. The increased size of all of the fragments combined (i.e. N_p or N'_p) will have a decremental effect on the payload retrieval probabilities (i.e. $prob_r$ and $prob'_r$), thus, reducing chances for the attack to be carried out successfully.

4 EVALUATION: METHODOLOGY AND PARAMETERS

As has been mentioned previously, this article mainly concentrates on secretly insert payload into plasmid DNA of *E.Coli* bacteria so a trigger sample can be created that will be captured by a spyware, but will not be easily detected by any other security algorithm. The sample needs to resemble the natural *E.Coli* plasmid DNA as much as possible in order to avoid detection. When it comes to measuring similarity of DNA, Bioinformatics offers a number of solutions, and the **Needleman Wunsch (NW)** global alignment score (NW_{score}) is by far the most popular. The score is included in a wide range of various bioinformatics tools boxes, including BLAST, which is developed and supported by NCBI[2]. To obtain the score, two DNA sequences are aligned as shown in Fig. 5, then the weighted values of the length of matching and miss-matching intervals of the two sequences, as well as length of the gaps is added to form the score. In this article. we use the NW score to evaluate the secrecy of the payload insertion into the carrier DNA. As the global alignment score NW_{score} for any DNA with itself is a non-zero number (i.e. weighted length of the DNA due to perfect matching), we strive to minimise the change in the NW_{score} between the carrier DNA and its injected version compared to the self-NW_{score} of the carrier DNA. Our aim is to evaluate the minimum change can be for various sizes of *payloads*, *fragments*, *secondary key*, and other parameters of the proposed algorithm. Additionally, we evaluate the impact of those parameters of the payload retrieval probability for various basecall error probabilities. The parameters that we consider in this article are as follows:

Parameter Name	Values
Carrier Size	100, 200, 300, 400, 500
Payload Size	10, 20, 30, 40, 50
Fragment Sizes	2, 4, 6, 8, 10
$key2$	3, 5, 7, 9, 11
$prob_e$	0.0025, 0.005, 0.0075, 0.01

For each scenario (a scenario consists of a combination of above mentioned parameters), we keep the $prob_e$ static and execute 100 simulation runs using randomly generated *DNA carrier sequences* and randomly generated payload as well as insertion of *payload* fragments during each execution in order to compare the result with the calculated $prob_r$. After mutating random nucleotides of the *Target DNA Sequence* considering a static $prob_e$, we try to retrieve the payload using the Trojan spyware. The retrieval probability is the percentage of successful retrievals from the 100 simulations. The retrieval becomes successful if the payload is not affected by the $prob_e$. Results obtained for the simulated $prob_r$ are then compared to $prob_r$ values calculated using 20 real DNA sequences as carriers. All are plasmid DNA sequences and the number of nucleotide bases are 7181, 3781, 8464, 3931, 3956, 3851, 5764, 6106, 5934, 5200, 3015, 6134, 5822, 3504, 3669, 3829, 3725, 10668, and 6706. This will result in 3000 total number of scenarios.

5 RESULTS AND DISCUSSION

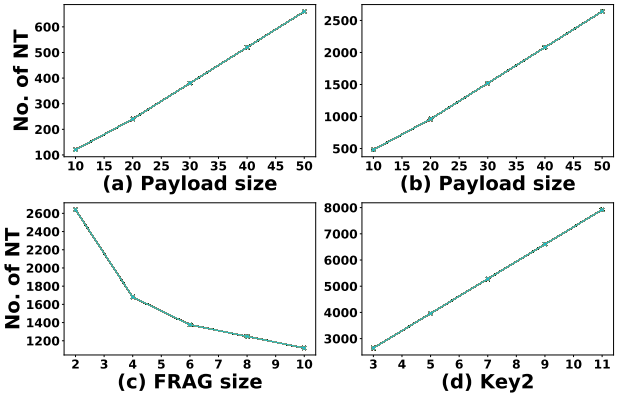


Figure 6: Nucleotides required to build payload considering (a) no encryption, (b-d) encryption.

The first result analyses the variations in the N_p with respect to payload hiding using only the first phase of the algorithm (Fig. 6(a), no encryption, carrier size of 100, $Size_{fr} = 2$), both phases (Fig. 6(b), encryption with $key2 = 3$, carrier size of 100, $Size_{fr} = 2$), and also for different $key2$ values (Fig. 6(d), encryption with various $key2$, carrier size 100, $Size_{fr} = 2$, $Size_p = 50$) while using both phases of the algorithm. Fig. 6(c) (encryption with $key2 = 3$, carrier size of 100, $Size_p = 50$) shows that with the increments in $Size_{fr}$, the N_p also decreases. This is due to the longer fragment size having an effect on less number of fragments, which also results in less number of additional start and end characters, as well as order

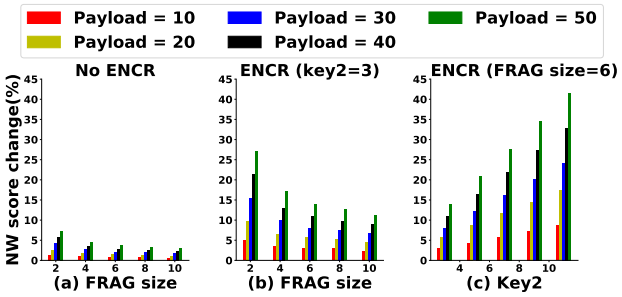


Figure 7: Changes in NW scores for (a) non encrypted, (b) encrypted payload and (c) key2 values

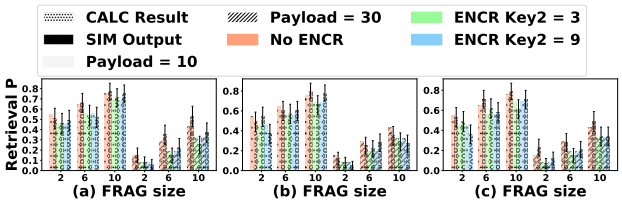


Figure 8: Retrieval probability, with carrier sizes of (a) 100 (b) 200 and (c) 300, where error rate is 0.005

numbers and separators. This means that using large $Size_{fr}$ is ideal in order to reduce the number of fragments. We can observe from the graphs that the second phase encryption process results in a linear increase of N_p. We can also observe that the size increases further when higher values of $key2$ are used, which means that a low $key2$ value should be used. That means shorter domain names and smaller port numbers will keep the N_p size reasonably low.

As described earlier, the NW Global alignment score (NW_{score}) provides us with an idea on how close two sequences are aligned. Higher scores represents closer alignments between two sequences and the score depends on the number of pattern matches, as well as the gaps and number of gap lengths. Our calculated scores are cross checked using BLAST, a widely renowned tool available online [2]. Fig. 7 presents the NW_{score} for different range of parameters. As long as the payload size increases, the gap will also increase and small fragment sizes will increase the number of parts (i.e., increments in the number of gaps with little increments in the total gap length will result in few more extra markers and order number characters). Since encryption will result in an increase in the nucleotide number, the score will be changed significantly compared to the non-encrypted payload since the gap size will be quite large. Therefore, if our objective is to minimize the changes to the score, we should prefer the non-encryption approach with larger fragment size so that the DNA does not change significantly when comparing the *target DNA sequence* with the alignment database.

The results shown in Fig. 8 are from our analysis for both randomly generated and real DNA cover sequences. It is clear that the retrieval probability does not depend on the carrier size, and this is understandable given that the equations used for the $prob_r$ does not consider the $Size_c$. The retrieval probability depends mainly on

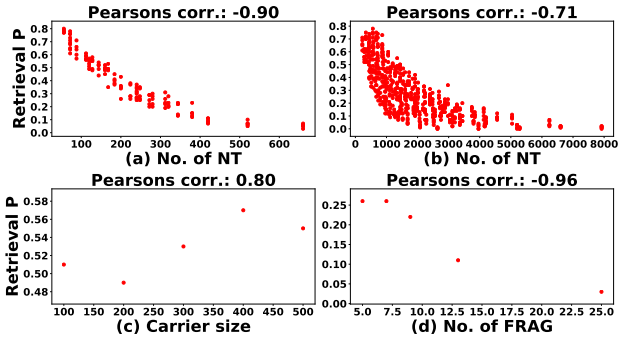


Figure 9: Correlations between retrieval probability and number of nucleotides for (a) non encrypted, (b) encrypted payload, (c) carrier size and (d) fragment size

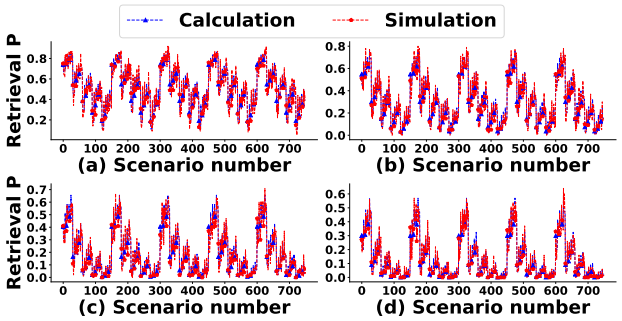


Figure 10: Calculated and simulated retrieval probabilities comparison for error rate (a) 0.0025 (b) 0.005 (c) 0.0075 and (d) 0.01 using randomly generated carrier.

the $Size_p$ and also varies with the $Size_{fr}$, as a higher $Size_{fr}$ means a lower N_{fr}. This is because the lower number of nucleotides for the payload will result in higher $prob_r$. For the encryption, the $prob_r$ decreases slightly and is very negligible for increasing value of $key2$. Therefore, to ensure higher retrieval probability, we should select a non-steganography technique. In the case where we have the flexibility of making a compromise with $prob_r$, then it is appropriate to choose the additional steganography technique using any number of $key2$ values. However, the payload size should be as small as possible with higher fragments. Results shown in Fig. 8 are further confirmed by Fig. 9. From Fig. 9 we see that the $prob_r$ is highly correlated with the N_p for the non-encryption case (See Fig. 9(d)). Correlation in the number of N_p is slightly less for the encryption case (Fig. 9(b)) compared to the non-encryption approach (Fig. 9(a)). This is because in encryption, the reading errors for many nucleotides are irrelevant in the sequencer. Correlation with the carrier DNA size is also very low (Fig. 9(c)), and this is due to the $prob_r$ having no dependence on the carrier size.

As we measured retrieval probabilities with respect to various basecall error probabilities and per scenario basis (where a scenario is a combination of parameters, e.g. payload size, fragment size, carrier size and $key2$ value if steganography is applied), we compared

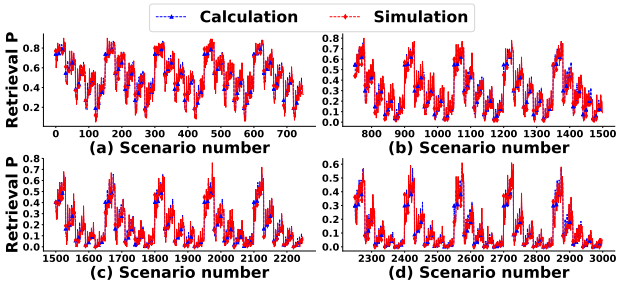


Figure 11: Calculated and simulated retrieval probabilities comparison for error rate (a) 0.0025 (b) 0.005 (c) 0.0075 and (d) 0.01 using real carrier

the calculated results for the $prob_r$ with the simulated results. The results are shown individually for all of the evaluated scenarios (i.e. scenario number as the horizontal axis). Fig. 10 shows that the calculated results are very close to the simulated results and it stands for all four different $prob_e$ values (Fig. 10(a-d)). This indicates that the equations are sufficient for estimating the retrieval probabilities considering parameters as fragment size, payload size, $key2$ values, etc.

Fig. 11 shows the results by repeating the experiments using real DNA sequences and the result is similar to Fig. 10, where the calculated and simulated $prob_r$ are quite similar. Similar to the previous figure, the results are shown individually for all of the evaluation scenarios. For both real and simulated carriers, we see repeated patterns (Fig. 10 and Fig. 11). As we have already seen that $prob_r$ does not depend on carrier size $Size_c$, so the pattern is repeating for each carrier. We find the number of repeats are 5 and 20 times in Fig. 10 and Fig. 11, respectively, as the number of carriers used is also 5 and 20, respectively.

6 CONCLUSION

We have shown how perpetrators can take advantage of Trojan spyware in a DNA-sequencing pipeline by triggering it only with specially engineered DNA sample leaving behind bare minimum footprints. Reveling such footprints can become more difficult by applying a state-of-the art steganography technique and that can allow the spyware more time before a security algorithm traces the payload inside the target DNA sample. We have shown the possibility of a successful attack considering error rates from the sequencer reading process by using both randomly generated and real DNA samples. In the future, we intend to perform wetlab experiments to validate the feasibility of such attacks and to perform an End-to-End evaluation.

ACKNOWLEDGMENTS

This work was supported by Science Foundation Ireland through the SFI VistaMilk (16/RC/3835) and CONNECT (13/RC/2077) research centres.

REFERENCES

[1] D. Ahn and G. Lee. 2015. BoundShield: Comprehensive Mitigation for Memory Disclosure Attacks via Secret Region Isolation. *IEEE Transactions on Dependable and Secure Computing* 12, 4 (2015), 387–399.
[2] National Center for Biotechnology Information (NCBI). 2019. Basic Local Alignment Search Tool (BLAST). https://blast.ncbi.nlm.nih.gov/Blast.cgi
[3] S. Jain, F. Farnoud Hassanzadeh, M. Schwartz, and J. Bruck. 2017. Duplication-Correcting Codes for Data Storage in the DNA of Living Organisms. *IEEE Transactions on Information Theory* 63, 8 (2017), 4996–5010.
[4] H. Jin, B. Liu, Y. Du, and D. Zou. 2018. BoundShield: Comprehensive Mitigation for Memory Disclosure Attacks via Secret Region Isolation. *IEEE Access* 6 (2018), 36341–36353.
[5] H.H. Nguyen, J. Park, S. Park, C.-S. Lee, S. Hwang, Y.-B. Shin, T. Ha, and M. Kim. 2018. Long-Term Stability and Integrity of Plasmid-Based DNA Data Storage. *Polymers* 10, 28 (2018), 1–10.
[6] Malathi P., M. Manoaj, R. Manoj, V. Raghavan, and R.E. Vinodhini. 2017. Highly Improved DNA Based Steganography. *Procedia Computer Science* 115 (2017), 651–659.
[7] D.R. Paoletti, D.E. Krane, M.L. Raymer, and T.E. Doom. 2012. Inferring the Number of Contributors to Mixed DNA Profiles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9, 1 (2012), 113–122.
[8] Kevin Santoso, Suk-Hwan Lee, Won-Joo Hwang, and Ki-Ryong Kwon. 2016. Sector-based DNA information hiding method. *Security and Communication Networks* 9, 17 (nov 2016), 4210–4226.
[9] Wei Sun, Katerina Kechris, Sean Jacobson, and et al. 2016. Common Genetic Polymorphisms Influence Blood Biomarker Measurements in COPD. *PLOS Genetics* 12, 8 (2016), 1–33.
[10] Milad Taleby Ahvanooey, Qianmu Li, Jun Hou, Hassan Dana Mazraeh, and Jing Zhang. 2018. AITSteg: An Innovative Text Steganography Technique for Hidden Transmission of Text Message via Social Media. *IEEE Access* 6 (2018), 65981–65995.
[11] Oxford Nanopore Technologies. 2019. MinION. https://nanoporetech.com/products/minion
[12] B. D. Unluturk, M. S. Islam, S. Balasubramaniam, and S. Ivanov. 2017. Towards Concurrent Data Transmission: Exploiting Plasmid Diversity by Bacterial Conjugation. *IEEE Transactions on NanoBioscience* 16, 4 (2017), 287–298.
[13] USENIX Association 2017. *Computer security, privacy, and DNA sequencing: Compromising computers with synthesized DNA, privacy leaks, and more.* USENIX Association.
[14] R.K. Varshney, V.K. Singh, J.M. Hickey, X. Xun, D.F. Marshall, J. Wang, D. Edwards, and J.-M. Ribaut. 2016. Analytical and Decision Support Tools for Genomics-Assisted Breeding. *Trends in Plant Science, Elsevier* 21, 4 (2016), 354–363.
[15] S.E. Wallen, M. Lillehammer, and T.H.E. Meuwissen. 2017. Strategies for implementing genomic selection for feed efficiency in dairy cattle breeding schemes. *Journal of Dairy Science, Elsevier* 100, 8 (2017), 6327–6336.

Appendix D

Using Deep Learning to Detect Digitally Encoded DNA Trigger for Trojan Malware in Bio-Cyber Attacks

Journal Title	Scientific Reports, Nature
Article Type	Regular
Complete Author List	Mohd Siblee Islam, Stepan Ivanov, Hamdan Awan, Jennifer Drohan, Sasitharan Balasubramaniam, Lee Coffey, Srivatsan Kidambi and Witty Sri-saan

scientific reports

Check for updates

OPEN

Using deep learning to detect digitally encoded DNA trigger for Trojan malware in Bio-Cyber attacks

M. S. Islam[1]✉, S. Ivanov[1], H. Awan[4], J. Drohan[3], S. Balasubramaniam[2], L. Coffey[3], S. Kidambi[5] & W. Sri-saan[2]

This article uses Deep Learning technologies to safeguard DNA sequencing against Bio-Cyber attacks. We consider a hybrid attack scenario where the payload is encoded into a DNA sequence to activate a Trojan malware implanted in a software tool used in the sequencing pipeline in order to allow the perpetrators to gain control over the resources used in that pipeline during sequence analysis. The scenario considered in the paper is based on perpetrators submitting synthetically engineered DNA samples that contain digitally encoded IP address and port number of the perpetrator's machine in the DNA. Genetic analysis of the sample's DNA will decode the address that is used by the software Trojan malware to activate and trigger a remote connection. This approach can open up to multiple perpetrators to create connections to hijack the DNA sequencing pipeline. As a way of hiding the data, the perpetrators can avoid detection by encoding the address to maximise similarity with genuine DNAs, which we showed previously. However, in this paper we show how Deep Learning can be used to successfully detect and identify the trigger encoded data, in order to protect a DNA sequencing pipeline from Trojan attacks. The result shows nearly up to 100% accuracy in detection in such a novel Trojan attack scenario even after applying fragmentation encryption and steganography on the encoded trigger data. In addition, feasibility of designing and synthesizing encoded DNA for such Trojan payloads is validated by a wet lab experiment.

Genetic sequencing has become an essential tool for analyzing numerous DNAs that are used in the field of medicine, agriculture, as well as forensics. Numerous systems have been developed over the years to increase accuracy, such as throughput shot-gun sequencing technologies (e.g., vector-borne pathogens detection in blood[1], food authentication and food fraud detection[2], or even molecular data to be transported through artificial biological networks[3,4]). Recent developments in sequencing technology have also been miniaturized to allow mobile sequencing and one example is the *Minion*[5]. We have recently witnessed the importance of timely sequencing from oral samples due to the COVID-19 pandemic, which continues to apply pressure on the health care system[6]. The clear benefits of expanded COVID-19 testing[7] calls for an expansion of the existing testing (e.g. STEMI[8]) approaches. The importance of sequencing can also be seen in detecting and tracking mutations in other types of infectious diseases, where examples include Lassa Fever[9] or other prevalent pathogens[10], such as seasonal flu[11] or bacterial infections where new strains resistant to existing antibiotics can be identified[12,13].

As the genetic sequencing will inevitably introduce additional pressure on the already overburdened healthcare services, it is likely that the genetic analysis may be outsourced to private sequencing services. Similar approaches have already been successfully adopted for other testing programmes (e.g. Cervical Screening Programme in Ireland[14]). The services will act as an on-demand genetic-testing infrastructure that receives and analyses samples on behalf of the hospitals, medical practices and other healthcare organizations. While this approach alleviates pressure on the healthcare system, the system is vulnerable to Bio-Cyber Hacking[15].

[1]VistaMilk Research Centre, Walton Institute, South East Technological University, Waterford, Ireland. [2]School of Computing, University of Nebraska-Lincoln, Lincoln, NE, USA. [3]Pharmaceutical and Molecular Biotechnology Research Centre, South East Technological University, Waterford, Ireland. [4]Munster Technological University, Cork, Ireland. [5]Department of Chemical and Biomolecular Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA. ✉email: sibleeislam@gmail.com

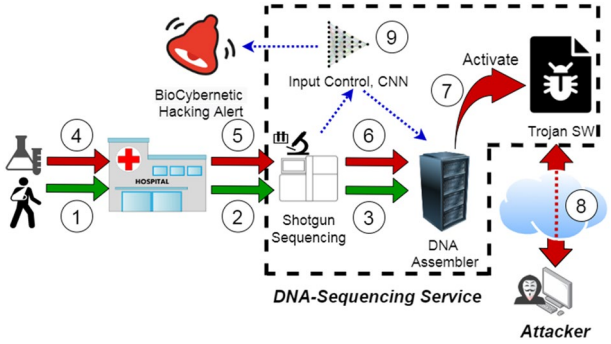


Figure 1. Hybrid Trojan Bio-Cyber Hacking Attack. Steps 1–3 indicate a typical genetic sequencing operation for patients. Steps 4–6 indicate a situation where a hacker has embedded their IP address and Port number into a DNA that will trigger a remote connection from a Trojan-horse infected software tool leading to a connection to the attacker in Step 8. Our proposed approach utilizes Deep-Learning to detect Trojan payload in digital data using encoded into DNA strands that can prevent the attack. (The image is drawn using draw.io).

Our definition of Bio-Cyber Hacking refers to an attack that is hybrid between ICT systems and biological mediums. From the ICT system side, we assume that the pipeline of the sequencing service uses a DNA-analysis toolbox infected with Trojan Software. Malware, such as a Trojan, can be implanted at the API levels[16], within mobile software[17] and even in machine learning models[18]. Trojans can also be implanted into hardwares[19–21] of computers, as well as IoT devices[22]. In our scenario, the Trojan contains an empty slot for the IP address and port number for remote connections to an external machine. On the biological side, an attacker encodes the IP address and port number into DNA strands. Using DNA-steganography, the attacker devises synthetic DNA that contains the payload and still maintains resemblance with natural DNAs. We will explain the process in Fig. 1, where we will first explain a sequencing process for normal DNA (steps 1–3) and then explain a hacking situation (steps 4–8). In (Fig. 1 (1–2)), the service uses one of the state-of-the-art sequencing techniques, e.g. shotgun sequencing, to analyze DNA materials extracted from each of the samples (e.g. *E. coli* Plasmid and Cellular DNAs). The machine randomly splits DNA molecules into multiple fragments or reads of a predefined length, then it concurrently sequences each read to establish its nucleotide structure. The original DNA is then assembled from the reads (Fig. 1 (3)). This is a computationally complex process that often involves the use of dedicated resources, often called DNA-sequencing pipeline[23]. Let us now consider an attack situation. Initially the Trojan remains dormant, while the toolbox performs the legitimate DNA-analysis. The trigger sample is collected by the hospital (i.e., by swabbing) and sends the samples to the sequencing service for analysis (Fig. 1 (4)). The samples are then analyzed by the sequencer (Fig. 1 (5)). There the sample is fragmented, sequenced and assembled (Fig. 1 (6)). During the assembly, the DNA toolbox retracts the payload and wakes the Trojan (Fig. 1 (7)), and this happens is when the DNA sample that contains the web address and port number of a remote server controlled by the attacker is detected by the digital DNA data that is passed from the sequencer to the computer that contains the DNA-analysis toolbox infected with the Trojan. The Trojan establishes a connection with the remote server (Fig. 1 (8)), where the Trojan either opens a cyber backdoor, transfers files, or executes commands from the attacker. Either of these actions presents a substantial threat to the integrity of DNA-analysis and patient diagnostics.

In this article, we develop a solution that is complementary to the existing general-purpose techniques. The solution builds on our previous work that only focused on steganography techniques to hide IP address and port numbers into DNA strands[24] and investigates the use of input control (Fig. 1 (9)) as a countermeasure to the Trojan Bio-Cyber attacks. The input control is an intermediary between the DNA-sequencer and the pipeline. With the help of a specially designed and trained Deep Neural Network (DNN), the control assesses each DNA read generated by the sequencer to establish whether the read comes from a trigger sample. Absence of suspicious reads assures cybersafety of further DNA-assembly, but a detection of a trigger sample terminates its further processing. This prevents activation of the Trojan software and limits the pipeline's exposure. In recent times, there is a lot of interest in the use of deep learning for malware detection[25,26,27]. Deep learning techniques are also applied to Trojan detection[19,28] in conventional cyber attacks. To the best of our knowledge exploiting the Buffer overflow vulnerability in DNA sequencing pipeline using a specially designed DNA was first demonstrated in[15]. To detect DNA sequences containing the payloads for buffer overflow exploit, we proposed Case-based Reasoning (CBR)[24], where we found that such payloads results in sequences that are quite different from the DNA sequences which are naturally available. Moreover, we investigated the recovery rate of the DNA sequences containing the malicious payloads that was inserted into bacteria after spraying them on various types of surfaces. In another article we have shown how a Trojan Attack is possible in a DNA sequencing pipeline[29], but the possibility of creating such sequence was not validated by conducting any wet lab experiments or detecting the trigger, which is what we have investigated in this work. In both of our previous work, we did not consider keeping the natural appearance of the DNA while designing the payload to make the detection harder. In this article we improved our algorithm of making the payload harder to detect and proposed a CNN based detection technique.

Figure 2 illustrates the construction of the payload that is embedded into a DNA sequence, and in this specific example we focus on a bacterial plasmid. We re-designed the construction of the payloads to make it similar to

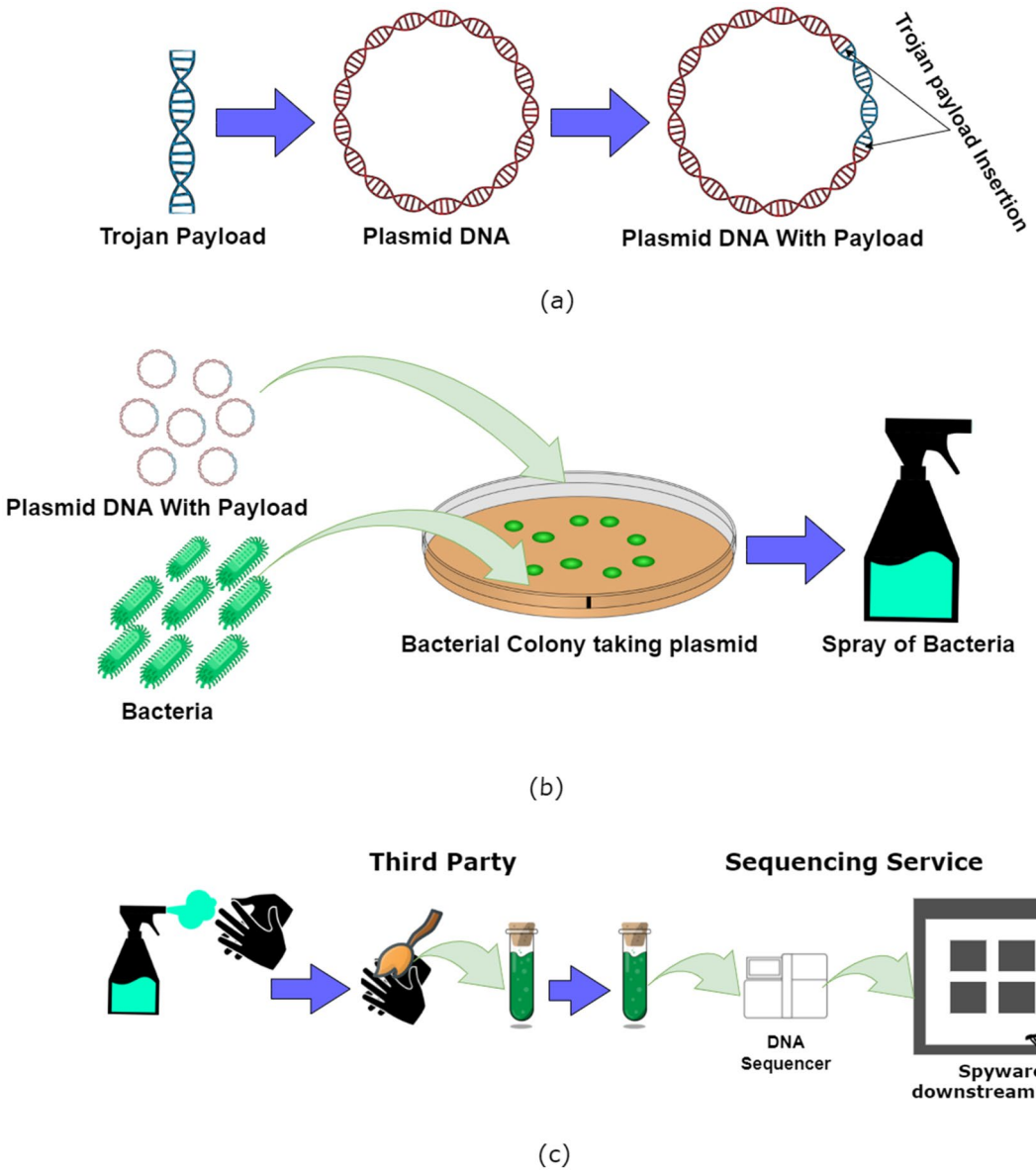


Figure 2. Trojan Bio-Cyber Hacking: Payload Preparation and Attack Scenario example using DNA plasmids. (**a**) A Trojan payload (using encryption and steganography) is encoded into a DNA sequence which is developed and inserted into the plasmid DNA. Antibiotic resistant gene sequences will also be inserted into the plasmid DNA in a similar way. (**b**) The DNA plasmid and the bacteria will be transferred into rich media so that the bacteria can uptake these plasmids[24]. Bacteria resistant to the antibiotic will survive and be transferred into a spray. (**c**) The bacteria can now be sprayed on hands or gloves and provided to a third party which can collect samples (from hand or gloves). The third party will then send these samples to the company for sequencing. When the sequence will be processed by the tools having the Trojan, it will be activated to perform the malicious activities. (The image is drawn using draw.io).

a natural DNA sequence in order to increase detection difficulty. The construction of the DNA is based on the sequence used in[29].

Methods

In this section, various terms used in the article will be defined and then the steganography techniques will be described, which is applied on the payload used for malicious activities as a means of secrecy of operations. Following that we will describe the injection method of the payload into a host DNA. This is followed up with the description of the deep learning model proposed as a detection method to counter the Trojan attacks.

Trojan payload. The payload DNA for triggering the Trojan malware will be encoded into a DNA sequence and will be referred simply as 'payload' in the rest of the article. The payload will be hidden inside a longer DNA string, which is considered as 'host DNA'. In order to prevent detection, the content of the payload will be first

divided into smaller parts and then encoded into smaller DNA sequences, which will be called as 'fragments' and this process will be known as 'fragmentation'. The fragments can be inserted in a random order and at random positions of the host DNA. Substitution technique, i.e., replacing a nucleotide of the host DNA with a nucleotide of the payload DNA or fragment DNA (if fragmentation is applied), is considered as an insertion technique. 'Retention' is the process of skipping a particular number of nucleotide positions of the host DNA to substitute by the nucleotide of the encoded/fragment DNA while performing the insertion. Both encryption and retention will be considered when steganography is applied, where the encryption process will be performed before the retention. The details of the processes including encryption will be described in the subsequent sections of the article. After completing the insertion process, the obtained DNA string is considered as the 'resultant DNA'.

In general the host DNA string will be significantly larger compared to the encoded DNA for the payload. Therefore, the Trojan software needs to perform processes such as identifying those fragments, applying decryption and decoding techniques before merging and rearranging them in order to activate the malware process to trigger the hacking operation. As a result, the Trojan software should apply these processes to integrate the substrings to create the full DNA string as an additional task beside performing its normal functional tasks. The caveat of such an approach is that the computational complexity will be significantly high and the Trojan software might be under suspicion straight away as it will take significantly higher time and consume higher memory. To prevent this suspicious behaviour, the Trojan software will need to efficiently determine the location to perform decryption and decoding and this will be achieved through 'tags'. The tags are tiny snippets of chosen DNA sequences that indicate the start and end of the fragments that will be searched by the Trojan software, and we refer to this process as 'tagging'.

One of the critical challenges in packaging the Trojan payload is the delivery system which can act as the carrier for the DNA materials. To this extent, liposomes and lipid-based nanoparticles have been extensively used for targeted gene delivery to various coordinates. Liposomes, also referred to as vesicles, are extremely versatile carriers that have been studied and utilized extensively for drug delivery applications including gene and mRNA due to their ease of creation, large protective hydrophilic inner cavity for encapsulation, high degree of freedom for exterior customization, and controllable drug release kinetics. Recent success of mRNA vaccines for COVID is attributed to such lipid based platforms as a delivery vehicle for mRNA. These can be extended to packaging the Trojan payload to enhance the stability of the DNA and also establish targeting capabilities to target specific locations for Cyber-hacking. Furthermore, there are innovative and robust platforms that can integrate these lipid nanoparticles embedded within substrate and matrix based on polymer based films that can control the release of these Trojan payloads and extend their stability[30]. Also this platform can also facilitate hiding these Trojan payloads from detection and embed multiple payloads. This platform provides ways to transport the Trojan Payload into the targeted region beyond security measures by embedding them into entities including clothes, skins, pens or papers as examples.

Steganography. In this article we consider a scenario where the perpetrator encodes the attack details (i.e., web address and port number) into a DNA, which are used as a trigger sample. To avoid the detection of this sample and cover the identity of the attacker, the encoding uses an extension of the DNA Steganography technique proposed in[29].

The extended steganography technique proposed in this article has five steps and this includes *fragmentation*, *encryption*, *encoding*, *tagging* and *retention*. First, the web address and port number injected into the DNA are divided into fragments of a predefined length. Since each fragment is shorter than the original address, this will increase the difficulty in the detection process post injection. Next, the binary of the fragment is XOR-encrypted using a predefined key. This is followed up by encoding with four basic nucleotides, i.e., "00" bit-pairs are encoded as "A", "01" as "C", "10" as "G" and "11" as "T". The ACTG-encoding (represent four nucleic acids, which are Adenine, Cytosine, Thymine and Guanine) is enclosed in the nucleotide brackets where the ACTG tags mark the beginning and the end of the injection within the DNA. These tags are selected so that the natural DNAs are unlikely to include both the start and end tags separated by a number of nucleotides that is required to encode a malicious fragment. The tags need to be sufficiently short in order to reduce the footprint of the injected fragment as well as increase the similarity with the host DNA and avoid detection. Finally, the retention stage expands the result of the tagging using the symbol "*" (see Eq. 1). The expansion is performed in a way that a predefined number of retention symbols is inserted between each of the two consecutive nucleotides. The positions of the retention symbol determine that the nucleotides of the host DNA will remain unchanged as a result of the malicious code injection. Thus, for a retention number equal to 2, only the first of each 3 consecutive nucleotides of the host DNA will be replaced. The second and third nucleotides will remain unchanged. This is done to increase the similarity between DNA of the trigger sample and the host DNA.

Injection methods. In this article we consider substitution as the preferred method of injecting the Trojan payload into the host DNA. Consider the case when the Trojan payload d_{load} (with encoded nucleotides and retention symbol "*" after applying encryption and steganography as described above) is injected into the DNA, d_{host}, at position i. The result of the injection will present a nucleotide string inj, having the length equal to the length of d_{load}. The length of the d_{host} and d_{load} strings is determined by a function called len, which reflects the number of characters in both strings. The nucleotide at position $j \in [0, len(d_{host}))$ of inj will be the insertion position i and based on $d_{load}[j]$. If the value of j does not fall between the range required for the injection position, which is from i to $i + len(d_{load}) - 1$ as this location is required for the payload injection, then the actual nucleotide of host $d_{host}[j]$ will be used, i.e. $inj[j] = d_{host}[j]$. Otherwise, the value of $inj[j]$ will depend on $d_{load}[j-i]$, since the value $[j-i]$ determines the index of the d_{load} and this has to be considered when it starts from 0 (for the very first substitution point $j = i$) up to $len(d_{load}) - 1$. If the $d_{load}[j-i]$ contains a retention

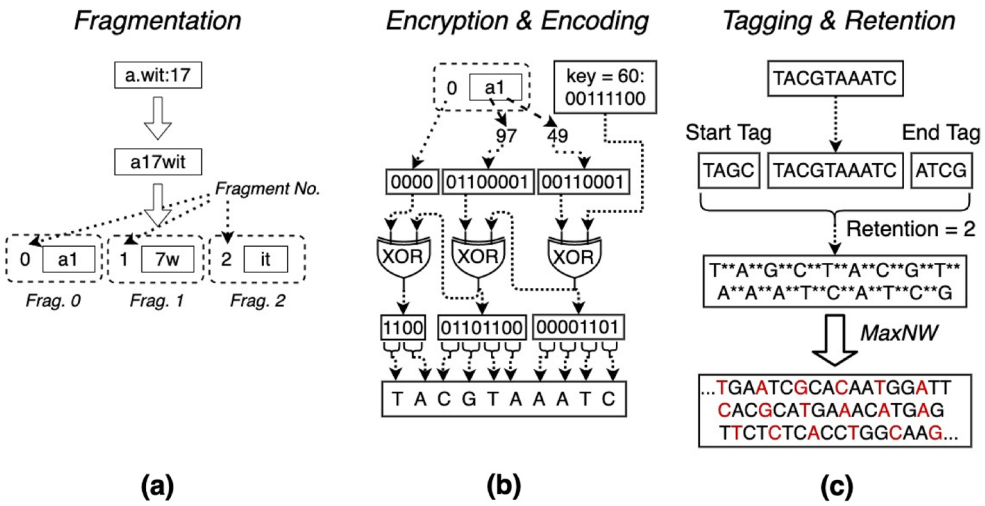


Figure 3. DNA Steganography, Workflow: (**a**) payload fragmentation, (**b**) fragment encryption and encoding, (**c**) tagging, retention and host injection. (The image is drawn using draw.io).

symbol "*", i.e. (d_{load} [j–i] == "*" then $inj[j] = d_{host}[j]$ (this means the original nucleotide is used for retention) otherwise $inj[j] = d_{load}[j-i]$. This substitution procedure can be defined as:

$$inj[j] = \begin{cases} d_{host}[j] \text{ if } j < i \text{ or } j \geq i + len(d_{load}), \\ d_{host}[j] \text{ if } j \in [i,\ i + len(d_{load})) \text{ and } d_{load}[j-i] = \text{"}*\text{"}, \\ d_{load}[j-i] \text{ if } j \in [i,\ i + len(d_{load})) \text{ and } d_{load}[j-i] \in \{A, C, T, G\}. \end{cases} \quad (1)$$

We define *elementary* domain dom_{ELM} that consists of all the possible positions for a Trojan payload injection. Naturally, such a substitution can be carried out only from the position i onwards and is represented as:

$$dom_{ELM} = [0, len(d_{host}) - len(d_{load}) + 1], \quad (2)$$

which is referred to as the *injection domain* and refers to the indices (i.e., values of i) of d_{host}. We use $inj(d_{host}, d_{load}, i)$ to denote the substitutions in dom_{ELM}. Similarly, to denote the substitutions carried out on subdomain $dom \subset dom_{ELM}$, we use $inj_{dom}(d_{host}, d_{load}, i)$. This subdomain introduces additional restrictions that may be required to preserve particular areas within the host DNA. Figure 3 presents the five stages/steps involved in the DNA steganography technique used in this article.

Note that in this article we only consider payloads that consist of a web address (represented by a Tiny URL) and port number of a remote server controlled by the attacker (For payloads used in the analysis, please see: https://bitbucket.org/sibleeislam/bio-cyber-hacking). The payload has the following semantics:

< prefix : character string > . < suffix : character string > : < port number :string of digits >

As mentioned above, the fragmentation (Fig. 3a) is the first stage of the DNA steganography. First, the payload is rearranged so that the address prefix is followed by the port number and then the address suffix. This representation allows the reduction in the auxiliary "." and ":" characters from the payload, and therefore, size reduction of the entire payload. Subsequently, the rearranged payload is divided into fragments, substrings of a predefined length (e.g. 2 characters as shown in Fig. 3). Each of the fragments is attached with its serial number as a prefix. As only tiny URLs are used in the tojoan payload address, we assume that no more than 16 fragments can be formed. If we want to consider a web address with subdomains then the top level domain will be the suffix and the rest of the part of the domain name will be the prefix.

The next step after frangementation is encryption, where each fragment is encrypted and nucleotide-encoded as illustrated in Fig. 3b. At this stage, the fragment is represented as a bit-array where the first 4 bits represent the fragment's serial number, followed by a series of 8-bit representations of fragment characters. Each character is represented by the binary of its ASCII code. The bit-array is then XOR encrypted using a predefined key (e.g. 60 as depicted in Fig. 3b). This results in a sequence of bit-pairs, which are then encoded into nucleotides strings that represent the DNA.

The next step after encryption is encoding as shown in Fig. 3c. The nucleotide-encoding of the fragment is attached with a start and end tag as prefix and suffix, respectively. The resultant string is then expanded so that a predefined amount of retention symbols is added between each two consecutive nucleotides (e.g., 2 symbols as in Fig. 3c). The expanded string is then injected into the host DNA using MaxNW procedure, which is described next.

MaxNW technique. Needleman-Wunsch, or NW score is one of the most popular methods to assess the similarity between two DNA samples. This score considers the string-based nucleotide representation of the DNA molecules and calculates the number of symbol substitutions, gaps (i.e., symbol insertion or deletion) and

their expansions (i.e., continuation of gaps) required to align two strings. Depending on the circumstances, a specific penalty system is applied to each of the operations as well as matches between DNA nucleotides. The system is constructed in a way to favor certain alignment patterns. As in the experiments performed in this work, injecting payload typically constitutes not more than 10% of the host DNA string size, therefore we use PAM10 substitution scoring matrix[31] as the cost matrix for nucleotide substitution. Following this methodology outlined in[32], we set the costs for the gap opening and extension to 15.79 and 1.29 for the PAM10 substitution, respectively.

In this article, we use NW scores to measure the similarity between d_{host} and $inj(d_{host}, d_{load}, i)$. Based on the penalties defined above, the NW score increases as similarity between d_{host} and $inj(d_{host}, d_{load}, i)$ increases and reaches its maximum if d_{host} and $inj(d_{host}, d_{load}, i)$ are equal. In other words, the injected payload fits into the d_{host} naturally at position i. Lets assume the NW score is maximum when the insertion position (the value of i) is i_{max}. To emulate the attacker, the malware NW score, $MaxNW_{dom}$, is defined as:

$$MaxNW_{dom}(d_{host}, d_{load}) = inj(d_{host}, d_{load}, i_{max}), \tag{3}$$

where

$$i_{max} = Arg\left(max_{i \in dom} NW\left(d_{host}, inj(d_{host}, d_{load}, i)\right)\right). \tag{4}$$

when multiple payloads for malicious activity injections $D_{load} = \left\{d_{load,1}, \ldots, d_{load,n}\right\}$ are introduced into the same host DNA, dynamic programing is used to determine the optimal positions for the injections. The technique employs a recursive procedure, where at each step the best insertion is sought amongst all possible positions. So, initially $inj(d_{host}, d_{load}, i)$ and dom_{ELM} are considered for the substitution and the domain for the substitutions for each of the payloads. Then the injection position of the payload having maximum NW Score will be considered for that particular payload injection and that portion of the injection will be restricted for further injections. For further steps, the subdomain dom and injection for subdomain $inj_{dom}(d_{host}, d_{load}, i)$ will be considered as the restriction is applied. Lets assume, the maximum NW Score and the indices considering subdomain are $MaxNW_{dom^*}$ and , i^* respectively. The injection process will be repeated until all the payloads are injected. Thus, this recursive procedure can be described as:

$$MaxNW_{dom}(d_{host}, D_{load}) = MaxNW_{dom^*}\left(MaxNW_{dom}\left(d_{host}, d_{load,i^*}\right), D_{load}/d_{load,i^*}\right), \tag{5}$$

where

$$i^* = Arg\left(max_{j \in [0, len(D_{load}))} NW_{dom}\left(d_{host}, d_{load,j}\right)\right). \tag{6}$$

Deep learning. Various machine learning algorithms and even regular expression based classification techniques might be useful to detect the presence of trigger samples in the DNA sequence as the tags will express a pattern. However, the tag will be unknown to the detection subsystem and the number of available tags will grow exponentially with the number of nucleotides used for the tags (please see Fig. 5a). Therefore, a machine learning algorithm will be a better option as the regular expression will need to consider a huge number of possible tags. Again, a big challenge of machine learning algorithms (e.g. Random Forest Support Vector Machine, K nearest Neighbors etc.) is feature extraction and feature selection. Very large number of features can be applied using these techniques. The success of the prediction technique mainly depends on finding appropriate feature extraction and feature selection techniques[33]. To extract the feature, Natural Language Processing (NLP)[33] is applied to a DNA sequence or DNA is considered as a time series or genomic signal and then signal processing techniques like Discrete Fourier Transform (DFT)[34] and Discrete Wavelet Transform (DWT)[35] are applied. RNN is good for sequential data or time series data and also where the context, especially the previous classification, is important[33]. In our classification problem, each classification is independent. However, it is interesting to see how to feed the sequence and get intermediate classifications in the hidden layer and then get the final classification. On the other hand, in CNN the convolutional layers part of the architecture does the feature extraction and selection for us and the flat layers followed by the convolutional layers does the classifications[33,36]. This is the reason we have selected the use of CNN. In this article, we use a 1-Dimensional Convolutional Neural Networks (1D CNNs) to identify the Trojan payload within the natural DNAs. This section will provide a brief overview of the CNNs we utilized for this work. An overview of various Deep Learning methods, including CNNs, used in genetics analysis can be found in[37].

Figure 4 depicts the typical architecture of a 1D CNN. Similar to any other neural network, the 1D CNN consists of neurons organized in layers. The architecture proposed in this article uses the following layers: input, convolution, pooling, and dense.

The first layer represents the input of the network. Here, each of the DNA sequences' classification is transformed into the set of primary features, i.e., inputs of the network. Each nucleotide of the DNA is represented by a vector of 5 boolean indicator values. The first 4 values indicate whether the nucleotide are found to be equal, whereas the 5th value indicates whether the nucleotide can be determined (i.e. N—undetermined). As an example, A-nucleotides of the DNA will be represented by (1,0,0,0,0) indicator vectors, C-nucleotides will be represented by (0,1,0,0,0), and undetermined nucleotides will be represented by (0,0,0,0,1). To formulate the primary features of the entire DNA, indicator vectors for all its nucleotides are concatenated in the order of the pattern that is found in the original DNA.

The input layer is followed by a number of CONV1D layers as shown in Fig. 4. At each layer, multiple filters are applied to the kernels of a particular size. The resultant product is then subjected to ReLU activation. CONV1D

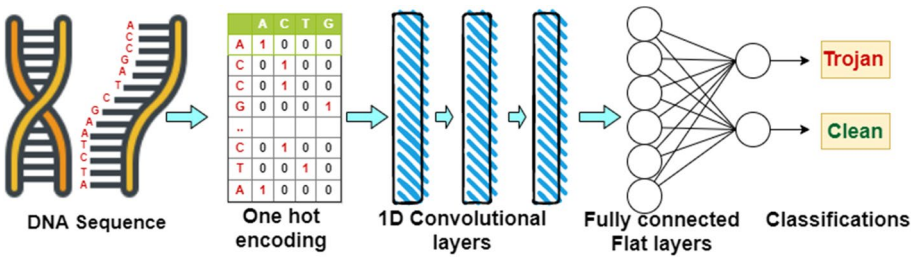


Figure 4. 1-Dimensional Convolutional Neural Network (1D CNN): Architecture. (The image is drawn using draw.io).

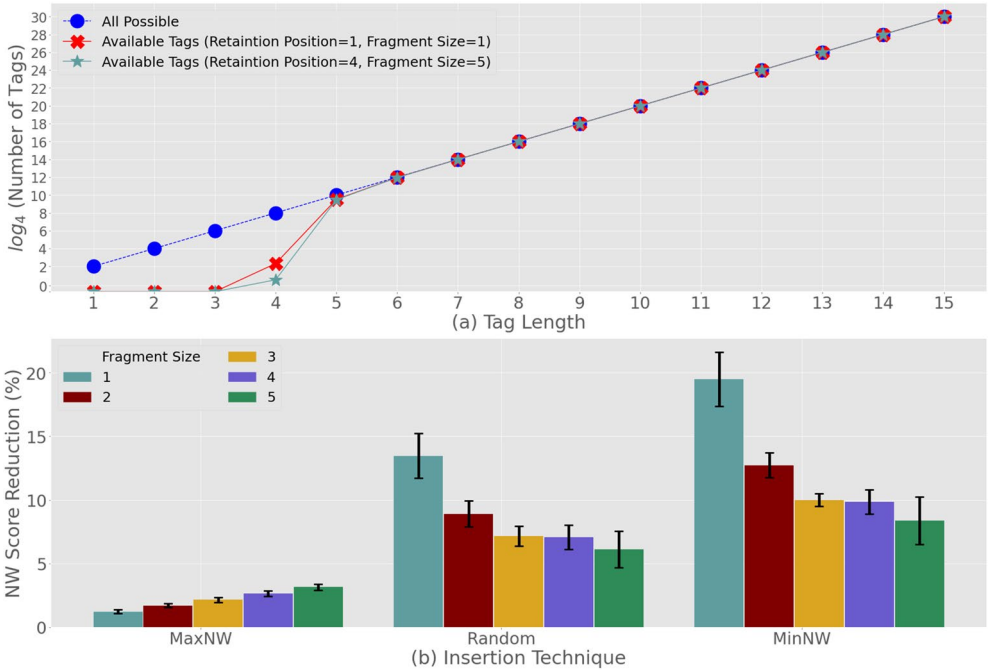


Figure 5. Trigger Sample Design with the use of DNA-Steganography: (**a**) nucleotide tag selection; (**b**) the impact of fragmentation.

layers are followed by 1 MaxPool, one dense layer with ReLU activation function, and finally 2-neuron SoftMAX layer, the output of which provides the certainty of the sample to be determined if it contains the address information. In this article, we consider networks with varying numbers of CONV1D layers, the size of their kernel and the number of filters used. We also investigated the impact of the kernel size of the MaxPool layer and the size of the ReLU dense layer. Each network is trained for 3000 epochs using 75% of all available DNA samples. The remaining 25% of the samples are used to test the performance of the trained network.

Results and discussion

For the Trojan infected softwares, the secrecy of operation is of paramount importance. The longer the Trojan remains undetected, the more extensive the damage it can cause. For the Bio-Cyber hacking attack considered in this article, it is of vital importance for the attacker to maintain a natural appearance of the trigger sample containing the address details. If we use an unnatural DNA structure as a part of the hybrid attack it can be flagged as suspicious not only by the detection method proposed in this article, but also by the similar less sophisticated versions of this system proposed in previous works[24].

In this section we begin the discussion by evaluating the possible actions of an attacker to design a natural trigger sample. We follow this up by investigating the accuracy with which these trigger samples can be detected by a CNN. Finally, we describe the wet lab experiments that were used to produce the DNA with the address, in order to validate the potential of creating such a DNA sequence that is used as the trigger sample for our attack.

Trigger sample design. For this article we propose the use of *E. coli* plasmids that will encode the address of the attacker. *Escherichia coli* bacteria have been sufficiently studied in literature and their plasmids can be synthesized and modified with relative ease. Once the attacker identifies a suitable DNA structure, *E. coli* plas-

mids can be readily synthesized in various laboratories across the globe such as *EuroFins Genomics and Twist BioScience*[24]. In this section, we present the design of the plasmid DNAs that contains the Trojan payload that will maintain the original *E. coli* plasmids sequence. Specifically, we evaluate the use of DNA steganography (as described in the Methodology section) for injecting the address payload into an *E. coli* plasmid (host) DNA to maximize similarity between the resultant *inj* and host DNAs d_{host}.

This evaluation requires 1000 bps reads randomly sampled from the plasmid DNAs made available via *AddGene* repository. The sampling serves two purposes. First, it mimics the operation of a DNA-sequencer (e.g., Roche 454 FLX + [38]) that may be specifically targeted by the attacker. In this case, a higher number of DNA-reads produced by the sequencer (i.e., 700–1000 bps) will provide better cover for the Trojan address payload and, thus, increase the chances for the hybrid attack to be successful. Secondly, the sampling can significantly increase the amount of DNA-data used in the evaluation, where we draw 4356 reads from 716 *E. coli* plasmid DNAs stored in the AddGene repository.

Since the steganography technique has five key steps, the *encoding* step is fixed and cannot be varied, but the attacker is free to finetune the tagging, fragmentation, encoding, retention, and encryption steps. In Fig. 5 we show the impact of different parameter combinations, e.g. size of the fragment, number of retention positions, and value of the encryption keys.

Figure 5a depicts the relationship between the length of nucleotide tags and their availability. The tags mark the start and the end of the Trojan payload injections into a plasmid DNA. These tags that mark the start and the end of the Trojan payload are two potentially different nucleotide sequences of the same length. The sequences are selected in a manner that a host DNA is unlikely to include both tags separated by nucleotides. Note that the number of these nucleotides are obtained directly from the fragment size and the retention (i.e. retention of host nucleotides) parameters of the steganography technique. The results in Fig. 5a correspond to various values of these two parameters. From these results we learn that a predictable growth of tag availability is associated with the increase in tag length. As the number of all possible nucleotide sequences grows exponentially, it can overcome the number of unique sequences in genuine DNA reads for 4-nucleotide tags. We also realize that any further increase in the tag length (i.e., 5 and beyond) will make the number of unique sequences negligible, leaving the attacker with ample choice of nucleotide tags. The strength of this effect is such that it can be seen for all fragment sizes and retention values. As a result of this observation, we use a minimum 5-nucleotide tags for the remainder of this article as this is the lowest length that allows for the substantive tag availability.

In Fig. 5b we study the impact of the fragment size selection on the similarity between the host DNA before and after the injection of Trojan payload. This similarity is assessed by using Needleman-Wunsch (NW) scores (described in Methodology). The system is designed in such a way that the Needleman-Wunsch score grows as the similarity between the two DNAs increases. The value of this score is absolute maximum (i.e. MaxNW) when either the DNAs are identical, or the Trojan payload address is inserted into the host DNA naturally. Since due to tagging this is not possible we use the maximum (i.e. the NW score between host the DNA and itself) value to benchmark the score reduction due to the payload injection. Furthermore, in order to ensure the optimal payload injection, the steganography uses MaxNW technique (described in Methodology). To demonstrate the efficiency of this technique, Fig. 5b presents a comparison of performance with two alternative techniques, i.e., Random and MinNW. Random technique injects the payload at an arbitrary position through uniform distribution, whereas MinNW is a dynamic programming technique that seeks the worst possible injection position for a payload. This means that MinNW is a mirror-image of MaxNW which can minimize the score between the host and injected DNAs. This phenomenon is reflected in Fig. 5b, where MaxNW results in significantly lower score reduction compared to MinNW, whereas the score reductions by Random technique lies approximately in the middle of those produced by MaxNW and MinNW. From this we conclude that the MaxNW and MinNW techniques can show the whole range of score reductions that may occur due to payload injections. This also reaffirms that MaxNW is the best technique amongst all three possible techniques. In addition, a closer inspection of the results for the MaxNW technique also clarifies the impact of payload fragmentation. We realize that using a larger fragment size in the host DNA can effectively reduce the similarity between the host and injected DNAs.

Next in Fig. 5a,b we investigate the impact of different *retention* as well as *encryption* choices of the attacker. The results are presented only for MaxNW which is the optimal injection technique we have selected. For both the retention of host nucleotides or payload encryption, we realize that there is no significant effect on the NW score. In particular Fig. 6a shows no change in the NW score reduction can be attributed to different retention numbers for various fragment sizes for payload encrypted with a key equal to 50. Figure 6b shows similar results, where payload fragments of 1 and 5 characters are injected using 1 and 5 retention numbers. For this case, we also observe no change in the NW scores when encription keys are utilized. Based on these results, we can conclude that neither *retention* nor *encryption* are likely to disguise the trigger sample. Although we note that neither of these two steps can help the payload appear more naturally, however they still remain an essential part of the steganography process. This is because these steps play a key role in maintaining the anonymity of the attacker as they are designed to protect the payload (i.e. network address and port number), which may identify the attacker. For the case when a trigger sample is identified, the retraction of the payload will require knowledge of both the *retention number* and the *encryption key* used by the attacker.

DNN detection accuracy. The DNA sequences of 716 *E. coli* plasmid DNAs are collected from the AddGene repository using web scraping. The Selenium Webdriver is used to crawl and collect the pages containing the DNA sequences. The page was parsed using a python script to get and store the DNA sequences. In total, 4356 reads with read size of 1000 were drawn from the DNA sequences.

Although the natural appearance of the trigger sample is necessary to disguise the hybrid attack and avoid detection by less sophisticated methods (e.g. NW comparison with known DNAs), the Trojan payload address

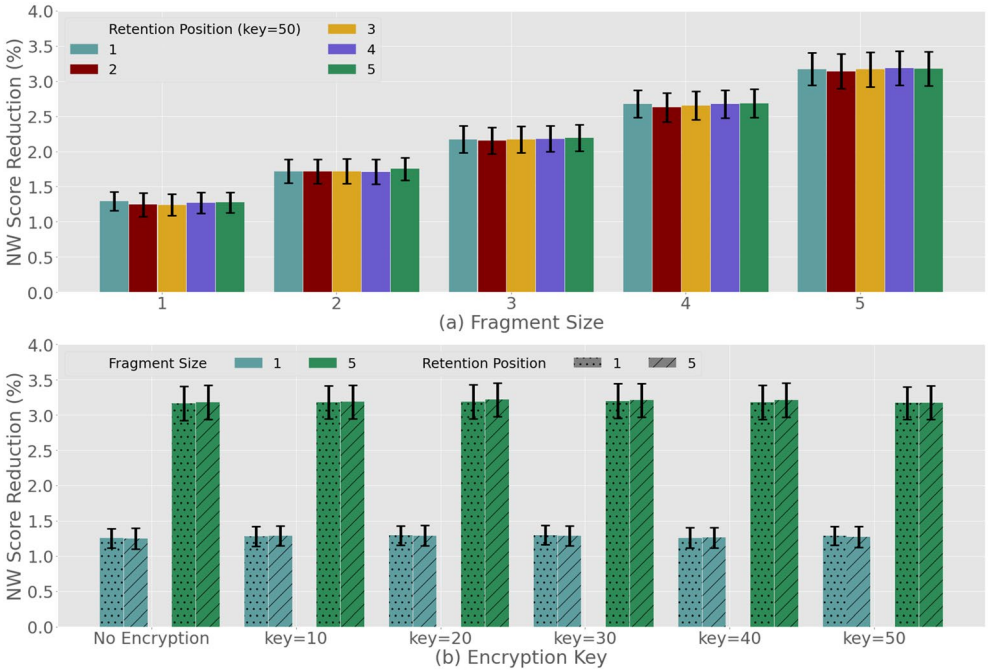


Figure 6. Trigger Sample Design, the use of DNA-Steganography: (**a**) retention of host nucleotides; (**b**) payload encryption.

injection may still be discoverable with the help of other techniques. In this section, we will explore this by evaluating the detection of trigger samples using a state-of-the-art Deep Learning approach. From 4356 reads, where the read size is 1000 nucleotide bases, 1000 reads are picked randomly. These are clean or natural samples. 1000 web addresses were considered for creating the malicious samples. A web address is rearranged, fragmented and then expressed as nucleotides and then inserted into the clean sample using the substitution method and using the technique described in the "Injection Methods" subsection in "Methods" section based on best NW Scores. The whole process is repeated 10 times to create 10 datasets. The idea is to execute 10 CNN model training and evaluations and take the average. However, if we consider 5 different fragmentations, 5 retentions and 5 encryptions then for all combinations it will be a huge number of evaluations. Furthermore, for the hyperparameter optimization there will be more scenarios to consider. Therefore, for every scenario we combine all 10 clean datasets into one and 10 malicious datasets into one. From 10,000 clean and 10,000 malicious data we pick 7500 (75%) data randomly from each as training dataset and remaining 2500 (25%) data from each as testing dataset. For training the model using the training dataset, we take a batch size of 100 at a time. We run the training for 3000 epochs with a learning rate of 0.001. In each epoch, 10 percent of the training data was used for validation, so that we can examine the learning (accuracy and loss comparison for training and validation over epochs) to avoid overfitting. Moreover, after every layer a dropout layer is also added to avoid the overfitting. Early stop monitoring is also used to avoid unnecessary continuation of the model training if it is reached to its optimal accuracy. The trained model is then evaluated by the corresponding test dataset. We achieve this by investigating the performance of a 1-Dimensional Convolutional Neural Networks (CNN). The results in Fig. 7a,b summarize the performance of various CNNs topologies with respect to the four hyper-parameters considered in this article. This includes, (i) the number of *hidden layers* (1 and 2), (ii) the sizes of the *filter* (4, 8 and 16), (iii) size of the *kernel* (3, 5 and 8), and (iv) size of the *maxpool* (2 and 4) used in the network. The results are then obtained for trigger samples obtained from natural DNA using 0-retention and no payload encryption. This means that we can establish a baseline predictive capacity of CNNs and determine the most suitable network topology. This suitable topology is then further tested to evaluate the ability to cope with additional uncertainties introduced by nucleotide retention and payload encryption.

For this purpose, we simulated 180 scenarios for 36 combinations of hyper parameters and for 5 different fragment sizes, with no retention and no encryption. We obtain the best accuracy (99.9–100%) for all 5 fragment sizes when we have 1 hidden layer, kernel size 16, 16 filters and 4×4 max pool size (Fig. 7a). Similarly, we obtain the best accuracy for the case we have an additional layer (2 hidden layers), 16 filters, kernel size 5 and 4×4 max pool (Fig. 7b). These features are mainly learned by the kernel, so larger kernels and higher number of filters result in achieving the best accuracy. However, in this article we prefer to use a smaller number of required hidden layers to increase the execution time performance. Therefore, for the rest of the experiment we consider the CNN topology with 1 hidden layer, kernel size 16,16 filters and 4×4 max pool.

Next in Fig. 8 we analyze the impact of the fragment size, retention values and encryption on the Trojan address detection. In particular, Fig. 8a presents the detection accuracy for the highest and lowest fragment size values (1 and 5), and all the retention numbers (1–5), when no encryptions are applied. We made an assumption

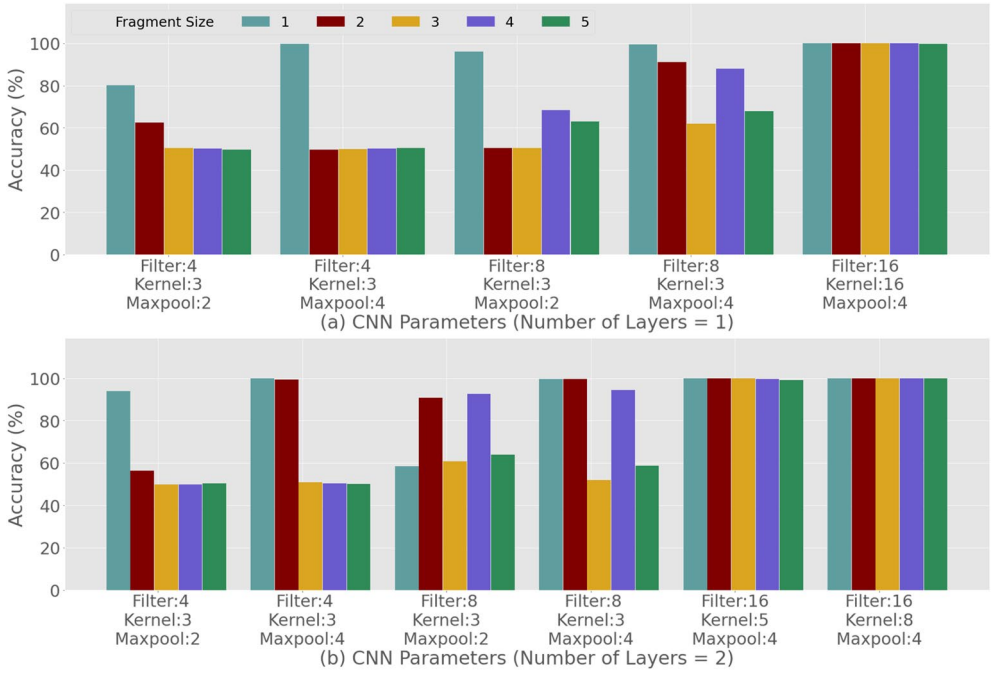


Figure 7. DNN-based detection of trigger samples amongst genuine *E. coli* plasmids: hyper-parameter optimization (no encryption or retention) using (**a**) 1 and (**b**) 2 hidden layers.

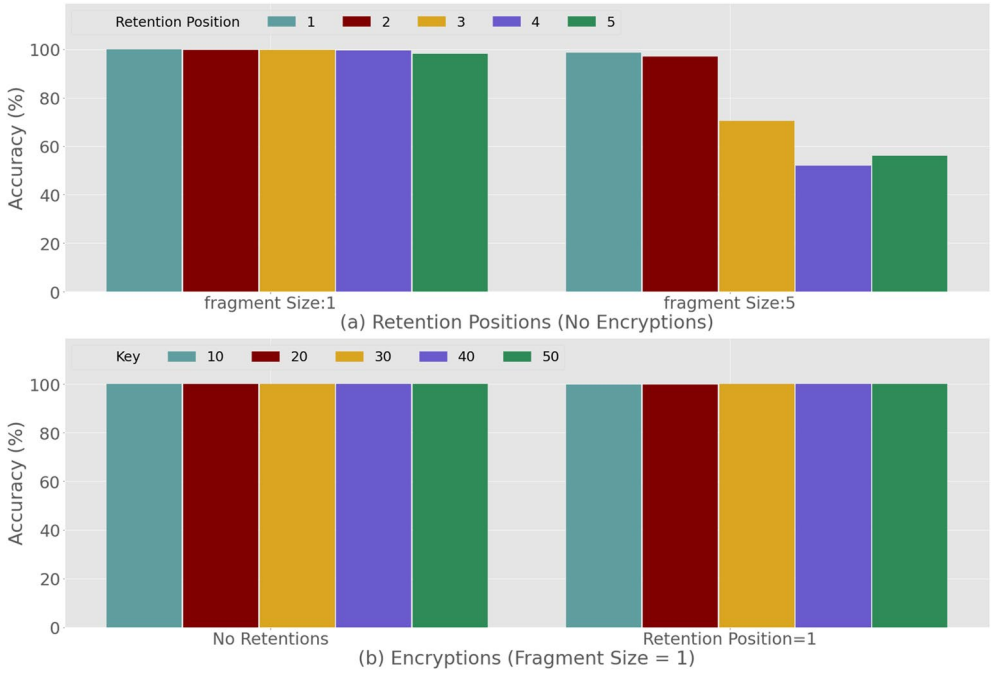


Figure 8. DNN-based detection of trigger samples amongst genuine *E. coli* plasmids: the impact of nucleotide retention (**a**) without encryption and (**b**) with encryption and with prior knowledge of the encryption key.

that if we split the payload into an increasing number of fragments it will be relatively easy to escape the detection. In such a case it will be comparably difficult to locate the complete Trojan payload address and, therefore, be relatively harder to make sense out of a more tinier part of the payload. Furthermore, as shown and explained in the previous section (Fig. 6a,b), the DNA sequences remain much more natural for smaller fragment sizes. Based on this knowledge, a potential hacker might prefer to choose a smaller fragment size. However in reality this approach will leave more tags as low fragment size translates to increase in number of tags. Therefore, this

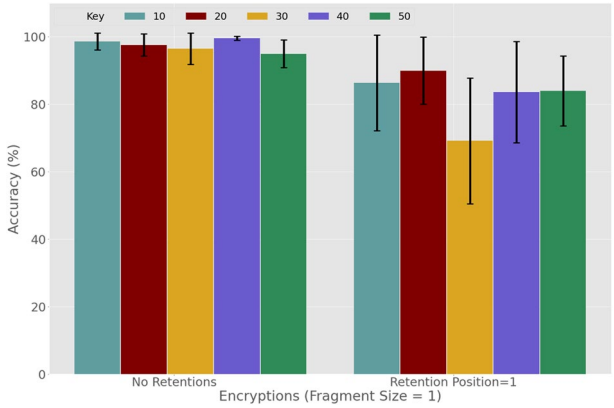


Figure 9. DNN-based detection of trigger samples amongst genuine *E. coli* plasmids: the impact of nucleotide retention, no knowledge of the encryption key.

approach can support the CNN model, which can learn from the tag patterns and the result in Fig. 8a illustrates this.

On the other hand, in a real world scenario it will be a significant challenge to design an optimal model which can account for many variations of tags. Interestingly, we observe that for higher fragment sizes, the accuracies deteriorate very slightly until there is a higher retention number as well (Fig. 8a). This indicates that the model proposed in the article does not completely rely on learning the tag patterns. Furthermore, the higher retention number means more number of nucleotides (from the original sequence inside the tags) which will result in more variations and harder detection. However, we note that for fragment size 1 the accuracies are very high for all retention numbers. Overall, the accuracies start to deteriorate significantly for the higher fragment sizes with higher retention numbers (Fig. 8a). To analyze the impact of encryption on the Trojan address payload detection, we consider fragment size 1 with no retention and retention size 1 as we obtain the best accuracy for these options. We apply encryptions with various key values ($key \epsilon \{10, 20, 30, 40, 50\}$). In Fig. 8b, the results show that there is no significant change in accuracy when applying various encryption keys. Please note that both the training and test data are using the same key value for encryption.

We will now further analyze the impact of encryption in detection. In Fig. 9 we present the detection accuracies where the rojan payload address in the test data is encrypted with a different key. The model is trained with a particular key which is tested by all the data encrypted by the remaining keys. For example, the model trained by the data encrypted using key = 10 will be tested by all the test data that are encrypted by other keys, i.e. keys = {20, 30, 40, 50}. Similarly, the model for key value 20 will be tested by all the test data encrypted by the keys = {10, 30, 40, 50}. In Fig. 9 we plot the average accuracy against the different key values used for training the model. From this result, we conclude that a higher accuracy can be achieved for encrypted payloads without retention even if the key is unknown. However, the accuracy will deteriorate if we apply retention along with encryption. This is because the higher retention will result in the DNA sequence having a more natural pattern, which makes it more difficult to detect.

Wet lab experiments. In the previous sections of this article, we have described how we can disguise the address payload for a Trojan attack to make the payload insert indistinguishable compared to a natural DNA sequence. Furthermore, applying encryption and steganography techniques will make it harder to detect the hybrid Trojan attack. However, it is also important to address how practical it is to synthesize such a DNA sequence. In our wet-lab, we constructed the Trojan payload sequences both without and with encryption and steganography (Figs. A.1 and A.2) via commercial gene synthesis with ease. These sequences were prepared and received already ligated into bacterial plasmid vector. These plasmids, pNOSTEG and pSTEG, were easily cloned into *E. coli* cells, propagated and purified in abundance (Fig. A.3). The Trojan payloads in both plasmids were both DNA sequenced completely and with 100% accuracy, with a sample chromatogram from pNOSTEG shown in Fig. A.4. We can assume that constructing natural DNA sequences will be easier and more achievable compared to synthesizing artificial DNA with unnatural sequences, due to possible runs and repeats of DNA bases that may cause problems in the synthesis reaction. As a result, there will be a need to construct a DNA that can allow multiple fragment inserts with the target information of the IP address and port number of the remote hacker's machine. With various techniques emerging for generating, producing or inserting multiple DNA sequences into carrier or expression systems, e.g., in-fusion cloning, gene assembly or multiple fragment cloning, hackers can bypass any gene synthesis issues by using a combination of these techniques to generate their final Trojan attack sequence. As such, our work presents valuable detection against very feasible attack scenarios.

Data availability

All data used in the manuscript are available in the Addgene repository (https://www.addgene.org/), where the DNA sequences of type plasmid of *E. coli* bacteria are collected for our experiments using web scraping. This data is also available as a supplementary document (all_plasmid_dna.txt). The Programming code developed to

conduct the experiments (also the scripts for the data collection from Addgene) is freely available in the publicly available git repository at the following URL: https://github.com/sibleeislam/trojan-malware-in-bio-cyber-attacks. For any further query related to data availability please contact using the email of the primary author (sibleeislam@gmail.com) of the manuscript.

Received: 25 February 2022; Accepted: 26 May 2022
Published online: 10 June 2022

References

1. Vijayvargiya, P. *et al.* Application of metagenomic shotgun sequencing to detect vector-borne pathogens in clinical blood samples. *PLoS ONE* **14**, e0222915 (2019).
2. Haiminen, N. *et al.* Food authentication from shotgun sequencing reads with an application on high protein powders. *NPJ Sci. Food* **3**, 1–11 (2019).
3. Akyildiz, I. F., Pierobon, M. & Balasubramaniam, S. An information theoretic framework to analyze molecular communication systems based on statistical mechanics. *Proc. IEEE* **107**, 7 (2019).
4. Unluturk, B. D., Balasubramaniam, S. & Akyildiz, I. F. The impact of social behavior on the attenuation and delay of bacterial nanonetworks. *IEEE Trans. Nanobiosci.* **15**(8), 959–969 (2016).
5. Laver, T. *et al.* Assessing the performance of the Oxford nanopore technologies MinION. *Biomol Detect Quantif* **3**, 1–8 (2015).
6. Yousefzai, R. & Bhimaraj, A. Misdiagnosis in the COVID-19 Era. *JACC: Case Rep.* **2**, 1614–1619 (2020).
7. Lim, J. T. *et al.* The costs of an expanded screening criteria for COVID-19: A modelling study. *Int. J. Infect. Dis.* **100**, 490–496 (2020).
8. Aitken, J. *et al.* Scalable and robust SARS-CoV-2 testing in an academic center. *Nat. Biotechnol.* **38**, 927–931 (2020).
9. Reuben, R. C., Danladi, M. M. A. & Pennap, G. R. Is the COVID-19 pandemic masking the deadlier Lassa fever epidemic in Nigeria?. *J. Clin. Virol.* **128**, 104434 (2020).
10. Capone, A. Simultaneous circulation of COVID-19 and flu in Italy: Potential combined effects on the risk of death?. *Int. J. Infect. Dis.* **99**, 393–396 (2020).
11. Hsih, W.-H. *et al.* Featuring COVID-19 cases via screening symptomatic patients with epidemiologic link during flu season in a medical center of central Taiwan. *J. Microbiol. Immunol. Infect.* **53**, 459–466 (2020).
12. San Millan, A. Evolution of plasmid-mediated antibiotic resistance in the clinical context. *Trends Microbiol.* **26**, 978–985 (2018).
13. Blackwell, G. A., Doughty, E. L. & Moran, R. A. Evolution and dissemination of L and M plasmid lineages carrying antibiotic resistance genes in diverse Gram-negative bacteria. *Plasmid* **113**, 102528 (2021).
14. Health Service Executive (HSE) Ireland. About CervicalCheck: Ireland's national cervical screening programme. Available On-Line, Retrieved from 13 Aug 2020 https://www2.hse.ie/screening-and-vaccinations/cervical-screening/about-cervicalcheck/about.html
15. Ney, P. *et al.* Computer security, privacy, and DNA sequencing: Compromising computers with synthesized DNA, privacy leaks, and more. USENIX Security 17, 2017.
16. Rabadi, D. & Teo, S. G. Advanced windows methods on malware detection and classification. In *Annual Computer Security Applications Conference* (2020).
17. Kouliaridis, V., Kambourakis, G. & Peng, T. Feature Importance in android malware detection. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2020).
18. Guo, W. *et al.* Towards inspecting and eliminating Trojan backdoors in deep neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)* (2020).
19. Pan, Z. & Mishra, P. Automated test generation for hardware Trojan detection using reinforcement learning. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference* (2021).
20. Yasaei, R., Yu, S.-Y. & Al Faruque, M. A. GNN4TJ: Graph Neural networks for hardware Trojan detection at register transfer level. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2021).
21. Lyu, Y. & Mishra, P. Automated test generation for Trojan detection using delay-based side channel analysis. In 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2020).
22. Guo, S., Wang, J., Chen, Z., Li, Y. & Lu, Z. Securing IoT space via hardware Trojan detection. *IEEE Internet Things J.* **7**, 11115–11122 (2020).
23. Black, A., MacCannell, D. R., Sibley, T. R. & Bedford, T. T. recommendations for supporting open pathogen genomic analysis in public health. *Nat. Med.* **26**, 832–841 (2020).
24. Islam, M. S. *et al.* Genetic similarity of biological samples to counter bio-hacking of DNA-sequencing functionality. *Sci. Rep.* **9**, 1–9 (2019).
25. Sreekumari, P. Malware detection techniques based on deep learning. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)* (2020).
26. McDole, A., Abdelsalam, M., Gupta, M. & Mittal, S. *Analyzing CNN based behavioural malware detection techniques on cloud IaaS*. In *Lecture Notes in Computer Science*, 64–79 (2020).
27. Kimmel, J. C., Mcdole, A. D., Abdelsalam, M., Gupta, M. & Sandhu, R. Recurrent neural networks based online behavioural malware detection techniques for cloud infrastructure. *IEEE Access* **9**, 68066–68080 (2021).
28. Sharma, R., Rathor, V. S., Sharma, G. K. & Pattanaik, M. A new hardware Trojan detection technique using deep convolutional neural network. *Integration* **79**, 1–11 (2021).
29. Islam, M. S. *et al.* Trojan bio-hacking of DNA-sequencing pipeline. In *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication* (2019).
30. Hayward, S. L., Francis, D. M., Sis, M. J. & Kidambi, S. Ionic driven embedment of hyaluronic acid coated liposomes in polyelectrolyte multilayer films for local therapeutic delivery. *Sci. Rep.* **5**, 14683 (2015).
31. Pearson, W. R. Selecting the right similarity-scoring matrix. *Curr. Protoc. Bioinform.* **43**, 3–5 (2013).
32. Rivas, E. & Eddy, S. R. Parameterizing sequence alignment with an explicit evolutionary model. *BMC Bioinform.* **16**, 1–23 (2015).
33. Gunasekaran, H. *et al.* Analysis of DNA sequence classification using CNN and hybrid models. *Comput. Math. Methods Med.* **2021**, 1–12 (2021).
34. Ghosh, A. & Barman, S. Application of Euclidean distance measurement and principal component analysis for gene identification. *Gene* **583**, 112–120 (2016).
35. Liu, D.-W. *et al.* Automated detection of cancerous genomic sequences using genomic signal processing and machine learning. *Futur. Gener. Comput. Syst.* **98**, 233–237 (2019).
36. Weimer, D., Scholz-Reiter, B. & Shpitalni, M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Ann.* **65**, 417–420 (2016).
37. Zou, J. *et al.* A primer on deep learning in genomics. *Nat. Genet.* **51**, 12–18 (2018).
38. Yin, Z., Mancuso, J. J., Li, F. & Wong, S. T. C. Genomics-based cancer theranostics. *Cancer Theranostics* 9–20 (2014).

Acknowledgements
This publication came from research conducted with the financial support of Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland (Grant Number [16/RC/3835] - VistaMilk). Artwork on Figs. 2 and 4 of the article was created by the author of this article Mr. Islam using free Draw.io (https://www.diagrams.net/about) software and free icons available on the web. Artwork on Fig. 1 and 3 of the article was created by the authors of this article Dr. Ivanov and Mr. Islam using free Draw.io software and free icons available on the web.

Author contributions
Mr. M.S.I. is the primary author of the article. Mr. I. was responsible for developing the software code used to perform computational experiment, executing the experiments, analysing and interpreting the results presented in this article, writing the manuscript. Dr. S.I. was responsible for overseeing and directing computational experiments presented in this article. Specifically, Dr. I. contributed to the development of the proposed steganography technique, where he proposed the dynamic programming technique for finding an optimal location for the payload for malicious activity to be injected into the host DNA. Dr. I. assisted Mr. I. in writing the manuscript. Dr. S.B. was the main scientific driver behind the experiments presented in the article. Due to his multidisciplinary background, Dr. B. identified the possibility for *E. coli* bacteria to be used as carriers of malicious DNA on-purposed engineered as part of a Trojan attack. That was the starting point for the research presented in the article. Subsequently, Dr. B. directed and oversaw the experiments conducted in this research. Dr. Lee Coffey planned and executed the wet lab experiments, including gene synthesis design, cloning and recombinant plasmid DNA purification. Dr. S.K. was responsible for providing expertise in methods for handling DNA based samples and background for DNA packaging/carrying. Ms. J.D. prepared the DNA samples for sequencing and carried out sequence analysis of the DNA fragments in order to verify sequence identity and fidelity. Dr. W.S. was the scientific driver behind the DNN analysis for the DNA strands with the injected code, as well as the development of the hacking scenarios. Dr. H.A. was responsible for the analysis of the data in the results section and in particular the analysis on performance based on variations in parameters.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-13700-5.

Correspondence and requests for materials should be addressed to M.S.I.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.