# Cyberbiosecurity: Remote DNA Injection Threat in Synthetic Biology

Dor Farbiash[1] and Rami Puzis[2]

[1]Jusidman Science Center for Youth at Ben-Gurion University of the Negev
[2]Cyber Security Research Center at Ben-Gurion University of the Negev
[3]Software and Information Systems Engineering Department at Ben-Gurion University of the Negev

***Abstract*—A weakness in the design of the Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA and its successor, the Harmonized Screening Protocol v2.0 enables these protocols to be circumvented using a generic obfuscation procedure. Insufficient cybersecurity mechanisms throughout the synthetic gene engineering pipeline allow a remote attacker to interfere with biological processes. Together, these weaknesses facilitate an end-to-end cyberbiological attack, in which a remote attacker may inject obfuscated pathogenic DNA into an online order of synthetic genes, using a malicious browser plugin. The attacker exploits residue Cas9 protein for deobfuscation of the adversarial sequences and transforms them into active pathogens. The end-to-end attack scenario stresses the need to harden the synthetic DNA supply chain with protections against cyberbiosecurity threats. We propose an improved screening protocol that implements the top homology principle and considers the possibility of in vivo gene editing.**

## I. Introduction

Synthetic biology is an emerging bioengineering technology that plays a significant role personalized medicine, the pharmaceutical manufacturing, etc.[1], [2]. Multiple kinds of "Hello World" organisms are readily available for those who want to develop their own biological systems [3]. Rapid development of biological systems is supported by large online libraries of genes [4], [5], [6], as well as integrated development environments (IDEs) and gene compilers for efficient gene coding [7]. Currently, the software stack used to develop synthetic genes is loosely secured (see Section III-B), allowing the injection of rogue genetic information into biological systems by a cybercriminal with an electronic foothold within an organization's premises.

DNA synthesis companies, which produce and ship the DNA sequences provided by their clients, are an important element of the growing synthetic biology market. Synthetic DNA is available in multiple ready to use forms. For example, such as a circular DNA molecule called plasmid (see Section II-A for more details). A synthesized plasmid can be inserted into an organism by following a simple biological protocol, after which, it can start producing proteins [8]. Many bioengineering tools are now easily accessible by biohackers and do it yourself (DIY) biology enthusiasts. Online interaction between bioengineers and DNA synthesis companies serve as an additional attack vector, through which, rogue genetic information can be injected into a biological system.

The products of biological systems may be extremely dangerous substances, such as toxins or synthetic viruses [9], [10]. As a case in point, just a few weeks after the beginning of the COVID-19 pandemic, researchers reconstructed the virus using a synthetic genome [11]. The dual use of synthetic biology as a powerful technology for the benefit of mankind and as a potential weapon is a long-standing issue [12]. The dangers of synthetic biology are many, and they require rigorous security controls [13], [14]. One such control is the U.S. Department of Health and Human Services' (HSS) Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA (HHS guidelines) [15] for short (see Section II-C). Unfortunately, these and similar guidelines have not been adapted to reflect recent developments in synthetic biology and cyberwarfare. Biosecurity researchers agree that an improved DNA screening methodology is required to prevent bioterrorists and careless enthusiasts from generating dangerous substances in their labs [12], [16].

Understanding the impact of cyber threats on biosecurity is extremely important. Cyber attacks are considered a potential threat to the security and privacy of genomic data [17], [18] and the analysis of genetic material [19]. Recent biodefense report [12] mentions cyber threats with respect to identifying potential targets and the development of bioweapons. Diggans and Leproust [20] highlight the value of cybersecurity methods applied in the domain of biosecurity. However, the potential of delivering a harmful biological agent through cyberspace has not yet been considered. It is currently believed that a criminal needs to have physical contact with the substance to produce and deliver it.

*In this paper, we discuss a cyber attack in which a remote attacker*[1] *tricks a victim into producing a dangerous substance in the victim's lab, without the victim's knowledge or physical interaction between the attacker and the lab components.* We refer to this threat as Remote DNA Injection (RDI). This attack exploit insufficient integrity controls at the software level, weakness of the HHS guidelines at the biosecurity level, and residue Cas9 protein (see Section II-A) at the biological protocol level. The main findings related to each vulnerability

[1]In some cyber security literature the term *remote attacker* refers to an attacker with network access to the victim system but without electronic foothold. Here and in the rest of the paper, we refer to an attacker without physical contact to the victim system. Network vulnerabilities and initial access techniques are out of the scope of this paper.

level are presented below.

1. Software: Insufficient cybersecurity controls allow man-in-the-browser to inject arbitrary DNA strings into an online order of synthetic genes (Sections III-B and III-C). Simple mitigation steps, mentioned in Section VI, may significantly reduce the threat of DNA injection.
2. Biosecurity: (a) A generic weakness in DNA screening guidelines advised by HHS permits an adversary to avoid detection by obfuscating the malicious DNA (Sections III-D). (b) The proposed Gene Edit Distance (Section IV) can efficiently detect sequences that can be decoded into malicious DNA within living cells. (c) A benchmark dataset of obfuscated DNA sequences demonstrates the superiority of Gene Edit Distance over rigorous implementation of the HHS guidelines (Section V).
3. Biological protocol: The residue Cas9 protein, when using CRISPR protocols, can be exploited so as to deobfuscate the malicious DNA within the host cells. We discuss possible changes in biosafety recommendations that can reduce the threat of malicious Cas9 exploitation and suggest future research directions in the detection and mitigation of vulnerabilities in biological protocols.

Although, simpler attacks exist where an attacker with electronic foothold within the victim's computer may manipulate biological experiments, we choose to discuss an attack that highlights all three vulnerability levels mentioned above. Our purpose is demonstrating the role that cybersecurity know how can play in hardening the bioengineering pipeline.

## II. Background

This section provides the most important information required to describe the potential cyberbiological threats and their mitigation in the following sections. Readers familiar with the basics of protein synthesis and the CRISPR-Cas system are encouraged to skip Section II-A. Important terms used later in the paper are highlighted in **bold** face.

### A. An introduction to DNA editing for cybersecurity experts

Genetic information in living cells is encoded in sequences of **nucleotides** called DNA. Nucleotides are commonly denoted by four letters, C,G,A, and T, corresponding four nucleobases, cytosine, guanine, adenine, and thymine, respectively. In a double-stranded DNA (**dsDNA**) molecule, nucleobases on the opposite strands are bound together, C with G and A with T. Thus, dsDNA is sometimes regarded as a sequence of **base pairs (bp)**. The fees charged by gene synthesis companies for synthetic DNA orders are typically based on the number of base pairs. The orders are commonly delivered in the form of cyclic dsDNA molecules called **plasmids** which are very stable and can be replicated within a living cell. The plasmid price can be as low as five cents/bp. Provided a plsamid one may employ a **sequencing** procedure to get the string representation of the DNA molecule. Companies that produce synthetic DNA usually also provide sequencing services.

A complement of a DNA sequence is formed by replacing every occurrence character with its pair ($C \leftrightarrow G, A \leftrightarrow T$). **A reverse complement** of a DNA sequence, an operation commonly used in gene design, is formed by reversing the complement of a DNA sequences. For example, the complement of GGCA is CCGT and its reverse complement is TGCC.

Among other cell functions, DNA encodes the proteins. First, an RNA sequence is generated from a DNA region surrounded by special sequences called **promoters** and **terminators**. An RNA molecule contains the same information as the respective DNA, but it is short-lived. In computer terms it can be considered as the volatile operative memory, whereas DNA would be considered persistent storage. RNA containing special sequences called **ribosome binding sites**, can be transformed by a ribosome into a sequence of amino acids. Every three nucleotides form one **amino acid**, but different triplets may form the same amino acid. Translating amino acids back to triplets of nucleotides is called **reverse translation**. The choice of the optimal triplets depends on the organism in which the DNA is expressed. There are 20 amino acids. Short sequences of amino acids are called **peptides**, while long sequences are called **proteins**. Both implement various functions within a living cell.

The clustered regularly interspaced short palindromic repeats (**CRISPR**) complex is a part of the bacterial immune system that was adapted by bioengineers to perform precise DNA editing in live biological systems. The most common DNA editing system consists of a **Cas9** protein and a **guide RNA sequence (gRNA)**. In this article we will use the term CRISPR to the refer to the DNA system consisting of Cas9 and gRNA. The Cas9 protein performs a cut in a dsDNA molecule at special locations called protospacer adjacent motifs (**PAMs**). The gRNA contains a short replica of the region following the PAM that needs to be cut by Cas9. In computer terms, gRNAs can be regarded as pointers in an associative memory. For the gRNA creation, the DNA should contain a promoter, a copy of the **gRNA target site**, and a terminator, collectively referred to as a **gRNA scaffold**.

A dsDNA that was cut by a CRISPR can repair itself. Such a repair process is error prone and can produce mutations at the cut point. If such a mutation results in producing a different amino acid during protein formation, the protein may become non-functional (a.k.a. a gene knockout). Precise repairs of the cut DNA can be performed using a process known as **homology directed repair (HDR)**. To activate HDR, the cell should contain a DNA sequence that repeats the sequence of nucleotides to the left and right of the cut point (left and right arms of the HDR template respectively) and a small number of nucleotides that may be inserted between them at the cut point. HDR can also correct a few small mutations close to the cut point. Using CRISPR and HDR it is possible to remove and replace large portions of DNA [21], [22].

### B. Sequence alignment

Next we discuss sequence alignment, which plays a significant role in bioengineering and in biosecurity. The **Basic Local Alignment Search Tool** (`BLAST`) [23] is the first of a long series of algorithms developed for aligning sequences of nucleotides or amino acids. `BLAST` algorithms are optimized for speed and searching large databases of sequences. Let $q$

denote a **query sequence** and $t$ denote a subject sequence (also called a **target sequence**). Let $q[i]$ denote the $i$'th character in $q$. BLAST operates by matching n-grams, short sequences of letters (words), and extending these matches to form local **alignments** between the sequences.

Provided a query sequence, BLAST returns a set of target sequences similar to the query sequence and a set of alignments (also called ranges) for each target sequence. Let $\mathcal{A}_{q,t}$ be a set of such alignments between $q$ and $t$ found by BLAST. Every alignment in $\alpha \in \mathcal{A}_{q,t}$ maps a range of character positions (a substring) in $q$ to a range of character positions (a substring) in $t$, such that any two successfully aligned character positions $i > j$, $\alpha(i) > \alpha(j)$. Although, $\alpha$ is not a function, we use the terms domain ($dom(\alpha)$) and image ($img(\alpha) = a(dom(\alpha))$) to denote the respective substrings. Let $\alpha^{-1}$ denote the inverse alignment, such that $dom(\alpha^{-1}) = img(\alpha)$ and $img(\alpha^{-1}) = dom(\alpha)$.

The score of an alignment is computed based on the number of **matched** characters ($M = |\{i : q[i] = t[\alpha(i)]\}|$); the number of **mismatched** characters ($MM = |\{i : q[i] \neq t[\alpha(i)]\}|$); the number of **gaps opened** in both the query and the target sequences $G = |\{i : \alpha(i-1) \neq \perp \wedge \alpha(i) = \perp\}| + |\{i : \alpha^{-1}(i-1) \neq \perp \wedge \alpha^{-1}(i) = \perp\}|$; and the total **extent** of the gaps ($GX = |\{i : \alpha(i) = \perp\}| + |\{i : \alpha^{-1}(i) = \perp\}|$), where $\perp$ means that the argument character is not aligned. There is a reward ($rm$) for every matching character and penalties for mismatching characters ($pmm$), gap opening ($pgo$), and gap extension ($pgx$). The reward and penalties are configurable. The score of an alignment is:

$$Score = rm \cdot M - pmm \cdot MM - pgo \cdot G - pgx \cdot GX$$

The fraction of $q$ characters successfully aligned is called **query coverage**: $QC(\alpha) = \frac{MM}{|q|}$. The **percent identity** is computed from the sizes of the domain, the image, and the number of successfully mapped characters: $PI(\alpha) = 2 \cdot MM/(|dom(\alpha)| + |img(\alpha)|)$

*Example 1:* Consider the following alignment: $\alpha = \{(1,1),(2,2),(3,\perp),(4,3),(\perp,4),(\perp,5),(5,6),(6,7)\}$ .

$$\begin{array}{rl} dom(\alpha) & \texttt{1234 56} \\ q = & \texttt{TAGT--CA} \\ \alpha = & \texttt{|| ||} \\ t = & \texttt{TA-CGGCA} \\ img(\alpha) & \texttt{12 34567} \end{array} \quad (1)$$

The first two characters and the last two characters are identical, the fourth is a mismatch, and there are three gaps. The query coverage of $\alpha$ is $QC(\alpha) = 4/6$, the percent identity is $PI(\alpha) = \frac{2 \cdot 4}{6+7}$, and the fraction of gaps is $Gaps(\alpha) = 3/8$.

### C. State-of-the-art in synthetic DNA order screening

Some DNA sequences may encode extremely dangerous products, such as toxic peptides. The Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA (**HHS guidelines**), published by the United States Department of Health and Human Services, suggests methods to minimize the risk of unauthorized distribution of select agents [15]. Highly related to the HHS guidelines are the Harmonized Screening Protocol v2.0 (HSP), employed by the International Gene Synthesis Consortium (IGSC) [24], and the International Association Synthetic Biology (IASB) Code of Conduct for Best Practices in Gene Synthesis [25].

The HHS guidelines, IASB Code of Conduct, and HSP outline standards and practices are aimed at preventing the misuse of synthetic genes. They define procedures for customer screening and synthetic gene order screening for the presence of possible toxins, pathogens, and other biological agents that pose a significant threat to public health and safety, which are collectively referred to as **sequences of concern (SoC)**. US regulation also defines items on the Commerce Control List as sequences of concern.

While the HHS guidelines recommends screening specifically for the presence of sequences unique to sequences of concern, the HSP recommends identifying sequences derived from or encoding a sequence of concern. It is generally advised to use a sequence alignment tool, such as BLAST (the Basic Local Alignment Search Tool) [23], to compare gene orders with known sequences in the GenBank database [6]. HHS guidelines recommends the ***Best Match*** approach to determine the legitimacy of an order based on the classification of the most similar sequence in the database. Specifically, every fragment of **200bp** in the order is searched within the database using sequence alignment. If the *Best Match*of any fragment is a sequence of concern, the order is deemed a **hit**, and it is forwarded for further investigation. Similar to computer security, the false hit rate is the greatest concern during screening.

**GenoTHREAT** [26] is a software[2] that implements the HHS guidelines. The query sequence is partitioned into 200bp fragments. According to GenoTHREAT an alignment is a *Best Match*if $QC = 100\%$ and $PI$ is the maximal over all other alignments of the query sequence. Note that more than one alignment may match these conditions. Further, if either the beginning or end of a 200bp fragment is aligned to a SoC and $QC > 50\%$, then the fragment is extended in order to identify a possible alignment of the preceding or the following fragment, with a SoC.

### D. HHS guidelines criticism and responses

There are multiple concerns regarding the HHS guidelines [27]. One such concern is the possibility of the assembly of SoCusing small synthetic DNA fragments and new bioengineering tools. For example, using Gibson assembly, oligos (short DNA molecules) may be assembled to construct larger fragments [28]. Since the HHS guidelines does not address oligonucleotide screening, one might order small DNA fragments, which are not screened [16] and can be assembled to create pathogens.

The HHS guidelines does not specify a database to use for screening, but it does provide the GenBank as an example of such a database. The lack of a formal database of SoC is a concern, since it may lead to inconsistent screening protocols between companies, false positives due to housekeeping genes shared between pathogenic and non-pathogenic organisms,

[2]Initially released as open source, but was retracted.

and increased cost of overall screening [16]. Housekeeping genes are required for basic cellular functions but do not produce toxins or other dangerous products. Nevertheless, housekeeping genes have long been known to cause false hits [29]. A curated database of SoC may reduce the cost of sequence screening [30], [31]. Other concerns are the possibility of poisoning public databases with adversarial sequences that have been misclassified as benign, mistakenly or with malicious intent [18].

The HHS guidelines recommends screening all sequences ordered, regardless of their length. Some argue that screening DNA fragments that are shorter than 200bp may lead to false positives and increased cost [26], while others argue that the 200bp cutoff is not scientifically justifiable [32]. We provide justification for a 60bp cutoff in Section III-D.

## III. The RDI threat

In this section we describe the attacker model, the general course of an RDI attack, and weaknesses along the bioengineering pipeline facilitating the attack. Then we deep dive into a DNA obfuscation method allowing to circumvent *Best Match*-based DNA screening. The widely available DIY biology instrumentation sufficient to become a victim of this attack is listed in appendix.

### A. Attack model and assumptions

In the following discussion, we use the names, Alice, Bob, and Eve, to represent the interacting parties. Let Alice be a do it yourself (DIY) biology enthusiast or a small bioengineering company that develops their own DNA sequences or combines existing genes to produce fuel, medical components, or resilient plants. We assume that Alice does not use her own facilities to produce the DNA but prefers ordering synthetic DNA strands from gene synthesis companies. Let Bob be a gene synthesis company that provides its services to Alice. The attacker, Eve, is a cybercriminal who wants to trick Alice into producing dangerous biological components.

Current DNA editors and common DNA file formats do not support electronic signatures or other methods to protect the integrity of the DNA sequences that Alice develops (see Section III-B) We also assume that Alice's cybersecurity awareness level is similar to the awareness level of most Internet users. Alice is driven by productivity and ease of use rather than by cybersecurity considerations.

Next, assume, for example, that Alice performs a series of genetic manipulations to E. coli (a commonly used bacteria for biological experiments). During some of these manipulations Alice employs the CRISPR system. Note that it is common practice to keep a Cas9 coding gene in a cell after it is first introduced. Alice, orders a custom synthetic gene that she developed from Bob and applies it either before or after using the CRISPR system. We assume that Bob, which is a large company with cybersecurity and biosecurity departments that adheres to all relevant regulations and guidelines, including screening DNA sequence orders for known pathogens [24], [15], [33]. We also assume that Bob performs rigorous sequencing of the synthesized DNA to detect any errors in the production, as most gene synthesis companies do.

Alice may perform sequencing of the DNA received from Bob at her own facilities or use a third party service. Due to low resources, Alice is likely to trust Bob's quality control and prefer to avoid additional expenses associated with sequencing or may not have the required expertise to do so. In either case, the sequencing results are inspected by Alice using the same computers she used to design the DNA.

In a possible scenario, an evil Eve possesses the resources of an average individual and has an intermediate sophistication level according to STIX Core Concepts [34]. Specifically, Eve understands HTML and common scripting languages, can build or modify a website, and execute the man-in-the-browser attack technique [35]. She is, however, unable to penetrate Bob's security premises. Eve targets DIY biology enthusiasts and small bioengineering companies, such as Alice. It is not important whether Eve targets Alice directly, sprays the attack throughout the biohacker community, or selects her target in a different way. Eve's motives may also span the entire spectrum of threat actor motivations [36]. Her immediate objective is *remotely producing toxins, pathogens, or other sequences of concern (SoC) in the victim's lab, without physical interaction with the lab components.*

Eve's reconnaissance and initial access techniques don't play a significant role as long as Alice's computer/s is/are subverted by Eve's malware. Using her malware, Eve is able to manipulate the DNA sequences developed by Alice in the design stage; intercept and replace DNA orders sent by Alice to Bob, as well as sequencing reports sent by Bob to Alice; replace DNA files or plant malicious DNA sequences within existing files; and more.

### B. Security weaknesses in the bioengineering workflow

Most bioengineers are well trained in biosafety protocols and aware of biosecurity screening and the dangers of engineering pathogens. However, like most computer users, bioengineers and DIY biologists can fall victim to social engineering or any initial penetration attack vector (Figure 1.a) Below we highlight the relevant weaknesses in software, biosecurity screening, and biological protocols.

*a) Software:* The first line of defense should have been provided by the integrated development environments (IDEs) for DNA coding. IDEs provide the ability to create and edit DNA sequences. We inspected the electronic integrity features provided by typical DNA IDEs, such as SnapGene, Serial Cloner, ApE (A plasmid Editor), and Genome Compiler, most of which use one or more common DNA file types, such as .genbank, .fasta, and .dna. Some of the file formats are binary, but they do not contain electronic signatures or other means of integrity protection (Figure 1.b). This allows a malicious attacker to change the sequences within DNA files, without the user's consent.

Most communication with gene synthesis companies, including gene orders, is performed through a company's website or email. All synthetic gene orders are validated prior to purchase and during production. Unfortunately, most validation reports are delivered through the same channel, presumably, in the case of an attack, controlled by the attacker, unencrypted and without electronic signatures (Figure 1c).

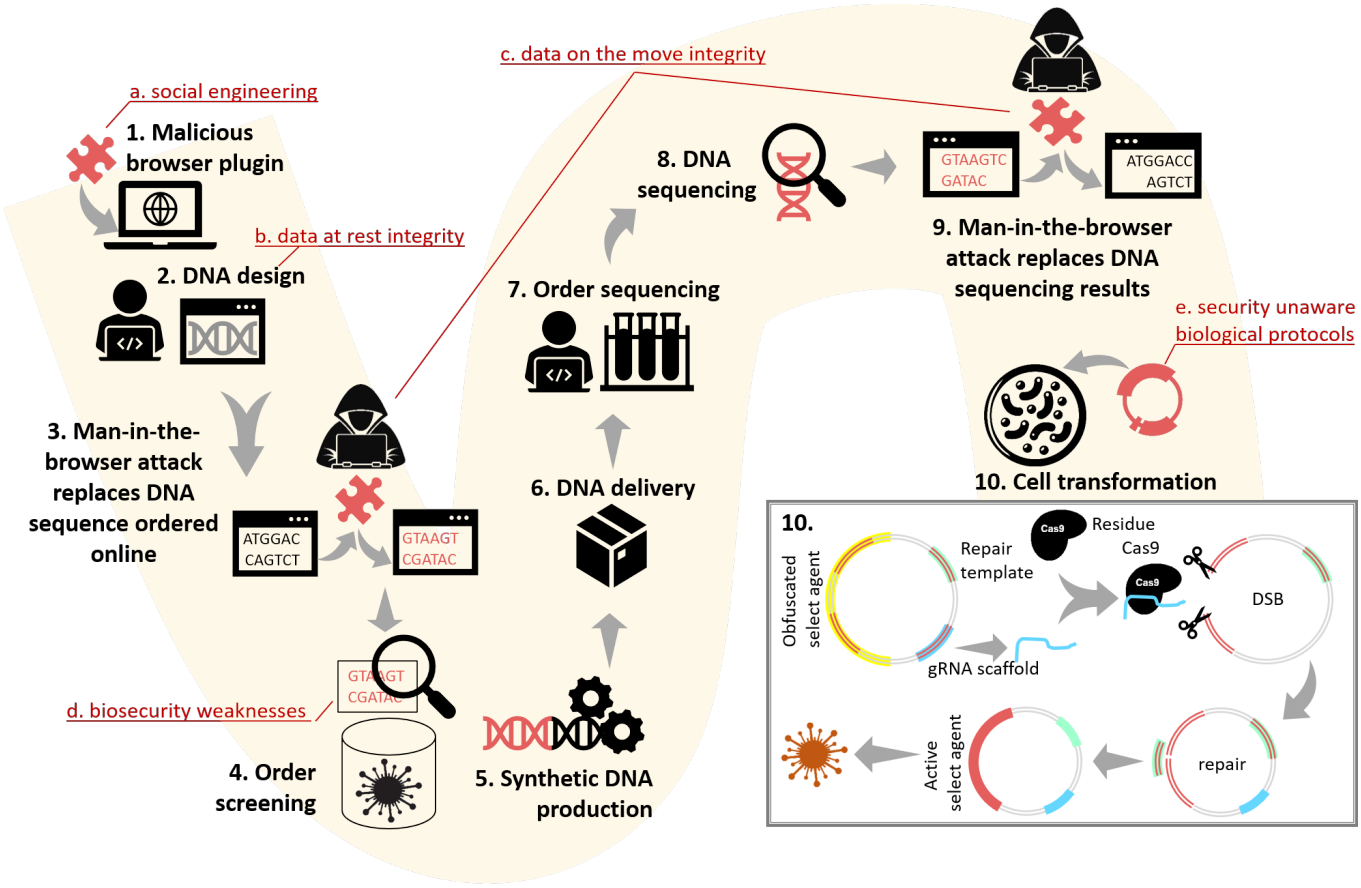


Fig. 1. The course of a RDI attack performed by a remote cybercriminal. (1) Alice's (a bioengineer) computer/s has/ve been subverted (for example, through a Trojan browser plugin) without her knowledge. (2) Alice designs an experiment and the DNA sequences for cell transformation. (3) When Alice orders synthetic DNA online from Bob, a DNA synthesis company, the attacker replaces some of the sequences ordered with obfuscated select agents and the sequences required for their future deobfuscation. (4) Bob screens the order but fails to identify the malicious sequences due to the obfuscation. (5) Synthetic DNA is produced and (6) delivered to Alice. (7-8) Possible sequencing of the delivered order is performed, either by a third party or by Alice herself. (9) The sequencing results are inspected using a subverted computer. (10) When performing the planned transformation, Alice uses the malicious DNA, along with other sequences including those that produce Cas9. During the cell transformation, Cas9 protein is combined with gRNA from the malicious sequence to form the CRISPR system that is used for multiple double-strand breaks. Malicious DNA repair templates are used for HDR to produce a viable pathogenic DNA. Letters (a-e) indicate weaknesses in the bioengineering workflow.

*b) Biosecurity screening:* Synthetic gene orders are screened by biosecurity algorithms oriented at computational performance and zero false hits. While these objectives parallel common practice in cybersecurity, the screening operates much like signature-based anti-virus software, neglecting detection evasion techniques (Figure 1d).

*c) Biological protocols:* Finally, as mentioned at the beginning of this subsection, bioengineering best practice focuses on safety and efficiency. Most biological protocols today do not consider cyber threats or insider threats; more often than not, they rely on physical perimeter security when biosecurity is a concern (Figure 1.e). One example of such biological grade vulnerability is negligence of Cas9 expressing DNA after performing the intended cuts in the DNA. CRISPR best practice recommends discharging of gRNA used perform the cuts in order to avoid unintended cuts during the next experiment phases.[3] However, the Cas9 expressing DNA is usually left intact within the cells.

[3]See Section VI for additional discussion on controlling Cas9 functionality.

### C. The course of the attack

Here we describe a possible course of the RDI attack in general terms. When planning her attack, Eve examines the websites of typical gene synthesis companies to identify fields and variables containing the DNA strings submitted by a customer. Eve builds or adapts a browser plugin with some useful functionality. For example, it can be a plugin that adds an annotated schematic adequately visualizing the DNA sequence, in addition to a raw textual representation of a DNA in text fields and HTML. The plugin's malicious payload can manipulate the DNA with a malicious sequence during form submission and replace the malicious DNA with the one the user entered in all follow-up reports. See Figure 8 in the appendix for an example of such a plugin that was successfully tested with online synthetic gene orders.

In this case, Eve tricked Alice into downloading a Trojan horse in the form of a useful browser plugin.

During the course of the attack, Eve's plugin replaces some of the DNA sequence ordered with a malicious sequence (Figure 1.3). Since most gene synthesis companies charge by

the number of base pairs [37], Eve should not significantly increase the length of the sequence ordered, in order to avoid detection resulting from unexpected additional costs. We discuss the methods for constructing the malicious sequences in Section III-D. In order to avoid being detected, Eve should also use her malware to track Alice's communication with gene synthesis companies and replace any occurrences of the malicious DNA sequences with legitimate sequences supposedly ordered by Alice.

Bob screens Alice's order. However, if Bob employs currently available screening protocols [15], [27], [24], he will be unable to identify the obfuscated select agents that Eve inserted into Alice's order. The obfuscation process is described in Section II-C. Next, Bob delivers the synthesized DNA to Alice with a sequencing report showing that no errors were found in the synthesized DNA. For simplicity, assume that Bob delivers plasmids, although the methods discussed are applicable to any DNA packaging.

Alice may send the DNA received from Bob to a third party for additional sequencing. If Alice performs sequencing on her own premises, in an air-gapped network disconnected from the Internet, on the computers used to order the genes, she will mitigate the RDI threat. However, it is difficult to believe that a DIY biologist, or even a small company, would perform additional sequencing after receiving the sequencing report of a trusted gene synthesis company, i.e., Bob. In a reasonable scenario, Alice would inspect the sequencing results using computer/s subverted by Eve (Figure 1.7-9). As a result, the order's sequencing report would reflect the original DNA sequence that Alice intended to order.

Alice then inserts the plasmid containing an obfuscated select agent into Cas9 expressing cells. If the plasmid is introduced into a non-Cas9 expressing cell, Alice might have a chance to detect the attack before any damage is done.[4] The malicious DNA sequence produced by Eve contains all of the components required to decode the select agent using Cas9 protein available within the cell. As the result, the select agent is expressed within the cells.

### D. Below the DNA screening radar

Next, we discuss a generic DNA obfuscation method for circumventing *Best Match*-based screening protocols. All DNA obfuscation methods discussed in this paper are facilitated by the maturity and prevalence of CRISPR-based DNA editing methods. For instance, a gene encoding the Cas9 protein, a key component of the CRISPR system, can easily be obtained from many sources in various packaging options. The general idea of SoC obfuscation revolves around interleaving SoC fragments with legitimate genes, such that for every 200bp fragment the best match is always a legitimate sequence. The size of the SoC fragments that pass below the radar of current screening methods is determined by the 200bp constraint and the similarity of the fragment to legitimate sequences. We elaborate on this next when discussing the SoCO1 and SoCO2 methods.

[4]See Section VI for a discussion of proposed defense measures.

*1) Level 1 SoC Obfuscation (SoCO1):* As an implementation of the HHS guidelines, GenoTHREAT [26] splits the query sequence into consecutive 200bp fragments. This is much more efficient than using a sliding window of 200bp, in terms of the number of `BLAST` queries. Yet, such a method is very sensitive to short DNA inserts. For the sake of clarity, we first discuss the simplest DNA obfuscation method which is specifically targeted at circumventing GenoTHREAT. We will refer to this method as Level 1 Sequence of Concern Obfuscation (SoCO1). Figure 1.10 depicts a plasmid with obfuscated malicious DNA and its reconstruction process. The pseudocode of SoCO1 is presented in Algorithm 1.

In order to hide the malicious DNA $P$ (payload) from GenoTHREAT, we split it into small fragments ($P = p_1 + \ldots + p_n$) of 154bp each, padded with NSoC fillers. Plus $+$ denotes string concatenation. This may look like a trivial obfuscation against pattern-based detectors that became obsolete in cyberwarfare long ago, but its biological implementation poses several challenges, as discussed below.

Similar to the old, well-known encrypted or oligomorphic malware [38], building the decoder is the most challenging part in designing obfuscated malicious DNA. Here the decoder should operate within living cells rather than in cyberspace. In general, the decoder needs to perform two tasks when reconstructing the malicious DNA: (1) it should cut out the NSoC fillers between the SoC fragments, and (2) it should stitch the SoC fragments forming operational DNA together.

In order to cut out the NSoC fillers during reconstruction, we specify the cut points before and after each SoC fragment using a unique 23bp sequence $f$, the details of which we omit here [39]. The following example of such a sequence is cut at the nucleotide marked with ✂.

$$f = \underbrace{\begin{matrix}\texttt{CCTTCC}\\ \texttt{GGAAGG}\\ \texttt{✂}\end{matrix}}_{\text{PAM}} \quad \underbrace{\begin{matrix}\texttt{ACAAGCTCGCCGAGGTG}\\ \texttt{TGCTCGAGCGGCTCCAC}\end{matrix}}_{\text{gRNA target sequence}} \tag{2}$$

It is important to note that CRIPSR may bind to either side of the double-strand DNA molecule to perform the cut. Therefore, a reverse complement of $f$ (denoted $f_{RC}$) is cut on the opposite strand, as shown in Equation 3:

$$f_{RC} = \overbrace{\begin{matrix}\texttt{CACCTCGGCGAGCTCGT}\\ \texttt{GTGGAGCCGCTCGAGCA}\end{matrix}}^{\text{gRNA target sequence}} \quad \overbrace{\begin{matrix}\texttt{✂}\\ \texttt{GGAAGG}\\ \texttt{CCTTCC}\end{matrix}}^{\text{PAM}} \tag{3}$$

We use $f + f_{RC}$ as the NSoC fillers between SoC fragments ($p_i$), such that the length of $f_{RC} + p_i + f$ is exactly 200bp. The DNA code block $p_1 + f + f_{RC} + p_2 + \ldots + f_{RC} + p_n$ is not detected by GenoTHREAT as an SoC even if $P$ is found within the SoC database, because there are no best matches with a query coverage of 100%. We show how to mitigate this problem in Section V-A. We assume that Cas9 protein is available within the cells. The decoder block of the malicious DNA should contain the gRNA scaffold targeting $f$ in order to form a CRISPR that will cut out $f + f_{RC}$ between consecutive SoC fragments ($p_i$✂$f + f_{RC}$✂$p_{i+1}$).

Next, there is a need to repair the DNA cut made by CRISPR when removing the residue PAM nucleotides[5] which

[5]For example, the leftmost `GG` on the lower strand in Equation 2.

**Algorithm 1:** SoC Obfuscation 1 (SoCO1)

**Input:** $soc$ – a Sequence of Concern
**Output:** $osoc$ – an obfuscated Sequence of Concern
/* @post-condition: Expressing $O$ in Cas9 containing environment results in assembly of $P$ */
Partition $soc$ into 154bp fragments $soc = soc_1 + soc_2 + \ldots + soc_n$
Let $r$ be a promoter and a ribosome binding site
Let $t$ be a terminator
Let $f$ be a 23bp long efficient Cas9 cutting site [39]
$f_{RC} \leftarrow$ reverse complement of $F$
$body_1 \leftarrow r + soc_1 + f$
$\forall_{i=2}^{n-1} body_i \leftarrow f_{RC} + soc_i + f$
$body_n \leftarrow f_{RC} + soc_n + t$
$Body \leftarrow r+soc_1+f+\left(\sum_{i=2}^{n-1} f_{RC} + soc_i + f\right)+f_{RC}+soc_n+t$
Let $grs$ be a gRNA scaffold targeting $f$
// $grs$ also targets $f_{RC}$ on the opposite strand.
**for** *each* $soc_i, soc_{i+1}$ **do**
  $s_i^{30} \leftarrow$ 30 bp suffix of $soc_i$
  $p_{i+1}^{30} \leftarrow$ 30 bp prefix of of $soc_{i+1}$
  $hdr_i \leftarrow s_i^{30} + p_{i+1}^{30}$
Let $nonce$ be a 40bp long DNA sequence, which does not contain $f$ or $f_{RC}$
$Decoder \leftarrow nonce+grs+hdr_1+nonce+\ldots+hdr_{n-1}$
// Assemble and return the obfuscated sequence
**return** $osoc = Body + Decoder$

**Algorithm 2:** SoC Obfuscation 2 (SoCO2)

**Input:**
$soc$ – a Sequence of Concern
SCREENER() – a black-box screening algorithm
**Output:** $osoc$ obfuscated Sequence of Concern
/* @post-condition: Expressing $osoc$ in Cas9 containing environment results in an assembly of $soc$ */
Partition $soc$ into 60bp fragments $soc = soc_1 + soc_2 + \ldots + soc_n$
Build $r$,$t$,$grs$, and $hdr_i$ as in SoCO1
Optimize $f$ to minimize detection rate
$Body =$CAMOUFLAGE$(R + p_1 + f)$
$Body \mathrel{+}= \sum_{i=2}^{n-1}$CAMOUFLAGE$(f_{RC} + p_i + f)$
$Body \mathrel{+}=$CAMOUFLAGE$(f_{RC} + p_n + T)$
$Decoder = grs + \sum_{i=1}^{n-1}$CAMOUFLAGE$(hdr_i)$
// Assemble and return the obfuscated sequence
**return** $osoc = Body + Decoder$
**Function** CAMOUFLAGE$(x)$**:**
  /* Find a gene $c$ with a highly scored gapless alignment ($\alpha$) to $x$. */
  Find best scored $(c, \alpha) \in$ BLAST($x$,*no gaps*) such that $\alpha(x)$ is at least 200pb from $c$'s ends
  $cx \leftarrow$ replace $\alpha(x)$ in $c$ with $x$
  **if** SCREENER$(cx)$ =*hit* **then**
    **error** obfuscation failed
  **else**
    **return** $cx$

are not a part of $P$. Such a repair can be performed using the HDR process which requires a 60bp long template ($HDR_i$) – a DNA sequence covering both the last 30 nucleotides of $p_i$ and the first 30 nucleotides of $p_{i+1}$. A 60bp long HDR template is the shortest HDR template shown to perform well in practice [22]. The longer the HDR templates are and the more replicas of these templates are found within the cell, the higher is the HDR efficiency. Since $\{HDR_i\}$ are also fragments of the SoC sequence, it is important to add NSoC fillers between them (although without $f$ or $f_{RC}$) in order to remain below the screening radar. In further discussions we will refer to both $p_i$ and $HDR_i$ as SoC fragments.

Overall, the malicious DNA sequence injected into an online synthetic DNA order, should contain the split SoC, the gRNA scaffold, and a set of HDR templates, as shown in Algorithm 1. We refer to such sequences as SoCO1 sequences. Generating SoCO1 sequences requires minimal computing time, but they are detected easily as shown in Section V. In particular, SoCO1 sequences can be detected by relaxing GenoTHREAT 's 100% query coverage constraint and returning a hit if a sequence with the highest score is a, SoC.

*2) Level 2 SoC Obfuscation (SoCO2):* Next, we discuss a more sophisticated obfuscation method that allows an SoC to remain below the screening radar for any *Best Match*-based screening implemented according to the HHS guidelines. Specifically, we exploit the screening protocol specification stating that if for every 200bp fragment of a query sequence, the best match is NSoC, the query is not a hit.

The main idea here is to find NSoC genes which are most similar to the SoC fragments ($p_i$ and $HDR_i$) and embed the fragments within NSoC genes, such that every 200bp window would better align with the NSoC gene than with the SoC. We will refer to such NSoC genes as *camouflage genes*.

In addition, reducing the size of the SoC fragments increases their likelihood of blending within the camouflage genes. However, reducing the size of $p_i$ increases the number of fragments, which leads to a larger number of cuts and repairs, consequently reducing the effectiveness of the decoder. Since 64pb is the currently known lower bound on the size of HDR templates [22], there is no point in using shorter SoC fragments. Therefore, in the following discussions, we assume that the length of SoC fragments and HDR templates is $64bp = |p_i| = |HDR_i|$.

Algorithm 2 presents the pseudocode of SoCO2. The general obfuscation process is depicted in Figure 2a-f. The SoC is split into overlapping fragments $p_i$ and $HDR_i$. Each fragment is embedded in the most similar NSoC sequence found in some public gene database. CRISPR cut points are set in between consecutive $p_i$ fragments in order to remove the fillers once the DNA is inserted into a cell. The decoder sequence includes a gRNA scaffold to perform the cuts and the camouflaged HDR templates to stitch the $p_i$ fragments.

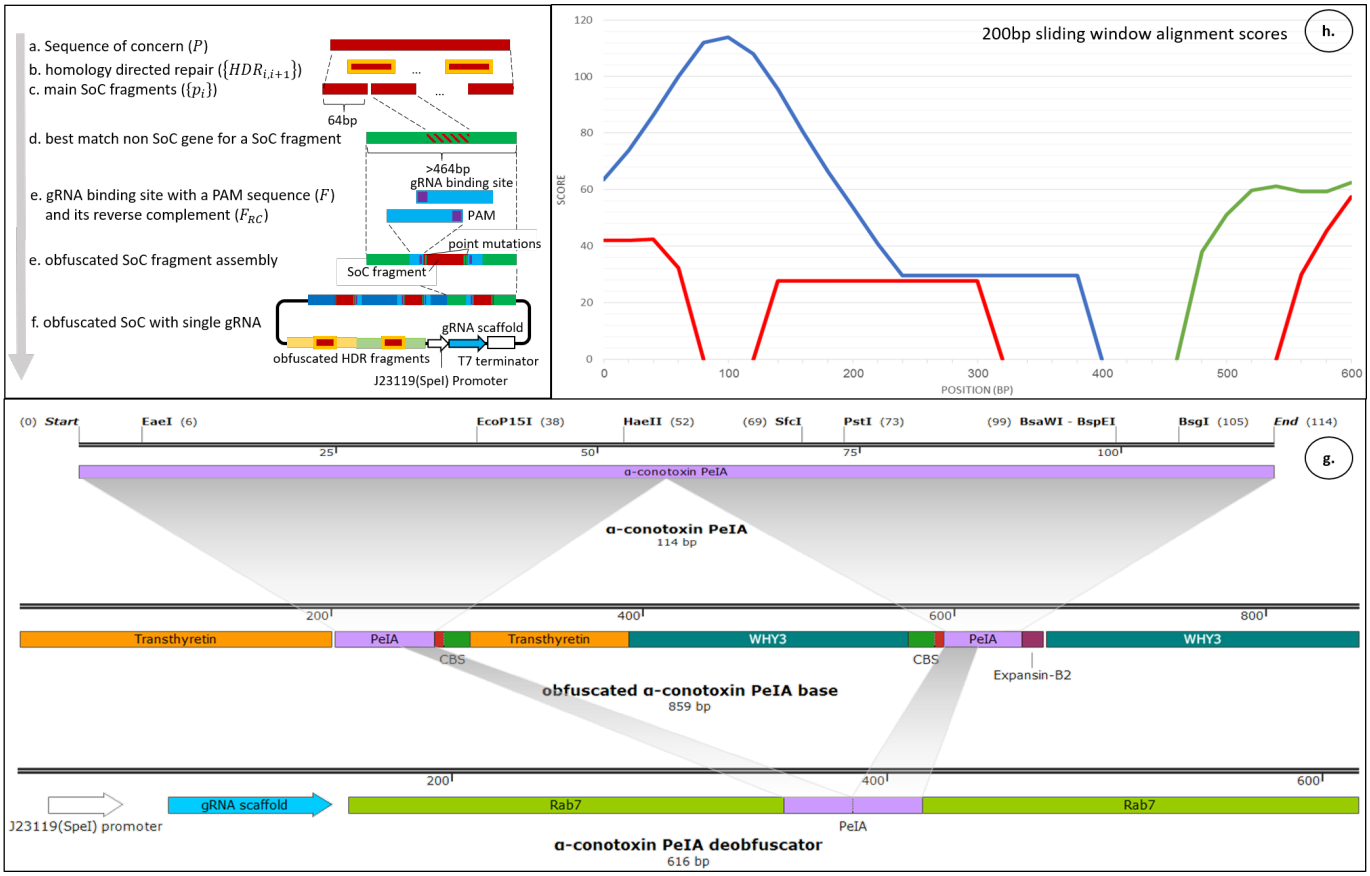


Fig. 2. Level 2 SoC obfuscation. Schematics of the SoC obfuscation process (a-f); a DNA containing obfuscated $\alpha$-conotoxin PeAI and a decoder sequence that facilitates its reconstruction (g); `blast` scores when scanning an obfuscated sequence with a 200bp sliding window (h). Colors correspond to the schematic on the left: green and blue lines are alignment scores with camouflage genes, and the red line is the alignment score with the SoC subject.

There could be many variants of this general obfuscation scheme, including variations of the gRNA binding sites. On the one hand, these sites may be designed individually for each cut point in order to improve the stealthiness of the obfuscation. But in this case they will require multiple gRNA scaffolds, each targeting a different cut point, potentially reducing the decoding's effectiveness. On the other hand, repeated gRNA binding sites may increase the likelihood of obfuscation failure. Our preliminary experiments showed that optimizing the gRNA binding sites is important for staying below the radar of a well implemented screening algorithm. We avoid disclosing the gRNA optimization details publicly to reduce exploitation risk. There are also various considerations when choosing the genes to use as a camouflage. An adversary may decide not to choose the most similar genes, but rather to choose NSoC housekeeping genes shared between regulated and unregulated organisms in order to mislead a human operator during follow-up screening.

## IV. Gene Edit Distance

In order to harden synthetic DNA order screening, reduce the non-regulated distribution of select agents and toxins, and prevent attacks like the one described in Section III, we propose a new DNA screening algorithm termed Gene Edit Distance (GED). The algorithm is designed to assess the difficulty of SoC assembly from a DNA sequence. In order to do so, GED screens the query sequence to find all substrings which are similar to fragments of an SoC. Then, GED quantifies the effort of assembling an SoC from these fragments. Although, designed with a focus on SoC, GED can quantify the effort required to assemble any target sequence $t$ from a query sequence $q$ using a standard CRISPR system. More specifically, we count the number of cuts and repairs required for constructing the target sequence from the query sequence.

In a standard biological sequence alignment, a typical objective is to identify genes conserved in different genomes. To achieve this objective, the match reward ($rm$), mismatch penalty ($pmm$), and gap penalties ($gpo$, $gpx$) must be well balanced when computing the alignment score. GED's objective is more complex. On the one hand, we need to find short conserved regions with minimal gaps within the query sequence. On the other hand, we want to concatenate the short conserved regions regardless of the gap length between them.

*Example 2:* A default configuration when aligning two sequences with `BLAST` [40] is $rm = 2$, $pmm = 3$, $pgo = 5$, and $pgx = 2$. Figure 3 presents the results of aligning the obfuscated $\alpha$-conotoxin PeIA[6] from Figure 2 to the PeIA sequence using default configuration. `BLAST` returns three

[6]A short toxic peptide.

ranges (alignments). As long as $pgx > 0$, these ranges will not be merged by BLAST due to the length of the gap that would be opened in the target sequence. However, when removing a sequence between two consecutive SoC fragments using the CRISPR system and HDR templates, the distance between them does not play a critical role.

*1) Gap removal penalty and adjusted alignment score:* In order to achieve the objective of GED, we introduce a new gap removal penalty ($prm$) that may substitute some gap opening and extension penalties within the target sequence. [7] Let $g = [a, b]$ be a gap in the target sequence ($\forall_{i \in g}, \alpha(i) = \perp$). Removal of $[a, b]$ from the query sequence requires two cuts, at $a$ and at $b$ respectively, followed by an HDR (e.g., Figures 1.10 and 5). These operations may fail. Let $\gamma$ be the probability that a gap is successfully removed.

To define GED we neglect some biological constraints of the CRISPR system, such as specific PAM sites targeted by Cas9, assuming that cuts can occur at any point in a DNA sequence. This is a worst-case assumption overestimating the potential risk. Future versions of GED may take biological constraints into account to reduce false positives. Currently, in the absence of advanced adversarial techniques, GED detects obfuscated SoC with 100% accuracy, as shown in Section V.

If the gap $g = [a, b]$ is removed, the score of the alignment will increase by $pgo + pgx \cdot (b - a + 1)$ and reduce by $prm$. Let $\mathcal{G}_q(\alpha)$ and $\mathcal{G}_t(\alpha)$ be sets of all gaps in the query and target sequences respectively according to the alignment $\alpha$. Let $\mathcal{G}_t^k \subseteq \mathcal{G}_t$ be a subset of the $k$ longest gaps in the target sequence to be removed. The probability that all gaps in $\mathcal{G}_t^k$ will be successfully removed is $\gamma^k$. Next, we define the alignment score adjustment as a result of designating $\mathcal{G}_t^k$ for removal.

*Definition 1 (Adjusted alignment score):* Given an alignment $\alpha$; a subset of gaps $\mathcal{G}_t^k$ in the target sequence of $\alpha$; gap opening and gap extension penalties $pgo$ and $pgx$ respectively; gap removal penalty $prm$; and gap removal probability $\gamma$; we define the change in the alignment score as the result of the removal of the $k$ longest gaps in the target sequence $t$ as:

$$Score^k(\alpha) = Score + \gamma^{|\mathcal{G}_t^k|} \cdot \sum_{g \in \mathcal{G}_t^k} (pgo + pgx \cdot |g| - prm).$$

Next, we examine the choice of gaps to be removed ($\mathcal{G}_{trm}$) and the parametrization of $Score^k$. We assume that the match reward ($rm$), as well as the mismatch, gap opening, and gap extension penalties ($pmm, pgo, pgx$) are set at their default values or according to the biological considerations (which are beyond the scope or this article).

$Score^k$ approaches $Score$ when the number of gaps to be removed $k$ increases and $\gamma < 1$. The choice of $\gamma$ depends on the assumptions made regarding the expected biological effectiveness of the attacks given typical bioengineering tools used by potential victims today. $\gamma = 0$ signifies that no attacker could ever rely on residue Cas9 protein in the cells and the SoC obfuscation attack described in Section III-D is impossible. $\gamma = 1$ signifies 100% success of gene editing, which leads to the successful decoding of SoCO2 sequences regardless of the number of SoC fragments.

[7] Note that variation of gap penalties for query and target sequences has successfully been used in the past for other use cases [41].

The gap removal penalty $prm$ should be set to a value that would justify gap removal using bioengineering tools. For example, if only the removal of gaps larger than $x$bp is justified, then the gap removal penalty should be set to:

$$prm = pgo + pgx \cdot x.$$

Obviously, the longer a gap is, the more worthwhile its removal is. Thus, for the sake of computing $Score^k$, only the longest gaps are selected. The number of gaps $k$ that should be included in $\mathcal{G}_t^k$ for the maximal $Score^k$ depends on the parameters and can be selected using a grid search or simple hill climbing algorithm.

*Example 3:* Assume, for example, that we examine a 10Kpb long query sequence, comparing it to a 2Kpb long SoC. Also assume a negatively scored optimal alignment between the two sequences which contains 2K matching base pairs, a few mismatches, and 50 gaps whose length is exponentially distributed, as depicted in Figure 4a. Although, such alignment exists, it will never be returned by the BLAST search engine, because its score is worse than the alignment with random sequences in the database. Assuming default BLAST configuration, the specific alignment score could be lower than $-4,500$. Assume a gap removal penalty of $prm = 20$. Replacing gap opening and gap extension penalties with gap removal penalties starting from the longest gaps would increase the score, as shown in Figure 4b. With $\gamma = 0.98$, setting $k = 10$ results in the highest $Score^k$ value. With $\gamma = 0.99$, the adjusted alignment score can reach $2,000$ when $k = 13$ in this example.

*Example 4:* Further following Example 2, consider the alignment of obfuscated $\alpha$-conotoxin PeIA with the unobfuscated peptide, as shown in Figure 3. The blue alignment in Figure 3c, was not identified by BLAST, because it contains very long gaps in the target sequence. Nevertheless, an adjusted score with the gap removal penalty of $prm = 20$ and $\gamma = 0.99$, according to definition 1, would result in a total score of 193 which is very close to a best match.

*2) Computing GED:* Next, we define the gene edit distance between the two aligned sequences as the number of gaps to remove (using one of the existing gene editing tools) that maximizes the adjusted alignment score of an optimal alignment.

*Definition 2 (Unidirectional gene edit distance):* The gene edit distance (GED) from $q$ to $t$ is:

$$GED(q, t) = \texttt{ARGMAX}_k \texttt{MAX}_\alpha Score^k(\alpha).$$

Although, we call GED a distance, it is not a valid mathematical distance metric, first and foremost, because it is asymmetric. GED quantifies the effort required to transform $q$ into $t$ but not vice versa. As an academic exercise, one can define a true gene edit metric, but such definition is out of the scope of this security article.

*3) Computing GED:* Current BLAST implementations will not return suitable alignments which minimize the small gaps but neglect the long ones. Thus, in order to reduce time-to-market and maintain backward compatibility with existing

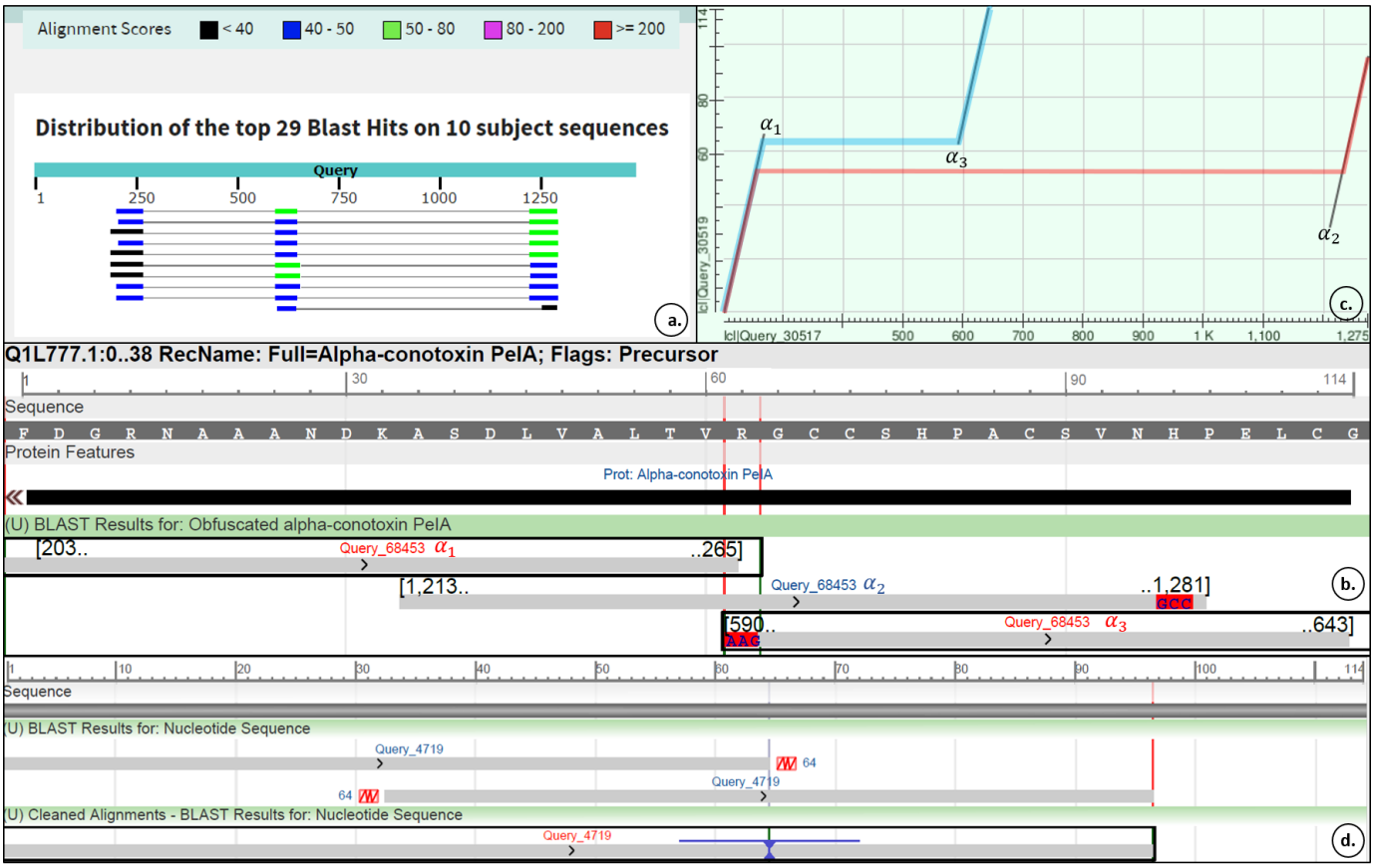


Fig. 3. Excerpts from alignment reports generated using BLASTX [40] for the obfuscated $\alpha$-conotoxin PeIA. (a) The alignment domains for 29 alignments to 10 target sequences specific to the conus family, a species that produces $\alpha$-conotoxin PeIA. (b) The images (as defined in Section II-B) of three alignments to the $\alpha$-conotoxin PeIA peptide. The alignments are arranged according to the image start positions, and the respective domain ranges appear in braces [..] at the alignments' ends. The alignment markers $\alpha_1$,$\alpha_2$,$\alpha_3$ and the domain ranges were added to the NCBI report. The scores of the three alignments range from 47.3 to 54.6. (c) A 2D plot presenting the alignments between obfuscated and non-obfuscated $\alpha$-conotoxin PeIA, found by Blast 2 sequences [42]. The x-axis represents the obfuscated DNA, and the y-axis represents the DNA that encodes $\alpha$-conotoxin PeIA. The red and blue lines represent extended alignments, with zero penalties for long gaps in the target sequence (deletion). (d) Alignments of the unified domains of $\alpha_1$ and $\alpha_3$ with their unified images. Two local alignments found by Blast 2 sequences are successfully merged into a cleaned alignment with 100 Percent Identity.

BLAST engines, we implement a GED heuristic as post-processing of standard BLAST outputs. The GED heuristic pseudocode is presented in Algorithm 3.

Standard BLAST algorithms compute a large set of small local alignments and extend these alignments when doing so increases the alignment score. We take a similar approach when computing a set of alignments using standard BLAST configuration and merging them to maximize $Score^k$. Recall that $\mathcal{A}_{q,t}$ is a set of local alignments between $q$ and $t$ returned by BLAST (Algorithm 3, line 1). Let $.start$ and $.end$ denote the first and the last position respectively, in the domain or the image of an alignment $\alpha \in \mathcal{A}_{q,t}$. Let $P = \alpha_1, \ldots, \alpha_k$ be a set alignments, whose domains are disjoint, $dom(\alpha_i).end < dom(\alpha_{i+1}).start$ (Algorithm 3, line 1). Their images may overlap, as depicted in Figure 5. A global alignment between the unified domains and the unified images of the alignments can be found using dynamic programming-based algorithms such as Needleman–Wunsch algorithm [43]. BLAST will also find a unified alignment with high probability when the query sequence fragment between the two alignment domains is removed. Such unified alignments are called cleaned alignments according BLAST.

*Example 5:* For example, consider the alignments of an obfuscated $\alpha$-conotoxin PeIA with the unobfuscated peptide in Figure 3. BLAST identifies three local alignments: $[1, 63] = \alpha_1([203, 265])$, $[33, 102] = \alpha_2([1213, 1281])$, and $[61, 114] = \alpha_3([590, 643])$. The images of $\alpha_2$ and $\alpha_3$ overlap significantly. The union of these images covers 102 nucleotides in the target sequence. As depicted in Figure 3d, BLAST successfully merges these alignments, provided the query sequence fragment between their domains is removed. We use this functionality in Algorithm 3, line 8 when merging alignments in line 3.

The output of Algorithm 3 is the number of cut and repair actions required to reconstruct the target sequence from the query sequence and the adjusted alignment score. The latter quantifies both the similarity of the body of an obfuscated SoC and the effort required to decode it within the cells. Unlike in the case of encrypted or oligomorphic malware where decoders are the easiest to detect, here we concentrate on detecting the main body of the obfuscated SoC, because the gRNA scaffolds and HDR templates comprising the decoder may be distributed among different plasmids or even different orders.

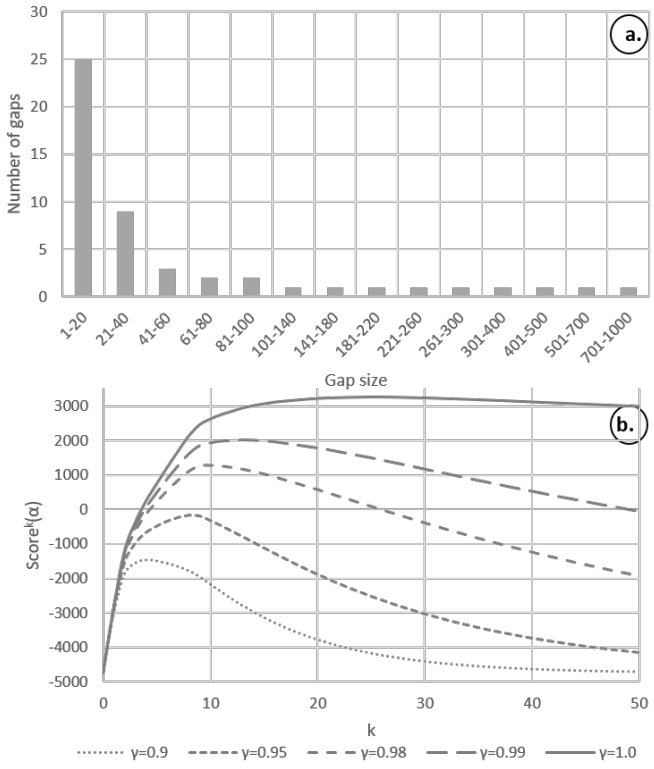


Fig. 4. Example of an alignment score correction using the GED $\Delta Score$. (a) The gap length distribution of an example alignment. (b) The corrected alignment score ($Score + \Delta Score^k$) for various choices of $k$.

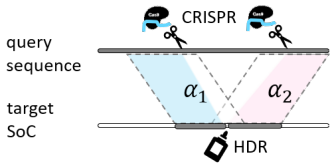


Fig. 5. Two local alignments with disjoint domains and overlapping images.

In order to return a final assessment of the risk a query sequence holds, the GED of the query sequence is evaluated against all SoC sequences in the database. The final judgement is made based on the maximal adjusted score of the query sequence for any of the SoC sequences. This allows the identificaton of seemingly benign sequences that can easily be transformed into malicious sequences that produce dangerous products. In addition, such an approach is more resilient to an attack in which a public gene database is poisoned with legitimate sequences, although poisoning with legitimate sequences maliciously marked as an SoC may result in false hits and require human attention.

## V. Evaluation

### A. GenoTHREAT+

As shown in Section III-D1, GenoTHREAT can be easily circumvented, because it scans non-overlapping 200bp fragments of the query sequence and employs very strict match constraints. In order to objectively assess the threat of DNA obfuscation, we implement a screening method (referred to as GenoTHREAT+) that does not rely on any assumptions beyond the 200bp constraint and the *Best Match* principle. While we don't elaborate on the algorithm here due to space constraints, we provide some highlights below.

GenoTHREAT+ scans the query sequence with a sliding window of length 200bp and step size of 1bp. Every such 200bp nucleotide sequence is searched in the NR and NT NCBI databases for nucleotide and protein matches respectively[8]. We use a set of keywords and anti-keywords to identify dangerous substances in the returned results. GenoTHREAT+ records the score of the best matching SoC as

[8]https://blast.ncbi.nlm.nih.gov/Blast.cgi

**Algorithm 3:** Gene Edit Distance

**Input:**
q – a query sequence
t – a target sequence
**Output:**
$k$ – the number of cut and repair operations
$Score^k$ – adjusted alignment score

```
// List local alignments A = {α} sorted by dom(α).start
A_{q,t} = BLAST(query=q,subject=t)
for each subset P ⊆ A_{q,t} of local alignments with disjoint domains do
    α_P = MERGE(P)
P* = argmax_P Score^{|P|-1}(α_P)
return k = |P*|, Score^k(α_{P*})
Function MERGE(α_1, ..., α_k):
    /* Find global alignment between unified domains and t */
    A = BLAST(query = q[⋃_i dom(α_i)], subject = t)
    α = cleaned global alignment unifying A
    /* Reintroduce gaps to form a continuous alignment */
    α∪ = {(i, ⊥) : i ∈ [dom(α_j).end, dom(α_{j+1}).start], 1 ≤ j < k}
    return α
```

TABLE I
Comparison between GenoTHREAT and GenoTHREAT+

| | Best Match | Scanning resolution | Hit | Pass |
|---|---|---|---|---|
| **Geno THREAT** | QC=100% and maximal PI | Partition to 200bp fragments with alignment extension if SoC, QC>50%, and extremity | At least one 200bp fragment, where all best matches are SoC | For all 200bp fragments, at least one best match is NSoC or no best matches |
| **Geno THREAT+** | Alignment with best score from BLAST | 200bp sliding windows | At least one 200bp window where the best match is SoC | For all 200bp windows, the best match is NSoC or no best matches |

well as the score of the best matching NSoC. Later, we derive a confidence value from the difference between them. Querying both nucleotide and protein databases for each sequence of 200bp is a highly inefficient yet very accurate approach that strictly follows the HHS guidelines. Table I highlights the differences between GenoTHREAT and GenoTHREAT+.

### B. Benchmark dataset

In order to compare GenoTHREAT, GenoTHREAT+, and Gene Edit Distance, we created a benchmark dataset. We selected 50 SoC from the UniProt database [44], all of which are marked as toxins, are manually reviewed, and contain 33 to 100 amino acids. These protein sequences were reverse translated to sequences of nucleotides using an EMBOSS online tool [45]. We ran `BLAST` queries using the nucleotide SoC to ensure that they are well detected within the GenBank database using `BLAST`. Doing so, we identified seven benign sequences with various levels of similarity to the SoC. We selected an additional 43 benign sequences from UniProt, all of which are not toxic and manually reviewed. The 50 SoC sequences were obfuscated using Algorithms 1 and 2. To create the SoCO2 sequences, we selected the camouflage genes from the 50 NSoC sequences described above. Finally, we generated 50 random nucleotide sequences (denoted as Rnd) as a reference point. There are a total of 250 nucleotide sequences in the benchmark dataset, equally split among NSoC, SoC, SoCO1, SoCO2, and Rnd.

### C. Screening Results

Benchmark sequence screening was performed using local copies of the complete GenBank NT and NR databases. A set of keywords and anti-keywords was used to identify toxic sequences based on their descriptions. While GenoTHREAT and GenoTHREAT+ require the entire database for accurate screening, GED only requires the SoC to compare with the query sequences.

We expect a good screening algorithm to report a hit when screening sequences in the SoC, SoCO1, or SoCO2. We expect a non-hit when screening sequences in the NSoC or Rnd. Similar to other studies focusing on malware detection of DNA screening, we consider the fraction of hits out of all malicious sequences as the true positive rate (TPR) and the fraction of incorrect hits out of all benign sequences as the false positive rate (FPR). Since some malicious sequences are easier to detect than others, we compute the TPR separately for the SoC, SoCO1, and SoCO2. In order to analyze the performance of the screening algorithms, we also inspect their confidence levels.

While GenoTHREAT only provides a binary decision on a query sequence, GenoTHREAT+ can provide a confidence level along with the decision. We compute the GenoTHREAT+ confidence as follows. Let $q$ be some 200bp fragment of the screened sequence. Let $MaxSoC(q)$ and $MaxNSoC(q)$ be the highest score of an alignment of $q$ with an SoC and NSoC respectively. If `BLAST` did not return

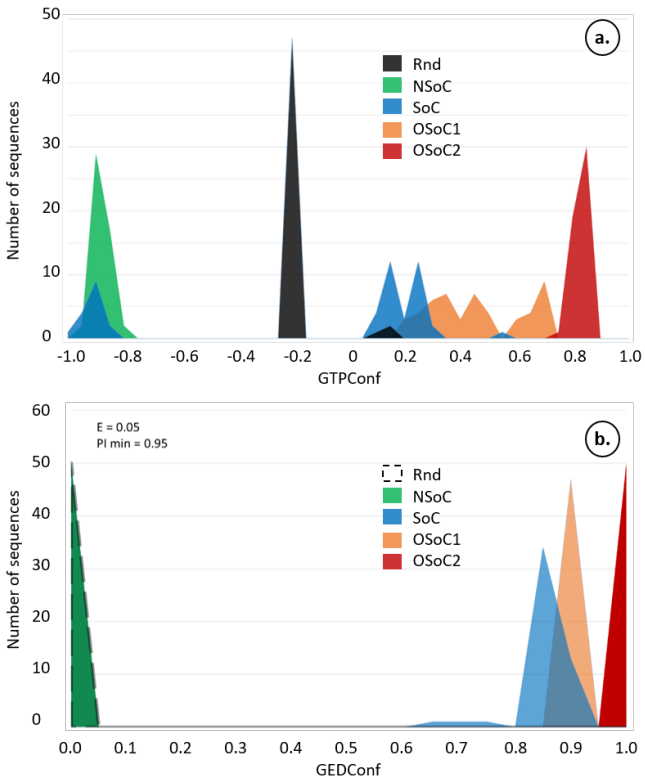


Fig. 6. GenoTHREAT+ and GED confidence levels for five types of sequences in the benchmark dataset.

TABLE II
NUMBER OF HITS PRODUCED BY DIFFERENT SCREENING APPROACHES.

| | GenoTHREAT | GenoTHREAT+ | GED |
|---|---|---|---|
| SoC hits | 50 | 50 | 50 |
| SoCO1 hits | 9 | 50 | 50 |
| SoCO2 hits | 7 | 34 | 50 |
| NSoC hits | 0 | 0 | 0 |
| Rnd hits | 0 | 3 | 0 |
| TPR | 0.44 | 0.89 | 1.0 |
| FPR | 0.0 | 0.03* | 0.0 |

* Due to hits on random sequences.

alignments with SoC or NSoC, we set the respective score to zero. We define GenoTHREAT+ confidence as

$$GTPConf = \max_{q} \left\{ \frac{MaxSoC(q) - MaxNSoC(q)}{\max\{MaxSoC(q), MaxNSoC(q)\}} \right\}$$

$GTPConf > 0$ means a hit, because there is at least one 200bp fragment that is more similar to an SoC than to an NSoC.

The confidence of GED is simply the maximal adjusted score it returns when screening $q$:

$$GEDConf = \max_{t \in SoC} GED(q, t)$$

The value of $GEDConf$ is between zero and one, where $GEDConf = 1$ means that $q$ is definitely a SoC.

Table II summarizes the performance of GenoTHREAT, GenoTHREAT+, and GED on the benchmark dataset. Confidence levels of GenoTHREAT+ and GED when screening the 250 benchmark sequences are presented in Figure 6.

As expected, all SoCs are the closest to being malicious (right part of the chart) and all NSoCs are the closest to being benign (left part of the chart). Most random sequences are classified by GenoTHREAT+ as benign (low negative confidence), because there are many more benign sequences than toxic sequences in the GenBank database. Nevertheless, some random sequences contain 200bp fragments that are more similar to an SoC than to an NSoC. One such fragment is sufficient to produce a hit and require human attention. We consider such statistical errors as an inherent deficiency of the *Best Match* approach. In contrast, according to GED, random sequences are similar to benign sequences with respect to the effort required to transform them into something dangerous.

Next, we observe a high variance of the confidence levels of obfuscated sequences in both Figure 6a and b. While SoCO1 sequences are always detected, some SoCO2 sequences remain below the radar of the *Best Match* approach, represented here by GenoTHREAT+. This is due, of course, to the camouflage genes which contribute the most to the PI value in each 200bp fragment. Nevertheless, some SoCO2 sequences are detected, because no sufficiently similar camouflage gene was found during obfuscation. An important conclusion from these results is that some sequences are easier to obfuscate than others.

GED successfully detects all obfuscated malicious sequences. Moreover the large confidence gap between the most suspicious benign sequence and the least suspicious malicious sequence ensures GED's robustness as a screening algorithm. We also note that GED is an order of magnitude faster than GenoTHREAT, because it compares the query sequence with a small database of SoC and does not need to query for every 200bp fragment.

## VI. Discussion and concluding remarks

### A. The threat

In this paper we demonstrated a potential attack where remote attacker injects a malicious DNA that produces dangerous substance into a biological pipeline of the victim. Similar to cyber physical attacks where a cyber hackeer is able to affect the physical world, here the adversary affects a biological substance without direct contact with it. The demonstrated attack makes use of multiple weaknesses at three levels of the bioenginerring pipeline: software, biosecurity screening, and biological protocols. Although, simpler attacks exist where a remote attacker may harm biological experiments we chose demonstrating the RDI threat here because it highlights the lack of cybersecurity protections along the bioengineering pipeline.

Furthermore, RDI threat highlights the opportunities for applying cyber security know how in new contexts such as biosecurity and gene coding. Current biosecurity practice suggests obscuring the details of the DNA screening algorithms in order to reduce the likelihood of circumvention. As a case in point HSP refers to the HHS guidelines but omits the protocol details. Detailed specification as well as benchmark DNA sequences are available to the IGSC members only under non-disclosure agreement. However, general cybersecurity practice advises against security by obscurity [46]. Opening security protocols to the public allows rapid identification of weaknesses eventually resulting in more robust and adversary-resilient protocols.

Opening DNA screening protocols to the review of cybersecurity professionals is timely now because it will not cause immediate significant harm. DNA screening is not required yet by international regulation. There are still multiple companies around the world providing synthetic DNA without biosecurity screening. So, if a bioterrorist would like to buy dangerous synthetic DNA he can do so today. Hopefully, in the next few years biosecurity screening will be a must in all countries. By then, we need robust adversary resilient DNA screening protocols.

### B. The mitigation

The cybernetic parts of the RDI threat can be easily mitigated using appropriate methods for maintaining electronic integrity as well as communication over multiple independent channels. First, all reports sent during and after the DNA production should be password protected following best practice of paperless communication in banking and insurance domains. Second, a hard-copy of the quality report including the DNA sequence should be delivered with the tube containing the synthetic DNA. Finally, a sticker on the tube indicating the most important information, from the security perspective, would be highly valuable. For example, are there gRNA scaffolds or Cas9 encoding genes within the tube.

In the biosecurity level, a hardened DNA screening method based on GED, that we present in this article, accurately identifies obfuscated malicious DNA that relies of CRSIPR or similar gene editing techniques. Future enhancements to DNA screening may rely on deep learning for sequence analysis and DNA function prediction. Adversarial learning techniques can be used to further increase the resilience of screening algorithms against malicious DNA sequences that are not yet on the SoC list.

In the biological level, any use of Cas9 should be considered a critical section of the biological protocol. A security aware biologist should use existing methods to inhibit the expression of Cas9 within the cells [47]. Currently such methods are only used to achieve the desire biological effects but not as safety/security measures.

### C. Ethics and responsible disclosure

As noted above, this paper comes in a time when dangerous DNA can still be purchased online without screening. Yet it is important to enlist cybersecurity specialists for reviewing and hardening biosecurity protocols. The weakness was disclosed to two companies heavily involved in biosecurity screening. One of the companies confirmed that the plasmid depicted in Figure 2.g is indeed dangerous. The details of the attack and immediate solutions are still discussed. We intend to disclose the full details to members of the IGCS and IASB consortiums and the relevant national institutes. For now we avoid disclosing the gRNA optimization details which showed to be important for circumventing *Best Match* based screening.

## REFERENCES


[1] Sven Panke Matthias Heinemann. Synthetic biology—putting engineering into biology. *Bioinformatics*, 22(22):2790—-2799, 2006.

[2] Florian Lienert, Jason J Lohmueller, Abhishek Garg, and Pamela A Silver. Synthetic biology in mammalian cells: next generation research tools and therapeutics. *Nature reviews Molecular cell biology*, 15(2):95, 2014.

[3] Ewen Callaway. Minimal'cell raises stakes in race to harness synthetic life. *Nature*, 531:557–558, 2016.

[4] Joanne Kamens. The addgene repository: an international nonprofit plasmid and data resource. *Nucleic Acids Research*, 43:1152–1157, 2015.

[5] Ken-ichi Yamazaki, Kim de Mora, and Kensuke Saitoh. Biobrick-based 'quick gene assembly'in vitro. *Synthetic Biology*, 2(1):ysx003, 2017.

[6] Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. Genbank. *Nucleic acids research*, 41(D1):D36–D42, 2012.

[7] Michael J Czar, Yizhi Cai, and Jean Peccoud. Writing dna with genocad$^{TM}$. *Nucleic acids research*, 37(suppl_2):W40–W47, 2009.

[8] Wlodek Mandecki, Mark A Hayden, Mary Ann Shallcross, and Elizabeth Stotland. A totally synthetic plasmid for general cloning, gene expression and mutagenesis in escherichia coli. *Gene*, 94(1):103–107, 1990.

[9] Jeronimo Cello, Aniko V Paul, and Eckard Wimmer. Chemical synthesis of poliovirus cdna: generation of infectious virus in the absence of natural template. *science*, 297(5583):1016–1018, 2002.

[10] Gregory D Koblentz. The de novo synthesis of horsepox virus: implications for biosecurity and recommendations for preventing the reemergence of smallpox. *Health security*, 15(6):620–628, 2017.

[11] Tran Thi Nhu Thao, Fabien Labroussaa, Nadine Ebert, Philip V'kovski, Hanspeter Stalder, Jamine Portmann, Jenna Kelly, Silvio Steiner, Melle Holwerda, Annika Kratzel, et al. Rapid reconstruction of sars-cov-2 using a synthetic genomics platform. *bioRxiv*, 2020.

[12] Engineering National Academies of Sciences, Medicine, et al. *Biodefense in the age of synthetic biology*. National Academies Press, 2018.

[13] Gigi Kwik Gronvall. Synthetic biology: Biosecurity and biosafety implications. In *Defense Against Biological Attacks*, pages 225–232. Springer, 2019.

[14] Rachel M West and Gigi Kwik Gronvall. Crispr cautions: Biosecurity implications of gene editing. *Perspectives in Biology and Medicine*, 63(1):73–92, 2020.

[15] Department of Health and Human Services. Screening framework guidance for providers of synthetic double-stranded dna. https://www.phe.gov/Preparedness/legal/guidance/syndna/Documents/syndna-guidance.pdf, 2010.

[16] Michael Montague Tom Inglesby Gigi Kwik Gronvall Amanda Kobokovich, Rachel West. Strengthening security for gene synthesis:recommendations for governance. *Health Security*, 17(6), 2019.

[17] Zhicong Huang, Erman Ayday, Jacques Fellay, Jean-Pierre Hubaux, and Ari Juels. Genoguard: Protecting genomic data against brute-force attacks. In *2015 IEEE Symposium on Security and Privacy*, pages 447–462. IEEE, 2015.

[18] Jacob Caswell, Jason D Gans, Nicholas Generous, Corey M Hudson, Eric Merkley, Curtis Johnson, Christopher Oehmen, Kristin Omberg, Emilie Purvine, Karen Taylor, et al. Defending our public biological databases as a global critical infrastructure. *Frontiers in bioengineering and biotechnology*, 7, 2019.

[19] Iliya Fayans, Yair Motro, Lior Rokach, Yossi Oren, and Jacob Moran-Gilad. Cyber security threats in the microbial genomics era: implications for public health. *Eurosurveillance*, 25(6):1900574, 2020.

[20] James Diggans and Emily Leproust. Next steps for access to safe, secure dna synthesis. *Frontiers in bioengineering and biotechnology*, 7:86, 2019.

[21] Liren Wang, Yanjiao Shao, Yuting Guan, Liang Li, Lijuan Wu, Fangrui Chen, Meizhen Liu, Huaqing Chen, Yanlin Ma, Xueyun Ma, et al. Large genomic fragment deletion and functional gene cassette knock-in via cas9 protein mediated genome editing in one-cell rodent embryos. *Scientific reports*, 5:17517, 2015.

[22] C Pohl, JAKW Kiel, AJM Driessen, RAL Bovenberg, and Y Nygard. Crispr/cas9 based genome editing of penicillium chrysogenum. *ACS synthetic biology*, 5(7):754–764, 2016.

[23] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[24] International Gene Synthesis Consortium. Harmonized screening protocol v2.0. https://genesynthesisconsortium.org/wp-content/uploads/IGSCHarmonizedProtocol11-21-17.pdf, 2017.

[25] International Association Synthetic Biology (IASB). The iasb code of conduct for best practices in gene synthesis, 2009.

[26] Laura Adam, Michael Kozar, Gaelle Letort, Olivier Mirat, Arunima Srivastava, Tyler Stewart, Mandy L Wilson, and Jean Peccoud. Strengths and limitations of the federal guidance on synthetic dna. *Nature biotechnology*, 29(3):208, 2011.

[27] Diane DiEuliis, Sarah R Carter, and Gigi Kwik Gronvall. Options for synthetic dna order screening, revisited. *mSphere*, 2(4):e00319–17, 2017.

[28] Daniel G Gibson, Lei Young, Ray-Yuan Chuang, J Craig Venter, Clyde A Hutchison III, and Hamilton O Smith. Enzymatic assembly of dna molecules up to several hundred kilobases. *Nature methods*, 6(5):343, 2009.

[29] Hubert Bernauer, Jason Christopher, Werner Deininger, Markus Fischer, Philip Habermeier, Klaus Heumann, Stephen Maurer, Heinz Schwer, Peer Stähler, and Tobias Wagner. Technical solutions for biosecurity in synthetic biology. *Industry Association Synthetic Biology, Munich, Germany*, 2008.

[30] Sarah R Carter and Robert M Friedman. Dna synthesis and biosecurity: lessons learned and options for the future. *J Craig Venter Institute, La Jolla, CA*, 2015.

[31] Dreycey Albin, Dan Nasko, RA Leo Elworth, Jacob Lu, Advait Balaji, Christian Diaz, Nidhi Shah, Jeremy Selengut, Chris Hulme-Lowe, Pravin Muthu, et al. Seqscreen: a biocuration platform for robust taxonomic and biological process characterization of nucleic acid sequences of interest. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1729–1736. IEEE, 2019.

[32] Jonathan B Tucker. Double-edged dna: preventing the misuse of gene synthesis. *Issues in Science and Technology*, 26(3):23–32, 2010.

[33] Hans Bügl, John P Danner, Robert J Molinari, John T Mulligan, Han-Oh Park, Bas Reichert, David A Roth, Ralf Wagner, Bruce Budowle, Robert M Scripp, et al. Dna synthesis and biological security. *Nature biotechnology*, 25(6):627, 2007.

[34] Bret Jordan Rich Piazza, John Wunder. *STIX$^{TM}$ Version 2.0. Part 1: STIX Core Concepts*. OASIS Committee Specification 01, 2017. http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part1-stix-core.html.

[35] Justin Warner ICEBRG. Man in the browser. In *MITRE ATT&CK*, number T1185. The MITRE Corporation.

[36] T. Casey. Understanding cyberthreat motivations to improve defense. https://communities.intel.com/servlet/JiveServlet/previewBody/23856-102-1-28290/understanding-cyberthreat-motivations-to-improve-defense-paper-1.pdf, February 2015. Intel.

[37] Robert Carlson. The changing economics of dna synthesis. *Nature biotechnology*, 27(12):1091, 2009.

[38] Babak Bashari Rad, Maslin Masrom, and Suhaimi Ibrahim. Camouflage in malware: from encryption to metamorphism. *International Journal of Computer Science and Network Security*, 12(8):74–83, 2012.

[39] Yingbo Cui, Jiaming Xu, Minxia Cheng, Xiangke Liao, and Shaoliang Peng. Review of crispr/cas9 sgrna design tools. *Interdisciplinary Sciences: Computational Life Sciences*, 10(2):455–465, 2018.

[40] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[41] Julie D Thompson. Introducing variable gap penalties to sequence alignment in linear space. *Bioinformatics*, 11(2):181–186, 1995.

[42] Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb Miller. A greedy algorithm for aligning dna sequences. *Journal of Computational biology*, 7(1-2):203–214, 2000.

[43] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[44] UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.

[45] Fábio Madeira, Young Mi Park, Joon Lee, Nicola Buso, Tamer Gur, Nandana Madhusoodanan, Prasad Basutkar, Adrian RN Tivey, Simon C Potter, Robert D Finn, et al. The embl-ebi search and sequence analysis tools apis in 2019. *Nucleic acids research*, 47(W1):W636–W641, 2019.

[46] Rebecca T Mercuri and Peter G Neumann. Security by obscurity. *Communications of the ACM*, 46(11):160, 2003.

[47] Felix Bubeck, Mareike D Hoffmann, Zander Harteveld, Sabine Aschenbrenner, Andreas Bietz, Max C Waldhauer, Kathleen Börner, Julia Fakhiri, Carolin Schmelas, Laura Dietz, et al. Engineered anti-crispr pro-

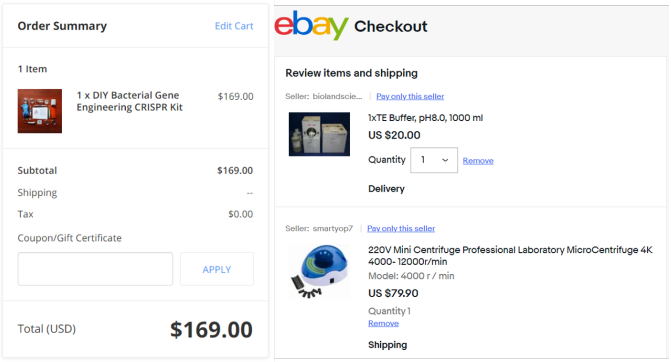


Fig. 7. DIY biology instrumentation, including a CRISPR kit from the-odin.com (left), and a microcentrifuge and TE buffer from eBay (right).

teins for optogenetic control of crispr–cas9. *Nature methods*, 15(11):924, 2018.

## APPENDIX

### A. DIY biology instrumentation

To conclude our discussion on the RDI threat, we note the wide applicability of the RDI threat in terms of biological instrumentation. Everything that is required for Alice to develop her own genes and apply them to cells is available for purchase by any DIY biology enthusiast. Moreover, no special expertise or sophisticated equipment is required. For instance, the DIY Bacterial Gene Engineering CRISPR Kit can be purchased from the-odin.com (Figure 7 left). The kit contains E. coli cells (that were stripped of any pathogens and are considered safe), Cas9 coding plasmids, growth media, petri plates, and other instrumentation, including detailed instructions on the CRISPR protocol. When a plasmid (such as in Figure 2.g) is ordered online, it comes in a dry form and needs to be resuspended in a TE buffer, which is a commonly used buffer solution. Most resuspension protocols suggest centrifuging the synthetic DNA upon receipt, and the required microcentrifuge and TE buffer are available for purchase on eBay (Figure 7 right).

Innocent Alice experimenting with bioengineering at home may become the victim of a cyber attack used to produce life threatening toxins or viruses.

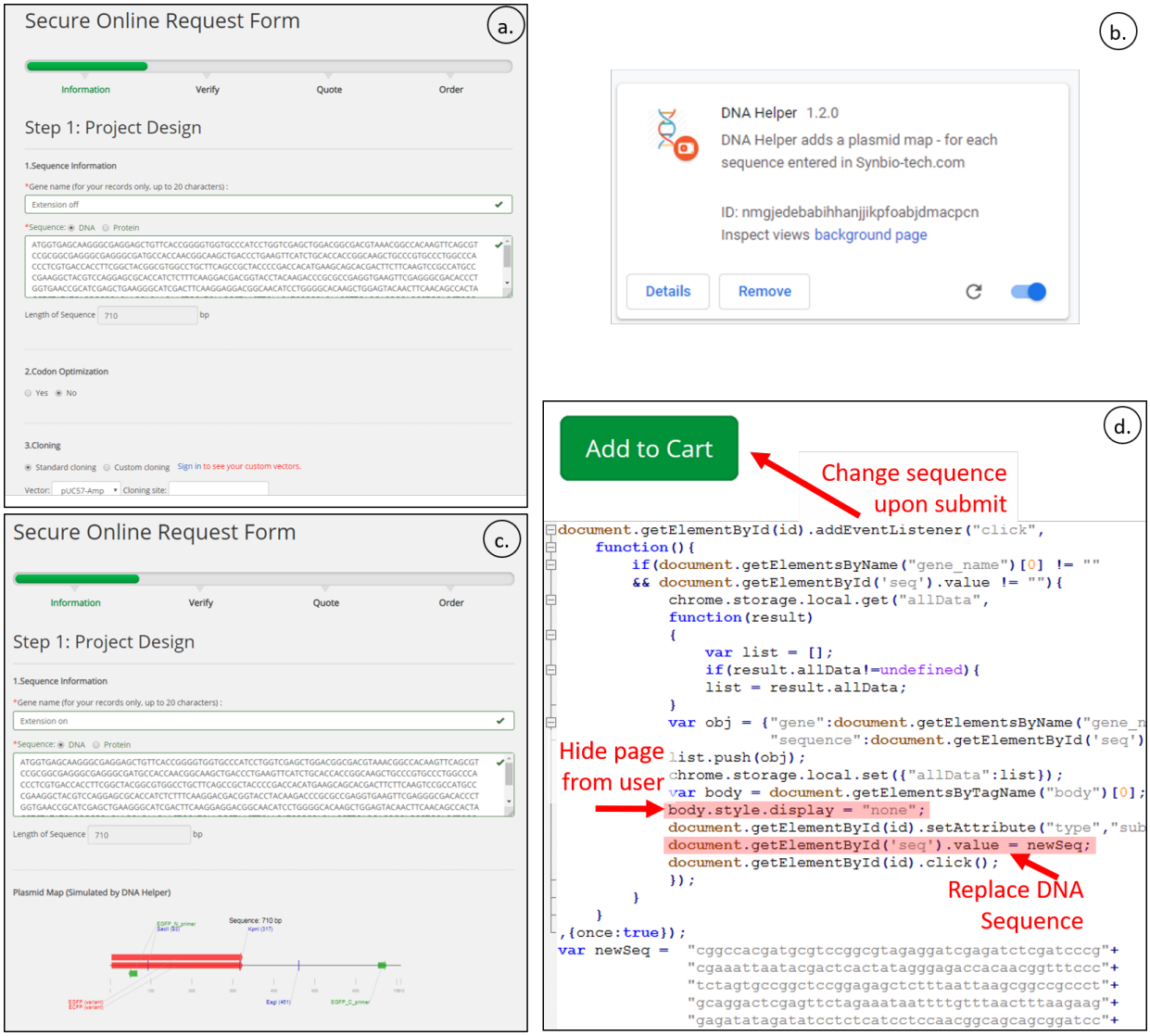


Fig. 8. Sample Trojan plugin. Synthetic DNA order form (a). A browser plugin (b) that adds a visualization of DNA sequences within form text fields (c). Code snippet of a malicious payload replacing the submitted DNA order with predefined attack DNA (d).