Securing the Future of Biotechnology: A Study of Emerging Bio-Cyber Security Threats to DNA-Information Systems

Peter Martin Ney

Executive Summary (see here for full dissertation):

DNA technology, and biotechnology more generally, has been revolutionized by the integration of biotechnology with computer systems. Examples of this change can be seen throughout the biotechnology industry and include the rise of computerized, high-throughput bio-sensors and molecular synthesis machines, automated robotic wet lab equipment, and complex computational pipelines that process biological data. This trend is especially apparent with systems that manipulate or process DNA data, due to the information properties of DNA, which can be read, written, or stored like any information. Consider the exponential decrease in the cost of DNA sequencing that has driven down the price to sequence the human genome by five orders of magnitude in less than 20 years. This advance was made possible with the development of high-throughput sequencing techniques that heavily rely on computers: flow cells that process DNA samples are imaged by computers to sequence of millions of DNA strands simultaneously and the resulting sequencing data is processed and interpreted by an extensive pipeline of bioinformatics software.

This dissertation is concerned with emerging cybersecurity issues that arise from the increasing computerization of biotechnology. There is a long history of computer security issues arising when technology is integrated and connected to computer systems and networks that has affected technologies as diverse as automobiles, implanted medical devices, industrial control systems, voting machines, and IoT devices. Similar to these other domains, I believe that the increasing computerization, connectedness, and access to biotechnology will pose new,

industry-wide cybersecurity risks. These security issues are in contrast to more traditional biosecurity concerns that focus on the risks of dual-use technology, particularly the creation of dangerous pathogens and toxins. I use the term *bio-cyber security* or *CyBio* to describe the emerging cybersecurity risks to biotechnology because new vulnerabilities emerge from the unique interaction of *biotechnology* with *computers*.

I believe that bio-cyber threats will follow attack paradigms that are well known in computer security but have not been considered in the context of biotechnology. Examples include the use of synthetically derived biomolecules, like DNA, to inject malicious code into computer systems that process biological data, side-channel vulnerabilities or information leakage attacks against bio-sensing instruments like DNA sequencers, the spoofing of biomolecules or biological information when it is used for authentication or identification, the remote compromise of robotic wet lab equipment, and adversarial machine learning attacks against biological processing pipelines. Bio-cyber threats will be more pressing as biotechnology is expected to become more ubiquitous, especially in security sensitive applications like personalized medicine and forensics, and as the public has greater access to biotechnology through open DIY bio makerspaces and popular consumer facing applications, like direct-to-consumer genetic testing.

This dissertation provides a framework for the computer security and bioengineering community to better understand and prepare for future bio-cyber security threats. To achieve this goal, I present three studies that demonstrate different bio-cyber attacks against DNA-based technology and describe possible defenses to mitigate these emerging bio-cyber security issues. Together, the results in this dissertation demonstrate that previously unconsidered bio-cyber security risks are not only possible but constitute an industry-wide security problem.

DNA Molecules as a Vector for Computer Malware

High-throughput DNA sequencers take a DNA sample as input and output digital DNA data files containing the sequencing data. Therefore, a sequencer acts as an interface between molecular information, encoded in DNA molecules, and digital information, stored in sequencing data files. It is well known that computer programs that process data are vulnerable to code injection if that data is not properly checked and filtered, especially if the program is written using a memory unsafe language. In this study I demonstrate that the DNA sequencing and data processing pipeline is, in fact, vulnerable to malicious code injection via unsanitized input through an attack vector that is not present in traditional computer systems: input from physical DNA molecules into the sequencing pipeline.

I show that it is possible to produce synthetic DNA molecules, that when sequenced, result in a malicious data file. Malicious sequencing files could be used to compromise software running on the sequencing instrument or downstream programs that are used to analyze sequencing data. I refer to a computer exploit that originates from DNA molecules as a *DNA-to-cyber* exploit, or more generally as a *molecular-to-cyber* exploit, if it comes from other kinds of molecules. I also show that DNA processing software has not been written with security best practices and contains latent vulnerabilities, likely because developers have not considered the possibility of computer exploits to DNA sequencers or downstream analysis software.

To demonstrate the feasibility of DNA-to-cyber exploits I attempted an end-to-end compromise of a sequencing pipeline — in other words, compromise a machine analyzing data that originates from sequencing a synthetic DNA molecule encoding malware. The focus was to understand the challenges that are unique to crafting DNA-to-cyber exploits and not on the well studied problem of finding exploitable vulnerabilities in software; therefore, I set out to exploit an

existing buffer overflow vulnerability that was inserted into a downstream processing program, using DNA that can be synthesized from a low cost synthesis service. Developing executable shellcode — the computer code in malware — in DNA is non-trivial because such DNA is difficult to synthesize and may not execute when processed by computer programs due to the stochastic nature of DNA sequencing.

The first attempt to encode standard shellcode in DNA resulted in difficult to synthesize DNA strands: common shellcode constructs, like stack addresses and NOP-sleds, create homopolymers — long runs of repeated bases that cause errors in synthesis; repetitive elements in shellcode result in DNA molecules that world fold in on themselves and form secondary structures; and the stability of shellcode strands was affected because of imbalanced GC-content. To address these issues I relied on an obscure ASCII-based shellcode that can be combined with a return-to-libc attack to gain a reverse shell via a TCP/IP socket, thereby giving an attacker remote control of the computer. Randomness in the sequencing process also caused issues because reads are randomly ordered and the read length is limited to around 300-600 DNA bases on most sequencing instruments. Therefore, it was necessary to work with very short shellcodes that could fit within a single read (~75 bytes). The final functioning exploit, which satisfied all synthesis and sequencing constraints, fit within a single 177bp read. This exploit strand was ordered using IDTs gBlock service and was sequenced with an Illumina NextSeq. When the resulting sequencing data file was processed by the vulnerable downstream program, it resulted in remote access to analysis machine (Figure 1).

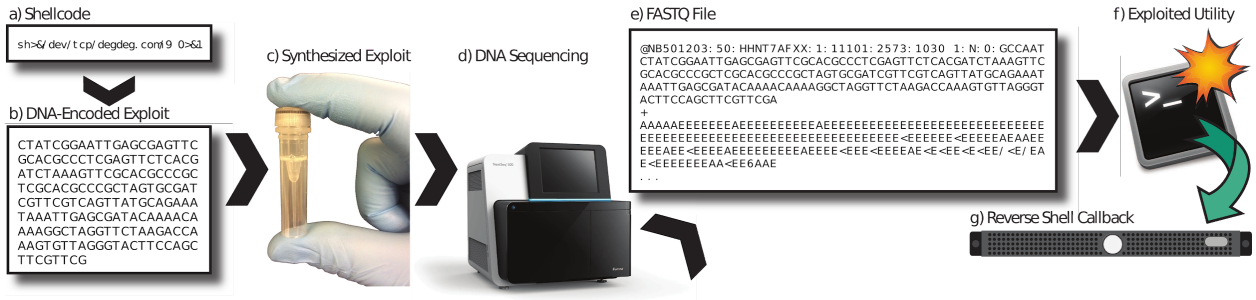

Figure 1: End-to-end Demonstration of a DNA-to-Cyber Exploit. The computer shellcode (a) is first converted into a DNA encoding (b) and then synthesized into physical DNA strands (c). After sequencing the DNA strands (d) this results in a malicious sequencing file (e) that when processed by a vulnerable analysis program (f) results in remote control of the computer (g).

The end-to-end compromise demonstrates that it is possible to overcome synthesis and sequencing challenges to produce DNA encoded shellcode; however, the shellcode I developed was not robust to sequencing errors and does not generalize to popular DNA processing pipelines. To understand the feasibility of these attacks to more typical software it was necessary to know the software security practices of downstream sequencing software. I evaluated the security practices of popular downstream sequencing programs using manual and automated methods and found that popular bioinformatics programs contained insecurities, including numerous buffer overflow vulnerabilities. As DNA-to-cyber exploits become more practical — which technological improvements in DNA sequencing and synthesis suggest — the security practices of DNA processing programs will make this vector an emerging risk to consider in security sensitive applications.

Corrupting Genetic Interpretation Using Side-Channel Vulnerabilities in DNA Sequencers

Next-generation DNA sequencers support multiplex sequencing, a method to sequence many independent samples simultaneously, to increase the overall throughput of the sequencing pipeline. The DNA fragments from each sample are barcoded with short DNA indexes that are 6-8 bases in length. These barcodes are appended to each fragment and are used to identify the samples corresponding to each strand when multiple samples are pooled and sequenced

together. While multiplex sequencing lowers costs, it increases sequencing errors because reads are occasionally demultiplexed into the wrong sample. These demultiplexing errors, called index cross-talk, occur because of barcode sequencing issues, flaws in the library preparation protocol, or when clusters overlap on the flow cell. In this study I show how index cross-talk is, in fact, an example of a well known class of vulnerabilities in cryptographic or computer systems called a side-channel vulnerability; vulnerabilities that unintentionally reveal information about the internal state of a system and can lead to data theft or other security issues. In this work I experimentally demonstrate how an attacker can use the index cross-talk side-channel to manipulate the genetic interpretation of other independent genetic samples.

Index cross-talk inadvertently leaks sequencing information between samples that are intended to remain independent and isolated — DNA sequencing reads leak both to and from all samples — and this leakage can be leveraged by an adversary to steal or manipulate the analysis of other samples. To demonstrate the feasibility of attacking this vulnerability I attempted to manipulate a medically relevant genetic variant in a wild-type human exome sample by sequencing it concurrently with a malicious DNA library. The malicious DNA library was designed to appear like the most common variant responsible for sickle-cell anemia; it was a 400-bp construct surrounding the human beta globin gene (*HBB*) that contained the sickle cell SNP and was highly amplified.

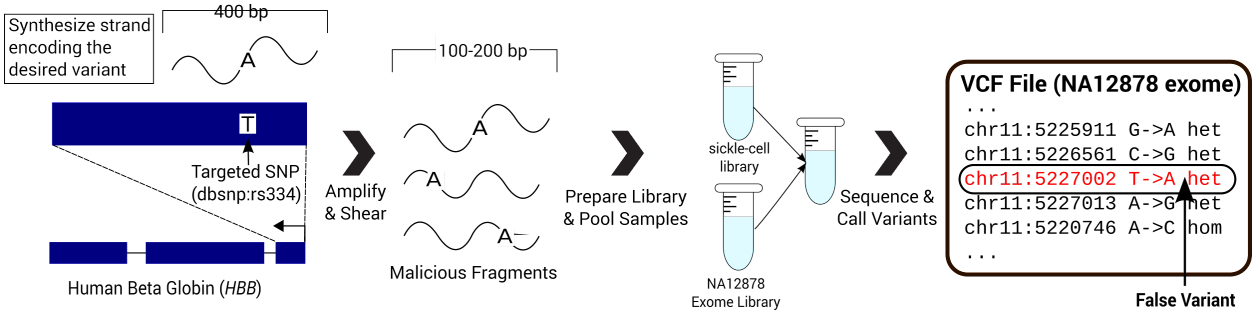

Figure 2: Design of a Malicious DNA Library. The adversary chooses a target genetic mutation (e.g., sickle-cell)

and orders a synthetic DNA strand containing this mutation. The malicious library is highly amplified and then sequenced alongside a human DNA sample. This results in a false sickle-cell variant call in the human sample.

When the malicious library was multiplex sequenced with the human exome sample, enough of the sickle-cell reads appeared in the exome sequencing data that it caused the variant to change from wild-type to heterozygous sickle cell trait in the exome sample (Figure 2). This false variant was called with very low levels of index cross-talk, which implies that this attack is effective even with very low levels of demultiplexing error. This experiment demonstrates that an adversary with a cheap and simple-to-design synthetic strand can manipulate arbitrary variants in independent sequencing samples. This attack could be done whenever an adversary can control samples that are sequenced alongside other independent samples, like at sequencing facilities, medical testing centers, or in forensic applications.

To resolve the security issues posed by index cross-talk side-channel vulnerabilities I tested two possible mitigations. If the species of all samples are the same and known in advance then the average read depth across all samples can be used to detect whether a particular genetic loci (e.g., the sickle cell variant) is being targeted for manipulation. Reads mistakenly assigned into the incorrect sample can also be effectively filtered using the quality score information of the read without affecting the genetic interpretation at other loci (Figure 3).

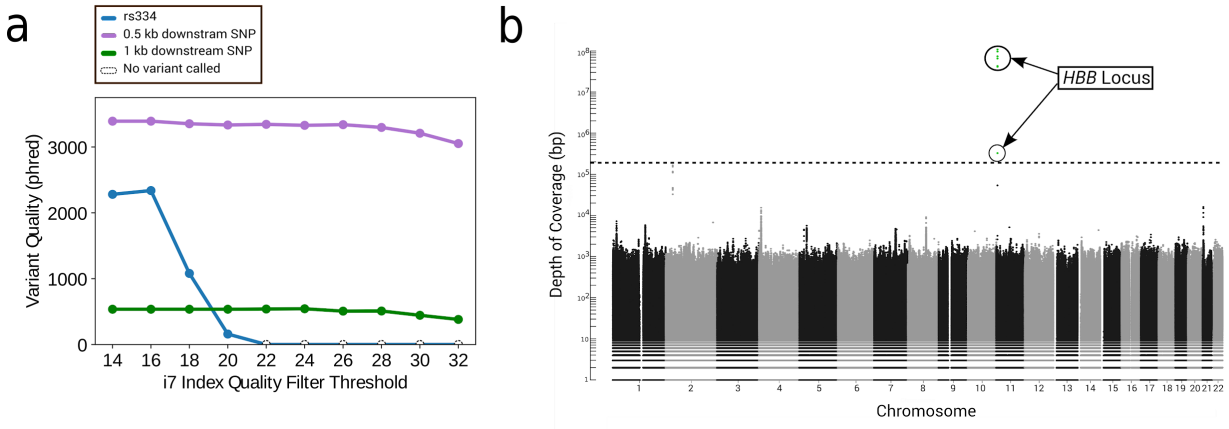

Figure 3: Defenses to Index Cross-Talk Attacks. Pane A: Filtering by index quality score reduces the quality of the attacked variant (rs334) relative to other non-attacked variants, which are unaffected. Pane B: Read depth of all

sequencing reads when aligned to the human genome. The attacked HBB position has 3 orders of magnitude more reads than any other position, and thus, is easily detected.

Security of Consumer Genetic Genealogy Databases

Over 15 million people have taken direct-to-consumer (DTC) genetic tests — popularized by companies like 23andMe and AncestryDNA — to learn about their ancestry, find genetic relatives, or better understand their health. It is common for customers of DTC genetic testing services to download their unprocessed genetic results and upload them to third party companies that offer additional analyses and features. One of the most popular third party services is GEDMatch, which specializes in genetic genealogy, a field that uses genetic data and other genealogical information like family trees to find genetic relatives. GEDMatch maintains a large database with around 1 million genetic profiles and offers software tools so that users can find distant relatives present in the GEDMatch genetic database.

In April 2018, it was revealed that law enforcement had been using the GEDMatch database to solve criminal cold cases. Genetic samples found at crime scenes were processed to look like DTC genetic data files and then uploaded to GEDMatch. Genetic genealogy programs were used to identify relatives of the suspected individual, which greatly narrowed the search for the suspect. The repurposing of genetic genealogy services for identification was controversial, but in this work I consider how the growth of genetic genealogy services raises previously unconsidered security issues. Using the GEDMatch platform I experimentally demonstrated that the design of some third party services exposes them to a number of security risks including the theft of private genetic data and the ability to spoof genetic data files designed to manipulate genetic genealogy or forensic analysis.

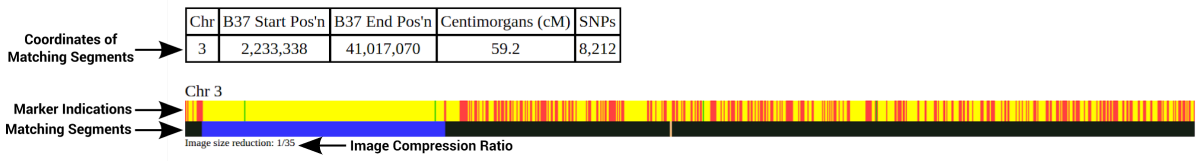

Figure 4: Chromosome Visualization. Chromosome visualization for chromosome 3 resulting from a comparison of two kits. The visualization includes the coordinates of the matching segments and colored bars representing how the two kits compare across the chromosome.

GEDMatch allows users to compare DTC genetic data files (called kits on GEDMatch) to reveal genetic relationships between users. These comparisons reveal the genetic segments that are shared between two kits, which provides an accurate estimate of relatedness. The results of genetic comparisons are displayed to users with chromosome visualizations that show where the two kits match — GEDMatch allows any two public kits to be compared but the underlying genetic markers (called SNPs) remain private (Figure 4).

I show that these chromosome visualizations leak information about the underlying private SNPs in each kit, and demonstrate that it is possible for an adversary to use this information leakage to steal raw SNP information from any user on GEDMatch. An adversary can construct 10-20 fake kits and compare these kits to any target user to reveal 94% of the user's raw SNPs with over 99% accuracy (Figure 5). This example illustrates how simple design choices, like the use of particular visualizations, can expose real services to major security and privacy risks.

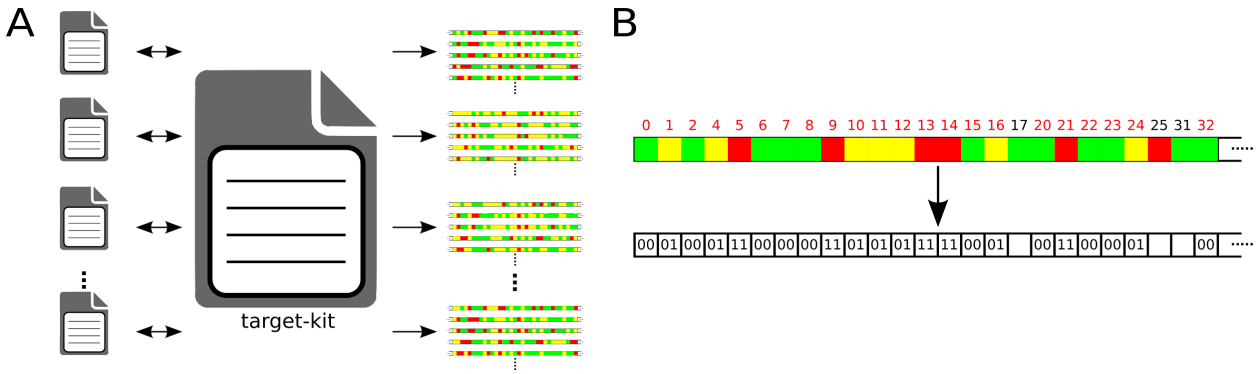

Figure 5: Method to Extract a Target Kit's Genotype. Overview of how the genotype of a targeted kit can be extracted. Pane A: An adversary compares 10-20 specifically designed extraction kits against the target kit, which generates chromosome visualizations that are saved by the adversary. Pane B: The adversary uses the resulting visualizations to infer the genotype of the target kit (see the full dissertation for details on how this algorithm works).

In a second experiment I show that an adversary also has the ability to manipulate the GEDMatch databases to create falsified genetic relationships. An adversary can design fake kits that appear like relatives to any target user on GEDMatch, of any desired relatedness. These kits can be combined with forged metadata (e.g., name and email address) to disrupt genetic genealogical searches. The ability to construct falsified relatives is a powerful technique that, in certain circumstances, can allow an adversary to avoid identification in a criminal investigation or be used to create false relationships designed to defraud victims.

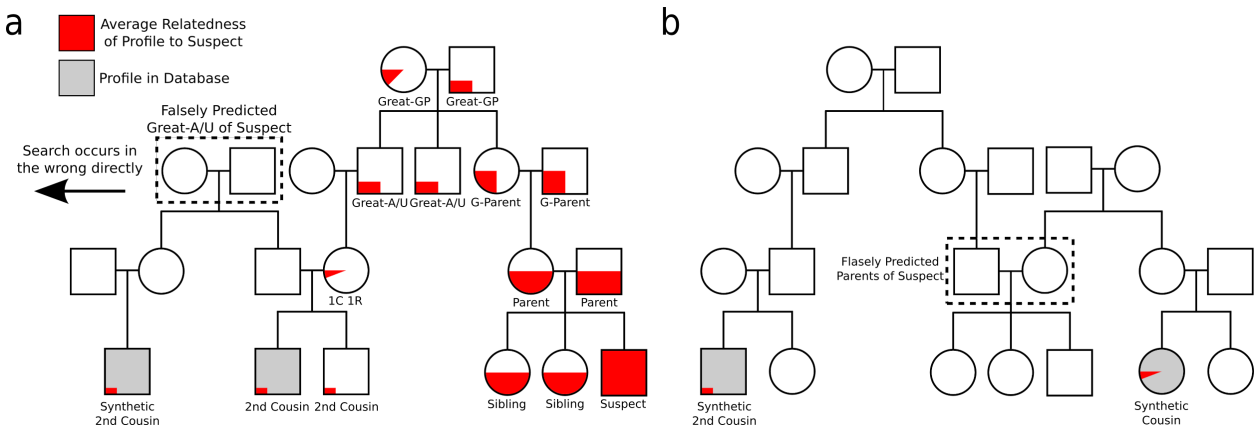

Figure 6: Possible Attacks with Forged Relatives. Pane A: A suspect wants to avoid identification in a genetic genealogy investigation. The suspect uploads a falsified profile under an identity of another person to cause an incorrect genealogy inference. Pane B: The suspect uploads two kits on different branches of the family tree, which causes an incorrectly prediction for the identity of the suspect.

These attacks demonstrate that genetic genealogy services have significant and pressing security problems. However, there are lessons from cybersecurity that can be used to greatly improve the security of GEDMatch and other third party services. The data extraction and false relative attacks are possible because an attacker can construct, upload, and make queries with arbitrary or falsified genetic data, with no requirement that the genetic data originated from a DTC testing company. One solution is to leverage cryptographic digital signatures: digital signatures can be used to ensure the authenticity of the genetic data files that are uploaded to

third party services. Third party websites can be restricted to only accept files that have been signed by a legitimate DTC company.

Conclusion

My dissertation ends with suggestions to bioengineers on how they can design more secure systems and can create a stronger security culture. A key lesson from computer security is that an early exploration of cybersecurity issues leads to safer and better designed technology. By addressing these cybersecurity issues now we can ensure a safer future for biotechnology.