An Introduction to Protein Cryptography

Hayder Tirmazi[1†*], Tien Phuoc Tran[†°],

1 City College of New York

†These authors contributed equally to this work.
* hayder.research@gmail.com ° phuoctran@fas.harvard.edu

Abstract

We introduce protein cryptography, a recently proposed method that encodes data into the amino acid sequences of proteins. Unlike traditional digital encryption, this approach relies on the inherent diversity, complexity, and replication resistance of biological macromolecules, making them highly secure against duplication or tampering. The experimental realization of protein cryptography remains an open problem. To accelerate experimental progress in this area, we provide an accessible and self-contained introduction to the fundamentals of cryptography for biologists with limited mathematical and computational backgrounds. Furthermore, we outline a framework for encoding, synthesizing, and decoding information using proteins. By enabling biologists to actively engage in the development of protein cryptography, this work bridges disciplinary boundaries and paves the way for applications in secure data storage.

Introduction

Messages have been encoded in some form since antiquity. In ancient Egypt, messages were written on tombs in an encoding, replacing unusual hieroglyphs in place of commonly used ones [1]. The ancient Chinese military collection, *Wu-ching tsung-yao*, recommends an encoding that maps items such as requests for bows and arrows to the ideograms of a poem [1]. The scribes of ancient Mesopotamia encoded their names into numbers [1]. Ancient Indian literature discusses secret writing, including the *Arthasastra* and *Kamasutra* [1]. The Arabs were the first to systematically write down cryptography methods and discover methods of breaking cryptographic protocols, a science called cryptanalysis [1]. There is a section on cryptography in the *Subh al-a 'sha*, an Arab encyclopedia completed in 1412 [1]. Modern cryptography can be traced back to the Second World War. British cryptographer Alan Turing and Polish cryptographer Marian Rejewski helped crack the German Enigma machine. Claude Shannon's 1945 paper *A Mathematical Theory of Cryptography* [7] is a foundational work in modern cryptography.

Cryptography Preliminaries

In cryptography terminology, we have a sender and a receiver (conventionally called Alice and Bob). Alice wants to send a message to Bob without having the message eavesdropped by an *eavesdropping adversary*, Eve. Eve can look at the message as it is being sent. The message in its original readable form is called the *plaintext*. A *ciphertext* is the same message is put in a state where it is difficult to read for Eve. The act of

converting plaintext to ciphertext is *encryption*. The reverse process is *decryption* [5]. To encrypt or decrypt a message, Alice and Bob need extra piece of information called a *key*. If Alice and Bob use a common secret key, that can encrypt and decrypt the message, they are using a *symmetric* cryptographic protocol [2,3]. One such symmetric protocol that guarantees perfect secrecy is the one-time pad (OTP) [3,7]. In OTP, for the message `HELLO`, Alice and Bob keep a single-use randomly generated key with the same length as the message e.g. `XMCKL`. Alice shifts `HELLO` letter-by-letter using key `XMCKL` (e.g $H + X = O, E + M = Q, ...$) to form the ciphertext `OQNVZ`. Bob shifts the ciphertext back to the plain text using the same key (e.g $O - X = H, Q - M = E, ...$). Eve cannot read the message as she does not have the key.

In situations where the security requirements are not as strict as **perfect** secrecy, other symmetric protocols are possible. Consider the security requirement that Eve should not be able to read the message with expected probability $\geq \frac{1}{N}$. Imagine Alice sends N letters in sequence to Bob. Alice and Bob have a secret key i which is a number uniformly randomly chosen from the set $\{1, \cdots, N\}$. Alice puts the correct message (`HELLO`) in Letter i and $N-1$ decoy messages (`GOODBYE`, `CAT`,...) in the other letters. Since Eve does not know i, Eve has a $\frac{1}{N}$ probability of reading the correct message whenever she intercepts a letter. The larger the number of letters, N, the less likely it is for Eve to be able to find the correct message if she intercepts one letter. We will use a biologically synthesized version of this exact protocol in Sec to introduce protein-based symmetric cryptography.

Biology Preliminaries

In this section, we list pertinent characteristics of proteins that have attracted cryptographers [9] concerning their use as a medium for data storage and potential for cryptography. Briefly, proteins are chains of amino acids. All living systems on Earth use a set of 20 different natural amino acids as monomers to build proteins. Beyond natural amino acids, there is a wide variety of non-natural, biologically orthogonal amino acids, which provide additional *alphabetical* diversity. A protein is synthesized by connecting amino acids via peptide bonds to form a linear sequence called a polypeptide chain, the primary, most basic structure of a protein [4]. Such synthesis can be readily performed via known laboratory procedures. Then, depending on the sequence, the linear chain might fold and assume a three-dimensional structure, often spontaneously. A protein may consist of multiple discrete unfolded and folded domains [11].

The diverse repertoire of amino acids, both the 1D and 3D structures and the resulting physical and chemical properties are features that can be creatively used for data storage and encryption [9]. Most fundamentally, a writer can encode a message m as a distinct sequence of amino acids, which the reader needs to determine via protein sequencing to retrieve the message. We will focus our paper's protocol for the most part on this approach. Proteins are particularly advantageous as tools for cryptography due to several assumptions based on current in challenges in protein sequencing. First, proteins are currently unclonable [8], meaning that they cannot be used as a template to synthesize more copies of themselves. In addition, while DNA provides a template for RNA, which in turn is a template for protein, and both DNA and RNA are clonable polymers, there is no way to make DNA/RNA from proteins. Coupled with this, all existing protein sequencing approaches, described in detail elsewhere, involve destructive data retrieval [9,10]. Illustrating this point is mass-spectrometry, which is the predominant and most efficient approach and one on which this paper's encryption protocol is based. This approach involves chopping up of a protein into smaller fragments, which are then ionized and further fragmented before analysis. Effectively, protein-based messages enable detection of unwanted access.

Proteins and Secrets

Recent work [9, 10] has suggested an implementation of symmetric cryptography based on proteins. This section builds on the data storage mechanism we suggested for proteins in the introduction. We now describe a more specific protein-based approach that provides resilience against an eavesdropping adversary. We may store a protein-encoded message m within a mixture with other proteins, and only the true recipient knows the correct means to purify the right protein for subsequent protein sequencing to retrieve the message. For example, the protein may contain a unique "epitope tag", a discrete sequence with strong binding affinity to an antibody, which ideally does not bind any other proteins in the mixture. A reader with the correct antibody can purify the protein from the mixture, while an adversary can only guess. Individual proteins in a mixture can also be separated via various chromatography and electrophoretic techniques, based on characteristics such as charge, size, pH. The lack of knowledge of the characteristics of the correct protein can frustrate potential adversaries, making them unable to distinguish real from decoy messages. The complexity of protein biochemistry enables a wide variety of strategies for adversary resilience. It is worth noting again that proteins are unclonable, so adversaries have limited guessworks with protein separation, and their eavesdropping can be detected by quantitative loss in protein amount.

We discuss a simple symmetric encryption protocol for protein-based data storage based on our N letter-sending introductory protocol.

Key Generation

Alice encodes a given message m as a polypeptide sequence, p_m. The writer then fuses p_m to a short peptide tag p_t which we call a header sequence. p_t is recognized by a specific predetermined antibody (Ab) k. This antibody, k, will serve as the key for our protocol.

Decoy Proteins

Alice creates a set of $N-1$ **decoy** proteins. Decoy proteins are other proteins that are designed to look like p_m in terms of composition (types of amino acids used) and length (number of amino acids in the chain). While these decoy proteins are similar in overall makeup (e.g., same length and similar amino acid types), their exact sequences differ from p_m. Each decoy protein contains an alternative header sequence different from p_t. These decoy proteins serve as fake messages mixed with the encoded protein to obscure its identity. A vial of this protein mix, consisting of the decoy proteins and p_m, serves as the storage component.

Message Retrieval

The only possible way for Bob to retrieve the message from the protein mix is for Bob to first identify and purify p_m from among the decoy proteins. For this, the reader must know the right header sequence p_t. The reader can employ the unique Ab that recognizes p_t for message retrieval. Finally, once p_m is purified, it can be decoded via mass spectrometry just like we discussed in the introduction. The decoding process is illustrated in Figure 1

The security for the letter-sending protocol we described in the Cryptography Primer Section relied on the low probability of Eve intercepting the correct letter. Similarly, the security of the protein protocol relies on the fact that effective purification of the

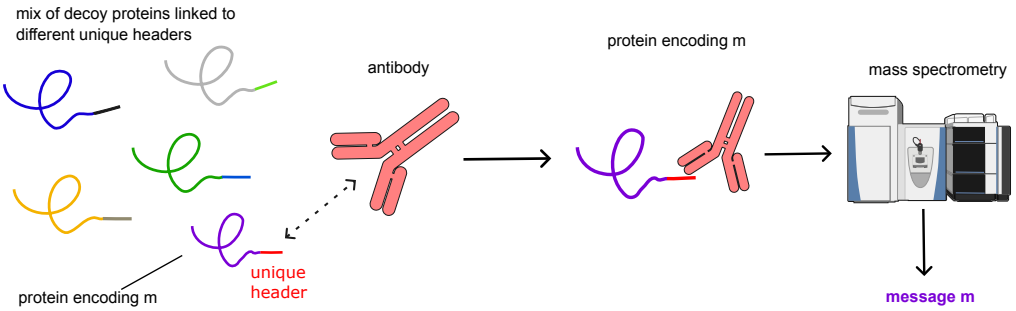

Fig 1. A reader encodes message m from the protein by first using the specific monoclonal antibody to purify the protein and then using mass spectrometry to identify the sequence of amino acids in the protein. This sequence of amino acids can then be decoded back to the plaintext message m.

desired protein from the decoys is improbable through standard methodologies if we do not have the matching Ab [9]. The steps of the protocol are summarized below.

Encryption: On the input of message m, and a mix of $N-1$ decoy proteins.

Step 1 Alice chooses a short peptide *header* that is only recognized by a specific Ab, and shares the Ab with Bob.

Step 2 Alice cryptographically encodes m as a peptide. She fuses the *header* to the cryptographically encoded peptide body and adds the construction into the mix of random peptides.

Decryption: On the input of a specific Ab, and a mix of N proteins.

Step 1 Bob adds the Ab *key* to the protein mixture, enabling the affinity selection and purification of the protein that encoded m.

Step 2 Bob uses mass-spectrometry on the peptide to get the amino acid sequence and uses the sequence to cryptographically decode message m.

Applications

The successful experimental realization of protein cryptography could enable several novel applications. Below are some theoretically possible applications:

Password-Controlled Vaults Protein cryptography can enable the creation of secure data storage devices that self-destruct after a specific number of incorrect password attempts [10]. For example, a user could encode a password into a protein sample, and any unsuccessful attempt to access the stored data would irreversibly destroy part of the material. This mechanism would make brute-force attacks nearly impossible.

One-Time Programs A one-time program is a system that allows a particular function to be executed only a limited number of times before becoming unusable. Using protein cryptography, a protein-based system could be designed to allow access to a specific operation (e.g., running an algorithm or accessing sensitive data) only once or a set number of times [9]. After the allowed usage, the protein sample would degrade, ensuring the function cannot be reused or reverse-engineered.

Password-Authenticated Delegation Protein cryptography could facilitate the secure delegation of cryptographic rights, such as decrypting a message or accessing a resource, to another person who possesses the correct password [10]. The process would rely on encoding the access rights into a protein, which could only be unlocked and used by someone with the specific Ab (the "key").

These applications stand out because they provide a level of physical security that is challenging to replicate with conventional digital methods. Many existing approaches to implementing these functionalities rely on trusted hardware. In contrast, protein cryptography leverages the inherent unclonability and destructive sequencing properties of proteins, offering a uniquely secure alternative. By integrating advancements in molecular biology with cryptographic principles, these applications open new pathways for creating secure, self-contained systems that are resistant to tampering and unauthorized access.

Conclusion

Protein cryptography represents a novel and interdisciplinary approach to secure data storage and encryption. By leveraging the unique properties of proteins—such as their diversity, unclonability, and destructive sequencing—this emerging field offers a biologically grounded alternative to traditional cryptographic methods. While the experimental realization of protein-based encryption remains a challenge, the frameworks and protocols outlined in our work provide a starting point for biologists to engage with this field. Future advancements in protein synthesis, sequencing technologies, and collaborative efforts between biologists and cryptographers could unlock transformative applications, from password-controlled data vaults to self-destructing programs. By fostering deeper integration between biology and cryptography, we aim to inspire innovations that expand the horizons of data security.

References

1. Larew, K. The Codebreakers: The Story of Secret Writing. By David Kahn. (New York: Macmillan Company. 1967. Pp. xvi, 1164). *The American Historical Review*. **74**, 537-538 (1968,12), https://doi.org/10.1086/ahr/74.2.537
2. Pass, R. & Shelat, A. A Course in Cryptography. (2010)
3. Katz, J. & Lindell, Y. Introduction to Modern Cryptography, Second Edition. (Chapman & Hall/CRC,2014)
4. Fowler, S. & Roush, R. & Wise, J. Concepts of Biology. (2013)
5. Rosulek, M. The Joy of Cryptography. , https://joyofcryptography.com, https://joyofcryptography.com
6. Ng, C., Tam, W., Yin, H., Wu, Q., So, P., Wong, M., Lau, F. & Yao, Z. Data storage using peptide sequences. *Nature Communications*. **12**, 4242 (2021,7), https://doi.org/10.1038/s41467-021-24496-9
7. Shannon, C. A Mathematical Theory of Cryptography. (1945)
8. Crick, F. On protein synthesis. *Symp Soc Exp Biol*. **12** pp. 138-163 (1958)

9. Almashaqbeh, G., Canetti, R., Erlich, Y., Gershoni, J., Malkin, T., Pe'er, I., Roitburd-Berman, A. & Tromer, E. Unclonable Polymers and Their Cryptographic Applications. *EuroCrypt 2022*. pp. 759-789 (2022), https://doi.org/10.1007/978-3-031-06944-4_26

10. Almashaqbeh, G. Password-Authenticated Cryptography from Consumable Tokens. *CSCML 2024*. pp. 26-44 (2024), https://doi.org/10.1007/978-3-031-76934-4_2

11. Alberts B, Johnson A & Lewis J & et al. Molecular Biology of the Cell. 4th edition. (Garland Science, 2002)