An Introduction to Protein Cryptography

Hayder Tirmazi[1†*], Tien Phuoc Tran[2†°],

1 City College of New York
2 Harvard University

†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

In cryptography terminology, an *adversary* is an entity we want to secure our message from. The message itself in its original readable form is called the *plaintext*. When the message is put in a state where it is difficult to read for an adversary, it becomes the *ciphertext*. The act of converting plaintext to ciphertext is *encryption*. The reverse process is *decryption* [4]. Symmetric cryptography is a way to keep information safe by using the same secret key to encrypt and decrypt the message. Both people sharing the information need to have the same key to read or protect it [2, 3].

Brief History

Messages have been encoded in some form since antiquity. In Ancient Egypt, messages were written on tombs in an encoding, replacing unusual hieroglyphs in the place of commonly used ones [1]. An 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.

Biological Framework

Proteins are chains of amino acids that can be synthesized via known laboratory procedures. Proteins can be used as a medium for data storage [5, 6]. Protein material consists of a sequence of 20 natural amino acids. A writer can encode a message m as a sequence of these amino acids. A reader may then use mass spectrometry to determine the order of amino acids in the protein. The sequence of amino acids can then be decoded back to the original message m.

Cryptographers have recently become interested in proteins as a tool for building cryptographic primitives [6, 10]. The protocol discussed in this work primarily relies on the following assumption for its security:

Adversary resilience: the only known way to retrieve any information about the data stored in a protein mixed in a vial with other proteins is by purifying the target protein using a monoclonal antibody (or mAb) and then sequencing this protein using mass spectrometry. If an adversary does not know the correct mAb, they can only guess a candidate mAb and check if sequencing will output a message.

Other assumptions that apply to proteins that make them interesting to cryptographers are listed below. Discussing cryptographic primitives that rely on these assumptions is beyond the scope of this paper. We refer the reader to recent papers by Almashaqbeh et al. [6, 9–11] for a detailed treatment of this area as well as its application.

Sequencing vs synthesis: Sequencing a given protein is a more difficult process than synthesizing the protein. Mass spectrometry usually requires pure, macroscopic, and fairly unpolluted samples.

Destructive data retrieval Sequencing a protein is an inherently destructive process. Mass spectrometry necessitates the chopping up of a protein into small fragments which are accelerated in mass spectrometry equipment for detection.

Unclonability: Proteins are (currently) unclonable. There is no existing method of cloning a protein that is provided in a small amount. This was originally hypothesized by Francis Crick in 1958 as the central dogma of molecular biology [8].

Symmetric Encryption

Imagine there are three researchers A, B, and E sharing time slots in a lab. A works in the morning, and B works in the evening, and E works in the afternoon . A and B are collaborating, and A wants to communicate progress to B. They come up with a simple **protocol**: A puts a note on B's desk before A leaves in the morning. When B arrives in the evening, B reads the note. Assume A and B do not trust E. How can they prevent E from reading the note in the afternoon, when neither of them is around? A and B come up with the following second protocol: They buy a locked box with two identical keys and place it in the lab. A and B take one key each. When A leaves in the morning, A unlocks the box using the key, places a note in the box, and then locks it. When B arrives in the evening, B unlocks the box, reads and destroys the note, then

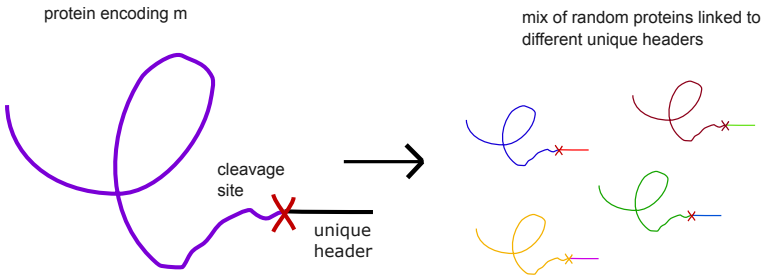

Fig 1. A writer encodes message m into a protein with a unique header. The writer then mixes it with other decoy proteins of similar length and composition. This mixing effectively encrypts the message. A vial containing this protein mixture is the storage component for the message.

locks the box again. E does not have the key, so E cannot read any communication between A and B.

Symmetric encryption that works well in practice can be implemented even without a physical lock and key [3]. Imagine the lab has a bookshelf with N books. A and B can decide a **uniformly random** number i between 1 and N, and hide the note in book i. Since C does not know i, C has to guess which book to look for the note in. C has a $\frac{1}{N}$ probability of guessing the correct book. The larger the number of books, N, the less likely it is for C to be able to find the note. The *secret-key*, in this new protocol, is not a physical object. It is simply something that A and B **know**: the right book to pick. We name this the ChiaKhoa protocol[1]. The steps in the ChiaKhoa protocol are summarized below.

The ChiaKhoa Protocol, a simple symmetric encryption scheme

Encrypt: On the input of message m, a bookshelf with N books

1. Choose a uniformly random number i between 1 and N. Share i with the receiver.
2. Write m on a note
3. Place the note in book i

Decrypt: On the input of a number i between 1 and N, and a bookshelf with N books

1. Choose book i in the bookshelf
2. Read the note inside book i to get m

Proteins and Secrets

Recent work [6, 9–11] has suggested an implementation of symmetric encryption based on proteins. This section builds on the data storage mechanism we suggested for proteins in the introduction. We discuss a simple symmetric encryption protocol for protein-based data storage.

[1] Chia Khoa is *key* in Vietnamese.

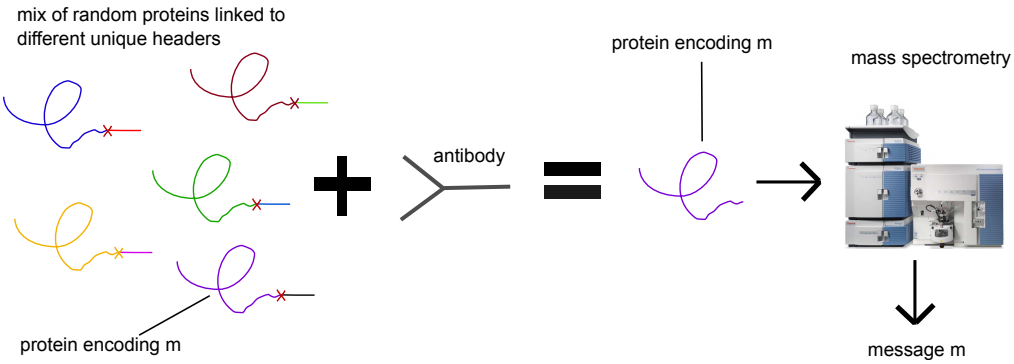

Fig 2. A reader encodes message m from the protein by first using the specific monoclonal antibody to purify the protein and then using mass spectrometry to identify the sequence of amino acids in the protein. This sequence of amino acids can then be decoded back to the plaintext message m.

Key Generation

A writer encodes a given message m as a peptide, 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 monoclonal antibody (mAb) k. This antibody, k, will serve as the key for our protocol.

Decoy Proteins

The writer 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 is conjugated to 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. The entire encoding process is illustrated in Figure 1.

Message Retrieval

The only possible way for a reader to retrieve the message from the protein mix is for the reader 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 mAb 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 2

We explain the end-to-end process by replacing each step in the ChiaKhoa Protocol with a protein-implemented version. We call this new scheme the Miftah Protocol[2]. The security of the Miftah protocol relies on the fact that effective purification of the desired protein from the decoys is impossible through standard methodologies if we do not have the matching mAb [6]. The steps of the Miftah protocol are summarized below.

[2]Miftah means *key* in Arabic.

The MIFTAH Protocol, is a symmetric encryption scheme implemented on proteins.

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

1. Choose a short peptide *header* that is only recognized by a specific mAb. Share mAb with the receiver.
2. Encode m as a peptide. Fuse the *header* to the encoded peptide body and insert a cleavage site between the header and the peptide body.
3. Add the construction into the mix of random peptides

Decrypt: On the input of a specific mAb, and a mix of N proteins.

1. Add the mAb *key* to the protein mixture, enabling the affinity selection and purification of the protein that encoded m. The protease cleavage releases the encoded peptide body.
2. Use mass-spectrometry on the peptide peptide to get the amino acid sequence.
3. Use the amino acid sequence to 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. 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. 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. The process would rely on encoding the access rights into a protein, which could only be unlocked and used by someone with the specific mAb (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. Rosulek, M. The Joy of Cryptography. , https://joyofcryptography.com, https://joyofcryptography.com
5. 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
6. Almashaqbeh, G., Canetti, R., Erlich, Y., Gershoni, J., Malkin, T., Pe'er, I., Roitburd-Berman, A. & Tromer, E. Unclonable Polymers and Their Cryptographic Applications. (Cryptology ePrint Archive, Paper 2022/658,2022), https://eprint.iacr.org/2022/658
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. Password-authenticated Cryptography from Consumable Tokens. (Cryptology ePrint Archive, Paper 2024/1283,2024), https://eprint.iacr.org/2024/1283
10. 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

11. 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