RSA WITH MUTANT PRIME KNOTS

KEVIN XU

Abstract. In this paper, we delve into the intersection of knot theory and cryptography, designing a RSA-style encryption scheme using mutant prime knots. In particular, this system is secure even under attacks from quantum computers, making it a reliable method to send messages back and forth in the near future. Although knowledge of knot theory and cryptography is not assumed in this paper, it is highly recommended for the reader to be familiar with basic terminology. This paper is dedicated to Sarah Fujimori for inspiration.

1. Introduction

As with all papers, we will proceed to hurl a multitude of definitions and properties at the reader with the desperate hope that the paper will seem longer, all in the name of background knowledge. We start with knot theory.

Definition 1.1. A *knot* is an injective, continuous map from $\mathbb{S}^1$ to $\mathbb{R}^3$, where $\mathbb{S}^k$ is a sphere in $\mathbb{R}^{k+1}$. The *unknot*, or *trivial knot*, is just $\mathbb{S}^1$.

Definition 1.2. The *planar diagram* of a knot is the projection of the knot onto $\mathbb{R}^2$ such that every point is mapped from at most two points of the knot. That is to say, no point on the *planar diagram* should be triply intersected.

For convenience, we will denote the set of all knots to be $\mathbb{K}$. The trivial knot, otherwise known as the *unknot*, is just $\mathbb{S}^1$ itself. Another famous knot is the *trefoil knot*, which is shown along with the unknot in the figure below.

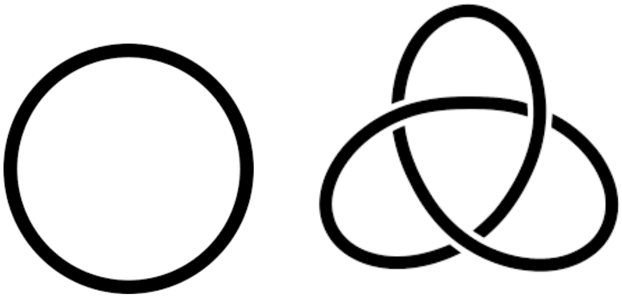

Figure 1. The unknot and trefoil

Next, we define a *crossing* to be a point on a knot where one part of the knot goes over another part of the knot. For example, the unknot above has 0 crossings and the trefoil 3. Now, suppose we pick any point on a knot, and travel either forwards or backwards along the knot until we reach that point again. This direction denotes the *orientation* of the knot.

Date: December 10, 2019.

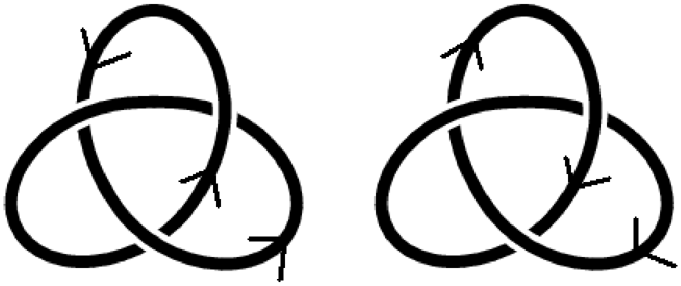

Figure 2. The two orientations of the trefoil

Definition 1.3. Two knots are *similarly-oriented* if they have the same orientation.

We can also combine knots together with the *connected sum*. Note that this is the same as taking the *connected sum* of two topological surfaces.

Definition 1.4. The *connected sum* of two knots k_1, k_2, denoted as $k_1 \# k_2$, is obtained by removing a small segment of each knot and then joining the endpoints together. It can be easily shown that the connected sum of two similarly-oriented knots yields another oriented knot.

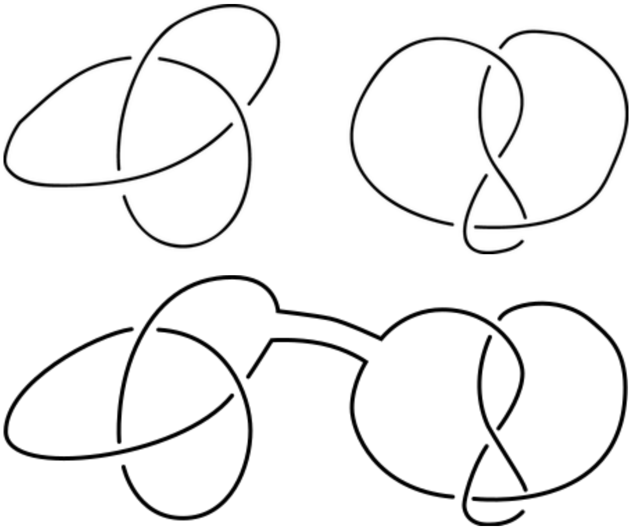

Figure 3. The connected sum

Definition 1.5. A *prime knot* is a knot that cannot be expressed as the connected sum of two non-trivial knots.

Definition 1.6. Two knots are *equivalent* if they are ambient isotopic to each other. That is, it is possible to continuously deform one into the other.

In fact, two knots are equivalent if it is possible to perform a series of moves called Reidemeister moves to change it to the other. One can use this to construct *invariants under knots*, a good way to tell whether a knot is not equivalent to another knot. Some invariants include the crossing number and the Jones and Kauffman knot polynomials, which I have discussed in a previous paper [3]. These are the basics of knot theory, and now we go into specifics relating to our algorithm.

2. Forming RSA with Knots

Definition 2.1. A *tangle* is an open region on the planar diagram of a knot with the property that the knot intersects its boundary at four distinct points.

Proposition 2.2. *Every knot can be represented as two tangles connected by four non-intersecting strands, as shown below. We call this the tangle presentation.*

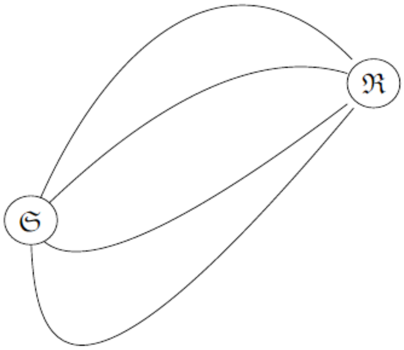

Figure 4. Knot representation with tangles $\mathfrak{N}$ and $\mathfrak{S}$

Proof. Clearly every knot is the connected sum of the unknot and itself. We take the two strands that connect these knots, and "pinch" them together in a way such that there exists an open set only these strands. Then we "fold" one of the strands without intersecting itself or the other strand, forming four non-intersecting strands. Lastly, we "shrink" the open set so that it only contains four non-intersecting strands, splitting the knot into two parts connected by four non-intersecting strands, as desired. ■

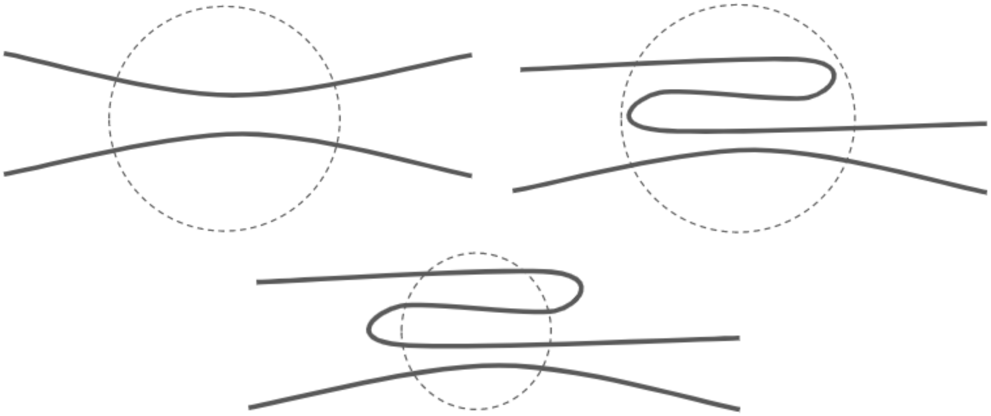

Figure 5. Pinching, folding, and shrinking

Definition 2.3. For an oriented knot k, we denote the *mutant knot* k' as the knot formed by rotating everything inside some tangle $\mathfrak{R}$ in its tangle presentation by π about either the plane formed by the planar diagram of k or the normal vector of that plane.

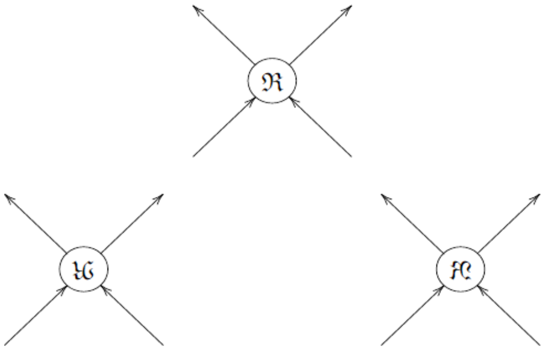

Figure 6. Rotations of the tangle

Rotation of π about the plane can also be thought as taking the mirror image of the tangle about the plane. Above are all the possible rotations of the tangle based on the orientation inside the tangle.

Now we are ready to convert these knots into methods of encryption/decryption. However, we will first need a quick and easy way to represent knots in terms of numbers. Hence, we introduce the Dowker–Thistlethwaite (DT) notation for an oriented prime knot k with n crossings, as follows:

(1) Pick a crossing in k and label it with a 1.
(2) Label the ith crossing you come to with $-i$ if you are *going over* another strand and i is even, and label it i otherwise.
(3) Repeat the previous step until all n crossings have been labeled from 1 to $2n$.

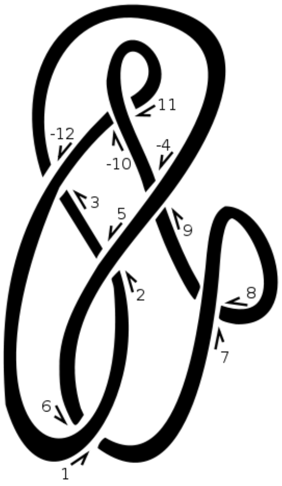

Figure 7. A knot with DT sequence $6, -12, 2, 8, -4, -10$

It is not hard to see that labelling in this way will result in one odd and one even number at every crossing. Then the DT sequence is the sequence of even numbers that share a crossing with $1, 3, 5, \ldots 2n - 1$.

The DT sequence uniquely determines a knot when the knot is *amphichiral*, or equivalent to its mirror image. Otherwise, it represents a knot and its mirror image. However, the proof of this goes beyond the scope of the paper.

The RSA encryption system is built on the concept of an "assymmetric" public key. If Alice wants to send a message to Bob, then the following steps would happen:

(1) Bob generates a public key and sends it to Alice.
(2) Alice uses the key to encrypt her message.
(3) Alice sends her encrypted message to Bob.
(4) Because Bob knows how his public key was generated, he can decrypt the message.

A classic RSA encryption system relies on the difficulty of factoring the product of two large primes. Similarly, we build a system based on the difficulty of "factoring" the connected sum of two prime knots with large crossing number. Suppose that Alice wants to send a message in the form of a finite sequence of knots $L_1, L_2, \ldots, L_n$.

(1) Bob chooses n random prime knots $K_1, K_2 \ldots, K_n$ (taken from some Knot Table). Then he applies a mutation (which can be no change) to each of the knots to form the sequence $K_1', K_2', \ldots, K_n'$. He sends this list to Alice.
(2) Alice takes the mutant prime knots and takes the connected sum $L_i \# K_i'$ for each $1 \leq i \leq n$. She then translates these knots into their DK sequences and sends those to Bob.
(3) Because Bob knows the DK sequences of $K_1, K_2, \ldots, K_n$, he can deconstruct the connected sums and retrieve the original message.

Passing these DK sequences back and forth is as efficient as standard RSA protocol, but using mutant prime knots (which alters the DK sequence in a peculiar way) is secure to quantum-based attacks. In the likely future that more efficient quantum computers are invented, an RSA algorithm built on mutant prime knots may be a secure and efficient way of sending messages.

References

[1] Adams, Colin C. *The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots.* American Mathematical Society, 2010.

[2] Marzuoli, Annalisa & Palumbo, Giandomenico. (2010). Post Quantum Cryptography from Mutant Prime Knots. Computing Research Repository - CORR. 8. 10.1142/S0219887811005798.

[3] Xu, Kevin. *An Introduction to Knot Theory.* Unpublished manuscript. 2019.

Euler Circle, Palo Alto, CA 94306
Email address: kevinxu144@gmail.com