Commutative Algebra and Cryptography

Rashid Zaare Nahandi

Institute for Advanced Studies in Basic Sciences

Overview

1. What is cryptography?
2. What is commutative algebra and algebraic geometry?
3. How to construct a bridge between them?

Cryptography

Cryptography is about converting readable information (plaintext) into an unreadable format (ciphertext) using encryption, and then converting it back to readable form using decryption. This ensures that only authorized parties can access the information.
Modern cryptography is essential for securing digital communications, protecting sensitive data, and ensuring privacy in various applications like online banking, e-commerce, and secure messaging.

Examples

1. Let $u_1 = x^3, u_2 = x^4, u_3 = x^5$. Then J is minimally generated by $\{t_1^3 - t_2 t_3,\ \ t_1^2 t_2 - t_3^2,\ \ t_1 t_2^3 - t_3^3,\ \ t_1 t_3 - t_2^2,\ \ t_2^5 - t_3^4\}$.
2. Let $u_1 = x_1^2, u_2 = x_1 x_2, u_3 = x_2^2$. Then J is minimally generated by $\{t_1 t_3 - t_2^2\}$.
3. (Veroese) Consider all monomials of degree d in $K[x_1, x_2]$ ordered lexicographically. Then J is generated by

$$\{t_i t_j - t_{i+l} t_{j-l} | 1 \leq i < j - 1 \leq n - 1, 0 < l < j - i\}$$

.

All generators are binomials. It is not by accident!

Public and Private Keys

In cryptography, public and private keys are fundamental components of asymmetric encryption.

Public Key

- Purpose: Used to encrypt data or verify a digital signature.
- Accessibility: Can be freely shared with anyone.
- Function: When someone wants to send you an encrypted message, they use your public key to encrypt it. Only your private key can decrypt this message.

Private Key

- Purpose: Used to decrypt data or create a digital signature.
- Accessibility: Must be kept secret and secure.
- Function: When you receive an encrypted message, you use your private key to decrypt it. Similarly, when you sign a document digitally, you use your private key, and others can verify your signature with your public key.

How They Work Together

• Encryption: Data encrypted with a public key can only be decrypted with the corresponding private key, ensuring that only the intended recipient can read the message.

• Digital Signatures: A message signed with a private key can be verified by anyone with the corresponding public key, ensuring the authenticity and integrity of the message.

This system is widely used in various applications, including secure communications, digital signatures, and authentication protocols

Role of Algebra

Algebra plays a crucial role in cryptography, particularly in the processes of encryption and decryption. Here are a few ways algebra is used:

1. Linear Algebra: Techniques from linear algebra, such as matrix multiplication, are used in various encryption algorithms. For example, the Hill Cipher uses matrices to transform plaintext into ciphertext. Each letter of the plaintext is represented as a number, and these numbers are grouped into vectors. These vectors are then multiplied by an encryption matrix to produce the ciphertext.
2. Modular Arithmetic: This is a system of arithmetic for integers, where numbers wrap around after reaching a certain value, known as the modulus. It's fundamental in many cryptographic algorithms, including the RSA algorithm. In RSA, large prime numbers and modular exponentiation are used to create public and private keys.

3 Finite Fields: Algebraic structures known as finite fields are used in cryptographic algorithms like the Advanced Encryption Standard (AES). Finite fields provide a framework for performing arithmetic operations on a limited set of elements, which is essential for creating secure encryption methods.

4 Elliptic Curves: Elliptic curve cryptography (ECC) uses the algebraic structure of elliptic curves over finite fields. ECC provides the same level of security as traditional methods like RSA but with smaller key sizes, making it more efficient.

Commutative Algebra and Algebraic Geometry

Classical Algebraic Geometry is working with systems of polynomial equations. Let K be a field and $K[x_1, \ldots, x_n]$ be the polynomial ring with n variables over the field K. Let

$$T = \begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \quad \vdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

be a system of (nonlinear) equations. Solving these kind of systems (finding roots of T) is known as on of the most difficult tasks in the world!
A known algorithm to solve this system is using Groebner basis.

Example from Cryptography (DLP)

A well-known problem used in cryptography is discrete logarithm problem (DLP):
Let $\mathbb{F}_q$ be a finite field of $q = p^r$ elements for some prime integer p and positive integer r. For a given $\alpha, \beta \in \mathbb{F}_q, \beta \neq 1$. Find (if exists) an integer $1 \leq x < q - 1$ such that $\beta^x = \alpha$ (in $\mathbb{F}_q$).
Finding x is hard. This problem has been used widely to produce public and private keys in cryptography. There are several algorithm to solve DLP. Here we want to transform this problem to a system of polynomial equations.

DLP as a system of PE

Write x in its binary form:

$$x = x_0 + x_1 \cdot 2 + \cdots + x_{n-1} \cdot 2^{n-1}$$

for $2^{n-1} \leq q - 1$ and $x_i \in \{0, 1\}$.
Then

$$\beta^x = (\beta)^{x_0}(\beta^2)^{x_1} \cdots (\beta^{2^{n-1}})^{x_{n-1}} = \beta_0^{x_0} \cdots \beta_{n-1}^{x_{n-1}}$$

where $\beta_i = \beta^{2^i}$ for $i = 0, \ldots, n-1$. We have

$$\beta_i^{x_i} = x_i(\beta_i - 1) + 1 = (x_i + \gamma_i)(\beta_i - 1)$$

where $\gamma_i = \frac{1}{\beta_i - 1} \in \mathbb{F}_q$.

DLP as a system of PE

$$\beta^x = \beta_0^{x_0} \cdots \beta_{n-1}^{x_{n-1}}$$
$$= (x_0 + \gamma_0) \cdots (x_{n-1} + \gamma_{n-1})(\beta_0 - 1) \cdots (\beta_{n-1} - 1).$$

Let $\alpha_1 = \frac{\alpha}{(\beta_0 - 1) \cdots (\beta_{n-1} - 1)}$. Let T be the following system.

$$T_1 = \begin{cases} x_1^2 - x_i = 0, i = 0, \ldots, n-1 \\ y_1 - (x_0 + \gamma_0)(x_1 + \gamma_1) = 0 \\ y_2 - y_1(x_2 + \gamma_2) = 0 \\ y_3 - y_2(x_3 + \gamma_3) = 0 \\ \quad \vdots \\ y_{n-2} - y_{n-1}(x_{n-2} + \gamma_{n-2}) = 0 \\ \alpha_1 - y_{n-2}(x_{n-1} + \gamma_{n-1}) = 0 \end{cases}$$

Solving this system, we find $x_0, \ldots, x_{n-1}$ and therefore we find x.

Example from graph theory

Let G be a simple graph with vertices $v_1, \ldots, x_m$ and edges $e_1, \ldots, l_n$. A matching in G is a collection of edge with no common vertex. A matching i called perfect if it covers all vertices. We want to find all matchings and perfect matchings in G.
We correspond a variable x_i to each edge e_i. In the ring $K[x_1, \ldots, x_n]$ suppose that T_1 is the following system.

$$\begin{cases} x_1^2 - x_i = 0, & i = 1, \ldots, n \\ x_i x_j = 0 & \text{if} \quad e_i \cap e_j \neq \varnothing \end{cases}$$

Solving system T_1, we find all matchings in G. And

$$T_2 = \begin{cases} x_1^2 - x_i = 0, & i = 1, \ldots, n \\ x_i x_j = 0 & \text{if} \quad e_i \cap e_j \neq \varnothing \\ x_{j_1} + x_{j_2} + \cdots + x_{j_s} & v_j \in e_{j_i}, i = 1, \ldots, s, (\textit{all edges}) \end{cases}$$

Solving system T_2, we find all perfect matchings in G.

Question and suggestion

We may use systems of polynomial equations over a field simply as $\mathbb{Z}_2$, to produce secure public and private keys. There are some methods in Commutative Algebra to measure difficulty degree of solving a system. Castelnuovo-Mumford regularity is one of them.

The End

Thanks for your attention