KATHMANDU UNIVERSITY

Department Of Mathematics

Modern Algebra(MATH 402)

REPORT ON

Multivariate Signature Schemes

January 1, 2024

Submitted by:

Samrajya Raj Acharya(01)

Anisha Lamsal(11),

Mukesh Tiwari(22),

Supervised by:

Dr. Gokul K.C.

1 Acknowledgements

We would like to express our sincere gratitude to our instructor Prof. Dr. Gokul KC for his exceptional guidance, unwavering support, and insightful feedback throughout the course of this research on multivariate signature schemes. His expertise in the field and dedication to the subject matter have been very useful in shaping the direction and depth of this exploration. The knowledge and skills acquired in understanding and delving into this subject matter which is a topic of grave importance today has been invaluable, and we are thankful for the opportunity.

Our gratitude extends to all who have contributed directly or indirectly in the shaping of this project.

Thank you.

2 Abstract

Shor's Algorithm, unveiled in 1994, assisted in an era where quantum computers could efficiently factorize numbers. With quantum hardware becoming tangible, it has become increasingly imperative to seek cryptosystems that don't rely on number theory's traditional properties such as factorization and discrete logarithms to ensure data security and robust digital signatures in the post-quantum landscape. We explore the multivariate signature schemes which offers versatile applications in asymmetric signatures, encryption, and authentication. Along with which we look at the python implementation of signature schemes such as C*.

***Keywords**: quantum,HFE,C*,post-quantum cryptography*

Contents

3 Introduction

In 1994, Peter Shor proposed an algorithm which solves number theoretic problems such as the integer factorization problem and the discrete logarithm problem in polynomial time on a quantum computer. This algorithm has therefore the potential to solve the mathematical problems underlying RSA and ECC efficiently.It has demonstrated the potential to efficiently factorize numbers, posing a significant threat to traditional cryptographic methods.The alternatives to the classical cryptographic schemes, which are not affected by quantum computer attacks and therefore can replace these schemes in a post-quantum era are called post quantum cryptosystems.This report embarks on a comprehensive exploration of this quantum shift, navigating through the intricate realms of quantum computing, quantum algorithms, and multivariate cryptosystems to find a path toward secure digital communication in the post-quantum era.

As we navigate through the mathematical tapestry of quantum computing and multivariate cryptosystems, our focus extends to two notable characters in this cryptographic field: C* and HFE.C* introduces a unique signature scheme, adding another layer to our exploration of post-quantum cryptographic solutions.Alongside C*,we acquaint with the Hidden Field Equations(HFE) cryptosystem,seeking to unravel its potential as a key system in the post-quantum cryptographic era.As the main families of post-quantum cryptosystems are:lattice-based, hash-based and multivariate cryptosystems,these schemes are based on multivariate polynomial cryptosystems.

In the convergence of theory and practice, this report delves into the C* and HFE,solving their mathematical intricacies.We navigate the transition from abstract principles to computer implementation.

4 Description

4.1 Cryptography

Cryptography is the science and practice of secure communication. It involves techniques and methods to encode information in a way that only authorized person can access and understand it.The goal is to ensure confidentiality, integrity, and authenticity of data in various forms, whether it's messages, files, or transactions. Cryptography uses mathematical algorithms and keys to transform plaintext into ciphertext and back, providing a layer of security in the digital realm.

Techniques: There are two main types of cryptography: symmetric and asymmetric. Symmetric involves a single key for both encryption and decryption, like a secret code that both the sender and receiver know. Asymmetric uses a pair of keys, public and private, where what one key encrypts, only the other can decrypt. This is like having a lock with one key that everyone knows (public key) and another key that only you know (private key).

Modern Cryptography: Nowadays, we use complex algorithms and mathematical functions to secure our communications. AES (Advanced Encryption Standard) is a symmetric encryption algorithm widely used today, while RSA and ECC (Elliptic Curve Cryptography) are examples of asymmetric encryption.

Hash Functions: These are like the fingerprints of data. A hash function takes input (or message) and produces a fixed-size string of characters, which is unique to that input. It's a one-way street,changing even a tiny bit of the input drastically changes the has.

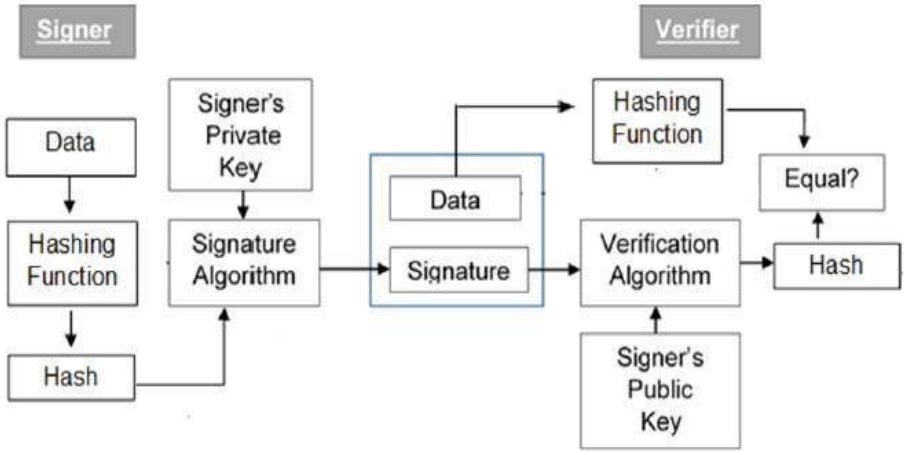
4.2 Cryptosystem

A cryptosystem is a comprehensive framework that combines cryptographic algorithms, protocols, and key management to secure communication and protect information. It encompasses methods for encrypting and decrypting data, ensuring confidentiality, integrity, and authenticity. Cryptosystems can be categorized into symmetric and asymmetric encryption, with symmetric using a single key for both encryption and decryption, while asymmetric involves a pair of keys for secure communication. These systems play a crucial role in safeguarding digital communication,digital signatures,online transactions, and data storage, forming the foundation of cybersecurity in our interconnected world.

Throughout history, various types of cryptosystems have emerged, adapting to the evolving challenges of communication security.For example,Public-key Cryptosystem.

4.3 Digital Signature

There are different types of encryption techniques being used to ensure the privacy of data transmitted over internet. Digital Signature is a mathematical scheme which ensures the privacy of conversation, integrity of data, authenticity of digital message/sender and non-repudiation of sender.It is embedded in some hardware device or also exits as a file on a storage device. Digital Signature are signed by third party some certifying authority.
The digital signature is created by applying a cryptographic algorithm to generate a unique code (hash) for the content. This code is then encrypted using the sender's private key. When the recipient gets it, they use the sender's public key to decrypt and verify the signature, ensuring that the data is intact and genuinely from the claimed sender.



4.4 Classical Cryptosystem

Classical cryptosystems refer to the early methods of encrypting and decrypting messages, primarily before the advent of computers. These systems relied on mathematical principles and basic algorithms to secure communications.Classical cryptosystems were the original methods of encoding messages before the digital age, relying on manual techniques and algorithms.Classical cryptography included substitution ciphers (like Caesar and Vigenère), transposition ciphers (Rail Fence, Columnar Transposition), and even electromechanical devices like the Enigma machine.Classical cryptosystems played a crucial role in securing communication, particularly in military and diplomatic contexts, laying the groundwork for modern encryption.However,vulnerabilities like frequency analysis made some classical cryptosystems less secure, highlighting the need for advancements in cryptography.
While classical systems had limitations, they were pivotal in shaping the evolution of crypto-

graphic techniques, leading to the development of more sophisticated and secure methods like RSA and ECC.

4.4.1 RSA

RSA,named after its inventors Ron Rivest,Adi Shamir,and Leonard Adleman,is a widely used asymmetric (public-key) cryptosystem that revolutionized the field of cryptography.
Key Components:

- Public Key: Used for encryption and is shared openly.
- Private Key: Kept secret and used for decryption.

Algorithms:

- RSA relies on the mathematical properties of large prime numbers. The security of RSA is based on the difficulty of factoring the product of two large prime numbers.

Key generation:

- RSA key pairs are generated by selecting two large prime numbers, multiplying them to get a modulus, and determining the public and private exponents. The security of RSA is closely tied to the difficulty of factoring the modulus.

Encryption and Decryption:

- To send an encrypted message, the sender uses the recipient's public key to encrypt the message. Only the recipient, holding the corresponding private key, can decrypt and read the original message.
- The recipient uses their private key to decrypt the message, revealing the original plaintext.

The security of RSA relies on the difficulty of factoring large semiprime numbers and on the length of the key.

4.4.2 ECC

Elliptic Curve Cryptography (ECC) is a branch of public-key cryptography that leverages the mathematical properties of elliptic curves over finite fields.It is particularly notable for offering strong security with shorter key lengths compared to traditional cryptographic methods.
Key generation:

- Select an elliptic curve defined over a finite field. Choose a base point (generator point) on the curve. Determine a private key, a random integer, as the scalar multiplier. Compute

the public key by multiplying the base point by the private key using elliptic curve scalar multiplication.

Encryption

- Convert plaintext to a point on the curve. Generate an ephemeral key, compute an ephemeral public key, and derive a shared secret. Combine the shared secret with the plaintext for ciphertext.

Decryption

- Multiply the sender's public key by the recipient's private key to compute the shared secret. Use the shared secret to decrypt the ciphertext and obtain the original plaintext message.

In ECC, the security is based on the difficulty of the elliptic curve discrete logarithm problem, providing a robust and efficient approach to secure communication.

4.5 Quantum Computing

Quantum computing was conceived to address the limitations of classical computers in solving certain problems efficiently. Traditional computers face challenges with computational complexity in areas like factoring large numbers, optimizing complex systems, and simulating quantum mechanics.
Quantum computing is a field of study that utilizes principles from quantum mechanics to perform computations. Unlike classical computers that use bits (0s and 1s), quantum computers use quantum bits or qubits.This allows quantum computers to process vast amounts of information in parallel, potentially solving certain problems much more efficiently than classical computers.
Qubits

- Qubits exist in a superposition of states, representing 0 and 1 simultaneously. This superposition enables quantum computers to explore multiple solutions to a problem at the same time.

Entanglement

- Qubits can be entangled, meaning the state of one qubit is directly related to the state of another, even if they are physically separated. Entanglement allows for correlated behavior between qubits.

Quantum Gates

- Quantum computers use quantum gates to perform operations on qubits. These oper-

ations exploit the principles of superposition and entanglement to carry out complex computations.

Quantum Speedup

- Quantum computers have the potential to solve certain problems exponentially faster than classical computers. This includes tasks like factoring large numbers, searching unsorted databases, and simulating quantum systems.

4.6 Quantum Algorithms

Quantum algorithms operate within the framework of quantum computation, which is fundamentally different from classical computation.
It harnesses the principles of quantum superposition and entanglement to achieve computational speedups in specific problem domains.Quantum algorithms provide a quantum leap in computational efficiency for specific problem domains. As we encounter increasingly complex challenges in cryptography,optimization, and beyond,quantum algorithms offer a promising avenue for overcoming classical limitations and unlocking new frontiers in cryptography.
Here are brief details on some of prominent quantum algorithms:
Shor's Algorithm

- Objective: Factorizing large integers exponentially faster than the best-known classical algorithms.
- Key Components: Utilizes quantum Fourier transform and modular exponentiation to identify prime factors efficiently.
- Impact: Threatens classical cryptographic systems relying on the difficulty of integer factorization, such as RSA.

Grover's Algorithm

- Objective: Quadratically accelerating the search of an unsorted database compared to classical algorithms.
- Key Components: Utilizes amplitude amplification to enhance the probability of finding the correct solution in a superposition of states.
- Impact: Poses implications for certain search and optimization problems, though not applicable to all types of problems.

Quantum Approximate Optimization Algorithm (QAOA):

- Objective: Finding approximate solutions to combinatorial optimization problems.
- Key Components: Utilizes a series of quantum gates to evolve a quantum state that encodes the optimal solution.
- Impact: Relevant for optimization challenges in various fields, such as logistics and finance.

Deutsch-Josza Algorithm:

- Objective: Determines whether a given function is constant or balanced in a single quantum query.
- Key Components: Utilizes quantum parallelism to evaluate the function at multiple points simultaneously.
- Impact: Demonstrates a quantum speedup for specific oracle-based problems.

4.6.1 Shor's Algorithm

Shor's Algorithm, named after mathematician Peter Shor developed in 1994,is a quantum algorithm designed to efficiently factorize large composite numbers. It's one of the most famous and impactful algorithms in quantum computing, it is an algorithm that can solve discrete logarithmic problem and Integer factorization problem in polynomial time on a quantum computer. It is one of the few known quantum algorithms with compelling potential applications and strong evidence of superpolynomial speedup compared to best known classical algorithms.
The ability to factor large numbers has significant implications for cryptography, particularly RSA encryption. RSA is based on the assumption that factoring large integers is computationally intractable.However,Shor's algorithm shows that factoring integers is efficient on an ideal quantum computer, so it may be feasible to defeat RSA by constructing a large quantum computer.
The core of Shor's Algorithm is the Quantum Fourier Transform (QFT), a quantum analog of the classical Fourier Transform. The QFT allows the algorithm to find the period of a specific mathematical function related to the number being factored. Once the period is found, the prime factors can be efficiently extracted using classical methods.
While Shor's Algorithm is theoretically efficient,its practical implementation on current quantum computers is challenging. It requires a significant number of qubits and error-corrected operations. Efforts to implement Shor's Algorithm have led to valuable insights into quantum error correction, algorithm optimization,and hardware development. Its discovery has had a profound impact on both the theoretical and practical aspects of quantum computing and cryptography.

4.7 Quantum Cryptosystems

Unlike classical cryptographic systems, which rely on the computational difficulty of problems like integer factorization for their security, quantum cryptosystems harness the unique principles of quantum mechanics to fortify information against potential adversaries armed with quantum computers. The significance of these systems lies in their capacity to provide secure communication channels in the quantum era, where classical cryptographic methods face the risk of being swiftly unraveled by quantum algorithms like Shor's Algorithm.Quantum Key Distribution (QKD) is a prime example of the importance of quantum cryptosystems.
As algorithms like Shor's Algorithm threaten the security of classical cryptographic systems, the need for quantum-resistant alternatives becomes paramount. Quantum-resistant algorithms, also known as post-quantum cryptography, are designed to withstand attacks from quantum computers. Embracing mathematical structures that quantum algorithms find challenging to exploit, these algorithms ensure the continued confidentiality and integrity of sensitive information in the quantum era.
As quantum technologies continue to progress, the importance of quantum cryptosystems in preserving the confidentiality and integrity of sensitive information becomes increasingly evident.

4.8 Multivariate Cryptosystem

Multivariate Quadratic Polynomial system:
Given a quadratic polynomial map $P : F_q^n \to F_q^m$ over a finite field F_q, find $x \in F_q^n$ that satisfies $P(x) = 0$.
Public key : quadratic polynomial map P
Private key : knowledge about inverting the map
let $m \in F_q^m$ be our message. then the signature s is simply $s \in F_q^n$ such that

$$P(s) = m$$

Multivariate Cryptosystem:
The core idea of multivariate cryptosystem is to use systems of equations based on multivariate polynomials as the basis for cryptographic functions. These systems are generally easy to create but hard to invert without specific information, making them suitable for cryptographic applications(NP hard problems).
One popular type of multivariate cryptographic system is the Multivariate Quadratic Equations

(MQ). In MQ cryptography, the public key is a set of multivariate quadratic equations, and the private key is the corresponding set of solutions to these equations. The security is solely based on the MQ Problem of solving a system of multivariate quadratic equations.
While multivariate cryptography offers some intriguing possibilities, it's important to note that it hasn't seen widespread adoption compared to traditional public-key cryptosystems like RSA or ECC. This is partly due to challenges in finding a balance between security and efficiency and the need for further research to fully understand the potential vulnerabilities and strengths of these systems.
Below mentioned C* and HFE are cryptographic schemes based on Multivariate cryptosystem,where C* emphasizes the difficulty of solving systems of multivariate quadratic equations, and HFE (Hidden Field Equations) relies on the complexity of solving systems of polynomial equations over finite fields for secure communication.

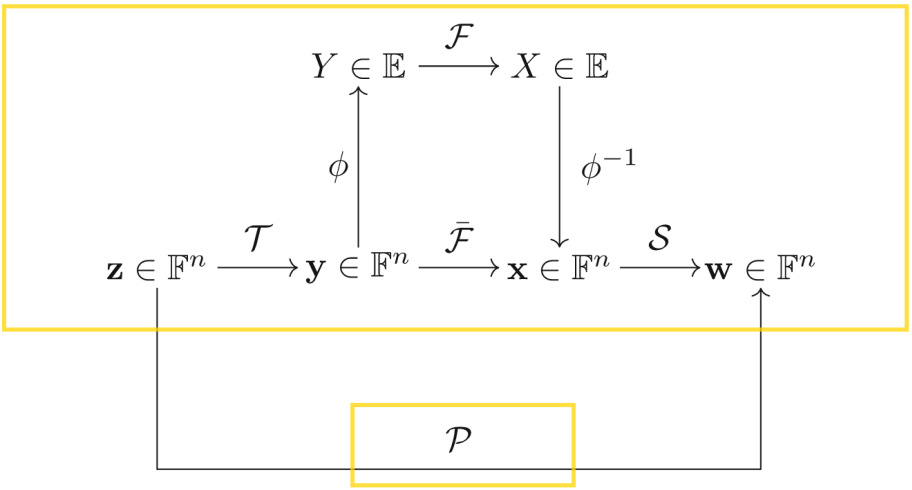
4.9 C-star

The Matsumoto-Imai cryptosystem (short MI or C^*) as proposed by Tsutomo Matsumoto and Hideki Imai in 1988, was one of the first multivariate cryptosystems and has attracted a lot of attention.

The basic Matsumoto-Imai (short MI or C^*) cryptsystem can be described as follows. Let F be a finite field of characteristic 2 with q elements and let $g(X) \in F[X]$ be an irreducible polynomial of degree n over F. Therefore, the field $E = F[X]/g(X)$ is a degree n extension field of F.

Let $\phi : F_n 1 E$ be the standard isomorphism between the vector space F_n and the extension field E, i.e.

$$\phi(x_1, ..., x_n) = n \sum_{i=1}^{n} x_i X^{i-1}$$



The central map $F : \mathbb{E} \rightarrow \mathbb{E}$ of the C^* scheme is a bijective map over the extension field E, which is defined as

$$F(Y) = Y^{q^\theta+1}$$

with $0 < \theta < n$ and $\gcd(q^n - 1, q^\theta + 1) = 1$.

In order to invert the central map F, we use the extended Euclidean algorithm to compute an integer h with $h(q^\theta + 1) = 1 \mod (q^n - 1)$. Therefore, we get

$$F^{-1}(X) = X^h = Y^{h(q^\theta+1)} = Y^{k(q^n-1)+1} = Y.$$

The public key of the scheme is the composed map $P = S \circ \tilde{F} \circ T = S \circ \phi^{-1} \circ F \circ \phi \circ T : F_n \rightarrow F_n$ with two invertible linear maps $S : F_n \rightarrow F_n$ and $T : F_n \rightarrow F_n$, the private key consists of S, h and T . However, we can also assume that h is public since θ is in a very small range.

4.9.1 MI as a Signature Scheme

4.9.2 Signature Generation

To generate a signature for a document d, one uses a hash function $H : 0,1^* \rightarrow F_n$ to compute the hash value $w = H(d) \in F_n$. After that, one computes $x = S^{-1(w)} \in F_n, X = \phi(x) \in E, Y = F^{-1}(X) \in E, y = \phi^{-1}(Y) \in F_n and z = T^{-1}(y)$. The signature of the document d is given by $z \in F_n$.

4.9.3 Signature Verification

To check, if $z = (z1, ..., zn) \in F_n$ is indeed a valid signature for the document d, one computes $w = H(d) \in F_n$ and $w' = P(z) \in F_n$. If $w' = w$ holds, the signature is accepted, otherwise rejected.

4.10 HFE

After the Matsumoto–Imai cryptosystem had been broken by his linearization equations attack,Patarin had the idea of the Hidden Field Equations (HFE) cryptosystem. The general idea is to use a polynomial over an extension field as a private key and a vector of polynomials over underlying finite field as public key. It's security is related to difficulty of solving a random system of multivariate quadratic equation over finite field.

4.10.1 HFE as a Signature Scheme

Although HFE can be used a encryption scheme, the public key sizes are prohibitive. It is more promising as a signature scheme.

Since, the HFE central map is not bijective, not every hash value $w \in F_n$ necessarily has a signature. To overcome this problem, one can use for example a counter $r\ in\mathbb{N}$ during the signature generation process. Instead of generating an HFE signature z for the hash value $w = H(d)$, one generates an HFE signature for the hash value $w = H(d||r)$.If w does not lead to a signature, one increases the counter r and tries again. The final signature σ sent to the verifier has the form $\sigma = (z, r)$.

Steps for Hidden Field Equations Signature Generation

To generate a signature for a document d, one starts with $r = 0$, computes the hash value $w = H(d||r))$ and performs the following three steps.

- Step 1. Invert the first affine map S, i.e. compute $x = S^{-1}(w) \in F_n$ and lift the result to the extension field E, i.e. compute $X = \phi(x)$.
- Step 2. Find a solution $Y \in E$ of the uni-variate polynomial equation $F(Y) = X$ for example via Berlekamp's algorithm or the Cantor–Zassenhaus algorithm. If the equation does not have a solution, increment the counter r, compute the new hash value $w = H(d||r)$ and start again with step 1.
- Step 3. Move the result Y down to the vector space F_n, i.e compute $y = \phi^{-1}()$ and compute the HFE signature $z \in F_n$ by $z = T^{-1}(y)$. Send (z, r) to the verifier.

Steps for Signature Verification

To check the authenticity of a signature (z, r), the verifier computes the hash value $w = H(d||r)$ and $w = P(z)$ .If $w = w$ holds, the signature is accepted, otherwise rejected.

5 Computer Implementation

5.1 Implementation of C* using SageMath

[1]:
```
from random import choices
```

[2]:
```
#Generate a finite field
q = 4
F.<a> = GF(2^2)
F
```

[2]:
```
Finite Field in a of size 2^2
```

[3]:
```
#Define a polynomial ring over the finite field
R.<x> = F[]
R
```

[3]:
```
Univariate Polynomial Ring in x over Finite Field in a of size 2^2
```

[4]:
```
F.gen()
```

[4]:
```
a
```

[5]:
```
#an extension field over the field is generated
n = 3
E.<X> = F.extension(x^3+a)
E
```

[5]:
```
Univariate Quotient Polynomial Ring in X over Finite Field in a of size␣
↪2^2 with
modulus X^3 + a
```

[6]:
```
#Central map that maps E to E
def central_map(X):
    return X^17
```

[7]:
```
# commputation of gcd
g,k,h = xgcd(q**n-1 , q**2 + 1)
h
```

[7]:
```
26
```

[8]:
```
# define the inverse of central map which is our private information
def inv_central_map(X):
    return X^26
```

[9]:
```
# phi is the canonical mapping of n-dimensional vector space over the␣
 ↪finite field to its extension field
def phi(x):
    return x[0] * X^0 + x[1] * X^1 + x[2] * X^2
```

[10]:
```
# affine map S used to hide the central map (kept secret)
Sh = matrix([[a,a,a+1],[0,0,a],[a+1,a,1]])
def S(x):
    return Sh * x + vector([0 , a+1 ,a])
S(vector([0,a,a+1]))
```

[10]:
```
(1, a, a)
```

[11]:
```
#inverse of affine map S (kept secret)
invSh = matrix([[1,1,1],[a,0,1],[0,a+1,0]])
def invS(x):
    return invSh * vector([x[0] - 0, x[1] - (a+1), x[2] - a])
```

[12]:
```
#assert that S and invS are inverses of eachother
for _ in range(100):
    rand_list = choices(F.list(),k=3)
    rand_vector = vector(rand_list)
    assert rand_vector == invS(S(rand_vector))
```

[13]:
```
# affine map T used to hide the central map
Th = matrix([[1,1,1],[a,a+1,1],[a,0,0]])
def T(x):
    return Th * x + vector([1, a+1 ,a+1])
```

[14]:
```
#inverse of affine map T (kept secret)
invTh = matrix([[0,0,a+1],[a+1,a+1,1],[a,a+1,a]])
def invT(x):
```

```
    return invTh * vector([x[0] - 1, x[1] - (a+1) , x[2] - (a+1)])
```

[15]:
```
#assert that T and invT are inverses of eachother
for _ in range(100):
    rand_list = choices(F.list(),k=3)
    rand_vector = vector(rand_list)
    assert rand_vector == invT(T(rand_vector))
```

[16]:
```
# public key :
def Publickey(x):
    x1 = T(x)
    x2 = phi(x1)
    x3 = central_map(x2)
    x4 = vector(x3.list())
    x5 = S(x4)
    return x5
```

[17]:
```
# decryption by Bob using his private key, the components used are only␣
 ↪known to Bob.

def Privatekey(y):
    y1 = invS(y)
    y2 = phi(y1)
    y3 = inv_central_map(y2)
    y4 = y3.list()
    y5 = invT(y4)
    return y5
```

[18]:
```
Privatekey(vector([a,a,0]))
```

[18]:
```
(0, a, a + 1)
```

[19]:
```
# Alice chooses a message to encrypt for Bob
msg = [0,a,a+1]
```

[20]:
```
# to encrypt the message, Alice uses Bob's map which is his public key
cipher = Publickey(vector(msg))
cipher
```

[20]: (a, a, 0)

[21]:
```
# to decrypt, Bob who has all the components of the private key to invert␣
 ↪the public map

recovered_map = Privatekey(cipher)
recovered_map
```

[21]: (0, a, a + 1)

6 Conclusion

The role on cryptography has been deeply influential all through the history. From the early battlefields to the transmissions in World Wars, It has shaped battles. In today's world where the world is connected and in communication at a moments notice, it is indispensable as a aid to protect privacy and confidentiality. In addition to power of computing hard ciphers by the computers, the immense scope for cryptanalysis has emerged hand in hand. Quantum Computers potentially present a whole new dimension and the world is scrambling to cope. Many alternatives to classical schemes have been proposed and one of those is the based on multivariate quadratic equations.

These schemes have come a long way from the very first scheme in 1988. These have been promising for a potential candidate for post-quantum signature schemes. A lot of work in the cryptanalysis aspect is yet to be done and explored.

References

[1] Olivier Billet and Jintai Ding. "Overview of Cryptanalysis Techniques in Multivariate Public Key Cryptography". In: *Gröbner Bases, Coding, and Cryptography*. Ed. by Massimiliano Sala et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 263–283. ISBN: 978-3-540-93805-7 978-3-540-93806-4. DOI: 10.1007/978-3-540-93806-4_15. (Visited on 10/16/2023).

[2] Olivier Billet and Henri Gilbert. "Cryptanalysis of Rainbow". In: *Security and Cryptography for Networks*. Ed. by Roberto De Prisco and Moti Yung. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 336–347. ISBN: 978-3-540-38081-8.

[3] Charles Bouillaguet, Pierre-Alain Fouque, and Amandine Véber. "Graph-Theoretic Algorithms for the "Isomorphism of Polynomials" Problem". In: *Advances in Cryptology – EUROCRYPT 2013*. Ed. by David Hutchison et al. Vol. 7881. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 211–227. ISBN: 978-3-642-38347-2 978-3-642-38348-9. DOI: 10.1007/978-3-642-38348-9_13. (Visited on 10/03/2023).

[4] Ryann Cartor et al. "On the Differential Security of the HFEv- Signature Primitive". In: *Post-Quantum Cryptography*. Ed. by Tsuyoshi Takagi. Vol. 9606. Cham: Springer International Publishing, 2016, pp. 162–181. ISBN: 978-3-319-29359-2 978-3-319-29360-8. DOI: 10.1007/978-3-319-29360-8_11. (Visited on 10/07/2023).

[5] Jiahui Chen et al. "Identity-Based Signature Schemes for Multivariate Public Key Cryptosystems". In: *The Computer Journal* 62.8 (Aug. 2019). Ed. by Joseph Liu, pp. 1132–1147. ISSN: 0010-4620, 1460-2067. DOI: 10.1093/comjnl/bxz013. (Visited on 10/08/2023).

[6] . *Demystifying Multivariate Cryptography*. Aug. 2023.

[7] Jayashree Dey and Ratna Dutta. "Progress in Multivariate Cryptography: Systematic Review, Challenges, and Research Directions". In: *ACM Computing Surveys* 55.12 (Mar. 2023), 246:1–246:34. ISSN: 0360-0300. DOI: 10.1145/3571071. (Visited on 09/22/2023).

[8] Jintai Ding and Albrecht Petzoldt. "Current State of Multivariate Cryptography". In: *IEEE Security & Privacy* 15.4 (2017), pp. 28–36. ISSN: 1558-4046. DOI: 10.1109/MSP.2017.3151328.

[9] Jintai Ding, Albrecht Petzoldt, and Dieter S. Schmidt. *Multivariate Public Key Cryptosystems*. Vol. 80. Advances in Information Security. New York, NY: Springer US, 2020. ISBN: 978-1-07-160985-9 978-1-07-160987-3. DOI: 10.1007/978-1-0716-0987-3. (Visited on 09/23/2023).

[10] Jintai Ding and Dieter Schmidt. "Rainbow, a New Multivariable Polynomial Signature Scheme". In: *Applied Cryptography and Network Security*. Ed. by John Ioannidis, Ange-

los Keromytis, and Moti Yung. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 164–175. ISBN: 978-3-540-31542-1. DOI: 10.1007/11496137_12.

[11] Jintai Ding and Bo-Yin Yang. "Multivariate Public Key Cryptography". In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg: Springer, 2009, pp. 193–241. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7_6. (Visited on 09/22/2023).

[12] Jean-Charles Faugère and Ludovic Perret. "An Efficient Algorithm for Decomposing Multivariate Polynomials and Its Applications to Cryptography". In: *Journal of Symbolic Computation* 44.12 (Dec. 2009), pp. 1676–1689. ISSN: 07477171. DOI: 10.1016/j.jsc.2008.02.005. (Visited on 10/05/2023).

[13] Jean-Charles Faugère and Ludovic Perret. "Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects". In: *Advances in Cryptology - EUROCRYPT 2006*. Ed. by Serge Vaudenay. Vol. 4004. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 30–47. ISBN: 978-3-540-34546-6 978-3-540-34547-3. DOI: 10.1007/11761679_3. (Visited on 10/03/2023).

[14] Yasufumi Hashimoto. "Multivariate Public Key Cryptosystems". In: *Mathematical Modelling for Next-Generation Cryptography*. Ed. by Tsuyoshi Takagi et al. Vol. 29. Singapore: Springer Singapore, 2018, pp. 17–42. ISBN: 978-981-10-5064-0 978-981-10-5065-7. DOI: 10.1007/978-981-10-5065-7_2. (Visited on 10/07/2023).

[15] Aviad Kipnis, Jacques Patarin, and Louis Goubin. "Unbalanced Oil and Vinegar Signature Schemes". In: *Advances in Cryptology — EUROCRYPT '99*. Ed. by Gerhard Goos et al. Vol. 1592. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 206–222. ISBN: 978-3-540-65889-4 978-3-540-48910-8. DOI: 10.1007/3-540-48910-X_15. (Visited on 10/03/2023).

[16] Aviad Kipnis and Adi Shamir. "Cryptanalysis of the Oil and Vinegar Signature Scheme". In: *Advances in Cryptology — CRYPTO '98*. Ed. by Gerhard Goos et al. Vol. 1462. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 257–266. ISBN: 978-3-540-64892-5 978-3-540-68462-6. DOI: 10.1007/BFb0055733. (Visited on 10/03/2023).

[17] Nibedita Kundu, Sumit Kumar Debnath, and Dheerendra Mishra. "A Secure and Efficient Group Signature Scheme Based on Multivariate Public Key Cryptography". In: *Journal of Information Security and Applications* 58 (May 2021), p. 102776. ISSN: 22142126. DOI: 10.1016/j.jisa.2021.102776. (Visited on 10/05/2023).

[18] Tanja Lange. "Multivariate-Quadratic Signatures - Definitions and Basic Concepts". In: . . . 1 ().

[19] Tanja Lange. "Multivariate-Quadratic Signatures - MQ-based Identification Scheme". In: *secret s* 1 ().

[20] Tanja Lange. “Multivariate-Quadratic Signatures III - Hidden-field Equatios”. In: *The G* ().

[21] Dalvir Singh Mandara. “Multivariate Cryptography and Algebraic Techniques Used in Cryptanalysis”. In: (2018).

[22] Tsutomu Matsumoto and Hideki Imai. “Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption”. In: *Advances in Cryptology — EUROCRYPT ’88*. Ed. by G. Goos et al. Vol. 330. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 419–453. ISBN: 978-3-540-50251-7 978-3-540-45961-3. DOI: 10.1007/3-540-45961-8_39. (Visited on 10/30/2023).

[23] Jacques Patarin, Louis Goubin, and Nicolas Courtois. “Improved Algorithms for Isomorphisms of Polynomials”. In: *Advances in Cryptology — EUROCRYPT’98*. Ed. by Gerhard Goos et al. Vol. 1403. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 184–200. ISBN: 978-3-540-64518-4 978-3-540-69795-4. DOI: 10.1007/BFb0054126. (Visited on 10/03/2023).

[24] Albrecht Petzoldt, Stanislav Bulygin, and Johannes Buchmann. “Selecting Parameters for the Rainbow Signature Scheme”. In: *Post-Quantum Cryptography*. Ed. by Nicolas Sendrier. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 218–240. ISBN: 978-3-642-12929-2. DOI: 10.1007/978-3-642-12929-2_16.

[25] *Selected Areas in Cryptology*. https://hyperelliptic.org/tanja/teaching/pqcrypto21/#MQ. (Visited on 10/23/2023).

[26] *Summer School on Post-Quantum Cryptography 2017*. https://2017.pqcrypto.org/school/schedule.html. (Visited on 10/23/2023).

[27] Christopher Wolf. *Multivariate Quadratic Polynomials in Public Key Cryptography*. 2005. ISBN: 978-90-5682-649-9.

[28] Christopher Wolf and Bart Preneel. “ASYMMETRIC CRYPTOGRAPHY: HIDDEN FIELD EQUATIONS”. In: . . . ().

[29] Christopher Wolf et al. “HFE in Java: Implementing Hidden Field Equations for Public Key Cryptography”. In: ().