CHAPTER-III
NEW DIGITAL SIGNATURE SCHEME USING POLYNOMIALS OVER NON-COMMUTATIVE GROUP

** This paper was accepted and published in International journal of Computer science and Network Security
CHAPTER - III

3.1 INTRODUCTION

The Digital signature scheme, described in chapter ii, is suite for general non-commutative division semirings. We can transfer this scheme to general non-commutative groups and non-commutative semigroups using monomorphism on groups in different way. Due to this, the encryption process in this signature scheme is not direct like in chapter ii, but it involves some complicated procedure i.e it encrypts and generate the signature via monomorphism. Hence the security of the scheme becomes more powerful.

3.1.1 BACKGROUND: DIGITAL SIGNATURE PROPOSALS AND OTHER CRYPTOGRAPHIC PROTOCOLS ON GENERIC GROUPS

The theoretical foundations for many cryptosystems and signature schemes lie in the intractability of problems closer to number theory than group theory. On quantum computer IFP and DLP as well as DLP over elliptic curves (ECDLP), turned out to be efficiently solved by algorithms due to Shor [SHO97], Kitaev [KIT95], Proos - Zalka [PZ03]. Although practical quantum computers are as least as 10 years away, their potential weakness will soon create distrust in current cryptographic methods [LEC04]. As addressed in [LEC04], in order to enrich cryptography, there have been many attempts to develop alternative schemes based on different kinds of problems.

► In 1984, Wagner et al. [WM85] proposed an approach to design public key cryptosystems based on the undecidable word problem for groups and semigroups. In 2005, Birget et al. [BMS05] pointed out that Wagner's idea is not actually based on word problem, but on another, generally easier, premise problem. And finally, Birget et al. proposed a new
public key cryptosystem, which is based on finitely presented groups and with hard word problem.

- In 1990, Chaum et al. [CA90] introduced the concept of undeniable signatures for limiting ability of third parties to verify the validity of the signature. An undeniable signature like digital signature depends on the signer's public key as well as on the message signed. Such signatures are characterised by the property that verification can only be achieved by interacting with the legitimate signer through a confirmation protocol. On the other hand, the signer can prove a forgery by engaging in denial protocol. If the signer does not succeed in denying, the signer remains legally bound to the signature. On the other hand, the signer is protected by the fact that his signature can't be verified by unauthorized third parties without his own cooperation.

Undeniable signatures have got immense real-life applications. Almost all the undeniable signature schemes constructed so far, have been based on integer factorization problem and DLP. In July 2007, Tony Thomas [TA06J] presented the first undeniable signature scheme based on Braid groups or even in any non-abelian group setting. The construction of efficient cryptographic protocols based on hard problems in Braid groups. These schemes are based on the conjugacy search problem, multiple simultaneous conjugacy search problem, braid decomposition problem and the multiple simultaneous Braid decomposition problem. A zero knowledge undeniable signature is also presented in [TA06J]

- In 1993, some attempts were made for cryptographic primitives construction using more complex algebraic systems instead of traditional finite cyclic groups or finite fields during the last decade. The originator [SCY93] in this trend was Sidelnikov, where a proposition to use non-commutative groups and semigroups in session key agreement
protocol is presented. Some realization of key agreement protocol using [SCY93] methodology with application of the semigroup action level could be found in [SB03]. Some concrete construction of commutative sub-semigroup is proposed there.

Traditionally, the main problem for a cryptographic primitives construction in the case of non-commutative groups is the word equivalence problem (word problem) and conjugator search problem (CSP) [KLC2K]. The word problem must be solvable and CSP must be intractable. Both these problems are considered in the non-commutative system presentation level, defining a finite set of generators and relations. For a cryptographic primitives design, OWF is constructed using the infeasible CSP as a core. As usual, the solution of the word problem in groups is based on the normal (canonical) forms construction. These normal forms, when used for cryptographic purposes, must reliably hide an information about the secret factors of the considered word. The unsolvability in general, of the word problem for semigroups was proved in [MAR47]. This means that, there is no unique normal form for equivalent words in general semigroup. So the cryptographic primitives construction in presentation level in general is problematic.

In 1999, Anshel et al. [AAG99], proposed a compact algebraic key establishment protocol. The foundation of their method lies in the difficulty of solving equations over algebraic structures, in particular non-commutative groups. In their pioneering paper, they also suggested that braid groups may be good alternative platform for PKC and Signature schemes. Since then, braid groups have attracted the attention of many cryptographers due to the fact that, they provide a rich collection of hard problems like the conjugacy problem, braid decomposition problem and root problem, and there are efficient algorithms for parameter generation and group operation.
In 2000, subsequently, K.H.Ko et al. firstly proposed new PKC by using braid groups. The security foundation is that the conjugator search problem (CSP) is intractable, when the system parameters such as braid index and the canonical length of the working braids, are selected properly. After that, the subject has met with a quick success [AAG06,KCC02]. However from 2001 to 2003, repeated cryptographic success [CJ03] also diminished the initial optimism on the subject significantly. Some authors even announced the premature death of the braid based PKC. Dehornoy's paper [DEH04] gives good survey on the state of the subject and evidently significant research is still needed to reach a definite conclusion on cryptographic potential of braid groups.

In 2001, Paeng et al. also published a new PKC built on finite non-abelian groups. Their method is based on the DLP in the inner automorphism of group defined via the conjugate action. Their systems was later improved to the so called MOR systems. Meanwhile, Maglivers et al. developed new approaches to design public key cryptosystems using one-way functions and trapdoors in finite groups. It is worth remarking that their method originates in group theory. Two public key cryptosystems based on the difficulty of computing certain factorizations in finite groups, have been introduced MST1 and MST2.

In 2002, certain homomorphic cryptosystems were constructed for the first time for non-abelian groups due to Grigoriev and Panomarenko [GRJ02]. Shortly afterwards Grigoriev and Panomarenko extended their method to arbitrary non-identity finite groups based on the difficulty of the membership problem for groups of integer matrices. Monico [MON02] has presented an example of a cryptosystem based on finite semigroup action problem (SAP). It is direct generalization of Diffie – Hellman Key exchange.
algorithm using finite semigroup of matrices or matrix polynomials over finite vector field. As a consequence, the proposed SAP is a multidimensional generalization of traditional (one dimensional) discrete logarithm problem (DLP) and is more hard. This cryptosystem is used for session key agreement protocol and ElGamal cryptosystem. According to the author this cryptosystem requires further investigations and first of all the secure key lengths needs to be determined.

Enlightened by the idea in the arithmetic key exchange [AAg99], Eick and kahrobaei [EK04] proposed a new cryptosystem based on polycyclic groups in 2004. Polycyclic groups are a natural generalization of cyclic groups, but they are much more complex in their structure than cyclic groups. Hence their arithmetic theory is more difficult and thus it seems promising to investigate classes of polycyclic groups as candidates to have more substantial platform perhaps more secure.

- In 2002, According to our knowledge, K.H.Ko et al. [KCC02], developed the first signature scheme in infinite non-commutative groups based on variation of the conjugacy problem. In fact these signature schemes can be implemented on any non-commutative group, where there is a gap between the computational version and the decision version of the conjugacy problem. The philosophy of the scheme is based on gap between the two versions of the Diffie-Hellman problem and this gap became a reality on the elliptic curve cryptography, but the difference is larger. In discrete logarithm problem the one way function taking the powers on a fixed generator is onto and so the decision version does not make sense. On the other hand, the one way function taking conjugates on a fixed element in a non-commutative group is far from being onto. Therefore a gap may exist not only between two versions of the Diffie-Hellman type problem, but also two versions.
of conjugacy problem itself. Obviously a gap between the later implies a gap between the former.

- In 2005, Spilrain and Ushakov [SU05] suggested that R.Thomson’s group may be a good platform for constructing public key cryptosystems. In their contribution, the key assumption is the intractability of the decomposition problem, which is more general than the conjugator search problem defined over R.Thomson’s group, also an infinite non-abelian group given by finite presentation.

- In 2006, Tony Thomas and Arbind Kumar [TA06F], proposed group signatures based on Braid groups. The group signature scheme is that employ confirmation and denial protocols for identifying the actual signer. The security of this scheme is based on the root problem, conjugacy problem and its variants.

Among the above cryptosystems and signature schemes, those based on generic algebraic systems, especially non-commutative ones, attract more and more attentions. So far, most cryptographic protocols using non-commutative algebraic systems are related to the difficulty of solving CSP over certain non-abelian groups. Although there are algorithms for solving some variants of CSP in certain groups, such as Braid groups. None of them can solve CSP itself defined over general non-abelian group in polynomial time with respect to the system parameters. However non-commutative group is a double edged sword:

On one hand, it makes CSP is meaningful. On the other hand, it brings some in convenience for designing cryptographic protocols. For example, In Diffie-Hellman like agreement protocol, we require that the operations executed by both of the participants are symmetrical and commutable. How to utilize non-commutative and overcome its
inconvenience is the key problem for developing cryptographic protocols over non-commutative algebraic systems.

3.1.2 BUILDING BLOCKS OF CRYPTOGRAPHIC PROTOCOLS

3.1.2 (a) Assumption: Extension of non-commutative groups

Given a non-commutative group \((G, \cdot, 1_G)\). Suppose that there is a non-commutative division semiring \((R, +, \cdot, 1_R)\) and a monomorphism \(\Phi: (G, \cdot, 1_G) \rightarrow (R, +, \cdot, 1_R)\). Then, the inverse mapping \(\Phi^{-1}: \Phi(G) \rightarrow G\) is also a well-defined monomorphism for \(a, b \in G\), if \(\Phi(a) + \Phi(b) \in \Phi(G)\), we can assign a new element \(c \in G\) as \(c = \Phi^{-1}([\Phi(a) \cdot \Phi(b)])\) and call \(c\) as the quasi-sum of \(a\) and \(b\), denoted by \(c = a \oplus b\). Similarly, for \(k \in R\) and \(a \in G\), if \(k \cdot \Phi(a) \in \Phi(G)\), then we can assign a new element \(d \in G\) as \(d = \Phi^{-1}[k \cdot \Phi(a)]\) and call \(d\) as the quasi-multiple of \(a\), denoted by \(d = k \otimes a\). Then, we can see that the monomorphism \(\Phi\) is linear in sense of that the following equalities hold.

\[
\Phi(k \otimes a \oplus b) = \Phi((k \otimes a) \oplus b) = \Phi(d \oplus h) = \Phi(\Phi^{-1}(\Phi(d) + \Phi(b))) \\
= \Phi(\Phi^{-1}(\Phi(\Phi^{-1}(k \cdot \Phi(a)) + \Phi(b)))) \\
= \Phi(\Phi^{-1}(k \cdot (\Phi(a) + \Phi(b)))) = k \cdot (\Phi(a) + \Phi(b))
\]

For \(a, b \in G\) and \(k, \Phi(a) + \Phi(b) \in \Phi(G)\). Further, for \(f(x) = z_0 + z_1x + \ldots + z_nx^n \in Z_0[x]\) and \(a \in G\), if \(f(\Phi(a)) = z_0, 1_R + z_1\Phi(a) + \ldots + z_n\Phi(a)^n \in \Phi(G)\), then we can assign a new element \(e \in G\) as \(e = \Phi^{-1}[f(\Phi(a))] = \Phi^{-1}(z_0, 1_R + z_1\Phi(a) + \ldots + z_n\Phi(a)^n) \in \Phi^{-1}(G)\), and call \(e\) as the quasi-polynomial of \(f\) on \(a\), denoted by \(e = f(a)\).

Clearly, for arbitrary \(a, b \in G, k \in R\) and \(f(x) \in Z_0[x]\), \(a \otimes b, k \otimes a\) and \(f(a)\) are not always well-defined. But we can prove that the following theorem holds.
3.1.2 (a) **Theorem:** For some $a \in G$ and some $f(x), h(x) \in Z_{>0}(x)$, if $f(a)$ and $h(a)$ are well defined, then

(i) $\Phi(f(a)) = f(\Phi(a))$

(ii) $f(a)h(a) = h(a)f(a)$.

**Proof:** At first, (i) is apparent according to the definition of quasi polynomial. Next we have

\[ f(a)h(a) = \Phi^{-1}(f(a)) \cdot \Phi^{-1}(h(a)) = \Phi^{-1}(f(a)) \cdot \Phi^{-1}(h(a)) = \Phi^{-1}(f(a)) \cdot \Phi^{-1}(h(a)) = f(a)h(a) \]

3.1.2(b) **Assumption:** Elements using polynomials on Non-commutative Groups

Similar to the assumptions on polynomials over non-commutative division semiring in section 2.3.1, we now consider assumptions on polynomials over non-commutative group $G$.

Suppose that $(G, \ast, 1_0)$ be a non-commutative group. For any randomly picked element $a \in G$, we define a set $P_a \subseteq G$ by $P_a = \{ f(a) \in G : f(x) \in Z_{>0}[x] \}$.

Then, we can define new versions of GSD and CDH problems over $(G, \ast)$ with respect to its subset $P_a$, and name them as polynomial symmetrical decomposition (PSD) problem and polynomial Diffie – Hellman (PDH) problem respectively.

**Polynomial Symmetrical Decomposition (PSD) problem over Non-commutative Group $G$:**

Given $(a, x, y) \in G^3$ and $m, n \in Z$, find $z \in P_a$ such that $y = z^m x z^n$. 

77
Polynomial Diffie–Hellman (PDH) problem over Non-commutative Group G:

Compute \( x^{122} (\sigma x^{52}) \) for given \( a, x, x^2_1 \) and \( x^2_2 \), where \( a, x \in G \) and \( z_1, z_2 \in P_a \).

Accordingly, the PSD (PDH) Cryptographic assumption says that PSD (PDH) problem over \((G, \cdot)\) is intractable, i.e. there does not exist probabilistic polynomial time algorithm which can solve PSD (PDH) problem over \((G, \cdot)\) with non negligible accuracy with respect to the problem scale.

3.2 MOTIVATION (PREVIOUS WORK)

In 2007, Zhenfu Cao et al. [CDW07], proposed a new method for designing public key cryptosystems based on general non-commutative groups. The key idea of the proposal is that for a given non-commutative group, we can define polynomials and monomorphism, and take them as underlying work structure. By doing so, it is easy to implement Diffie-Hellman like key exchange protocol and consequently Elgamal like encryption scheme can be derived immediately. We briefly present the above two algorithms. The mathematical background, which is necessary for these schemes is already presented in the section 3.1.2.

3.2.1 DIFFIE–HELLMAN KEY EXCHANGE PROTOCOL

ON NON-COMMUTATIVE GROUPS

Now, let us take ‘G’ be a non-commutative group with intractable PSD and is the fundamental work infrastructure and design a Diffie-Hellman like key exchange protocol, by which two entities, say Alice and Bob, can reach an agreement on a shared secret session key, via a public insecure network. The protocol is described as follows.

(i). One of the entities Alice sends two random small, positive integers (less than 
10) \( m, n \in Z_{>0} \) and two random elements \( a, b \in G \) to another entity Bob, as
the signal of launching the protocol.

(ii). Alice chooses a random polynomial $f(x) \in \mathbb{Z}_q[x]$ such that $f(a) \neq 0$, $f(\Phi(a)) \in \Phi(G)$, then takes $f(a)$ as her private key.

(iii). Bob chooses a random polynomial $h(x) \in \mathbb{Z}[x]$ such that $h(a) \neq 0$, $h(\Phi(a)) \in \Phi(G)$, then takes $h(a)$ as his private key.

(iv). Alice computes $r_A = f(a)^m h(f(a))^n$ & sends this to Bob.

(v). Bob computes $r_B = h(a)^m b h(a)^n$ & sends this to Alice.

(vi). Alice computes $K_A = f(a)^m r_B f(a)^n$ as the shared session key.

(vii). Bob computes $K_B = h(a)^m r_A h(a)^n$ as the shared session key.

Logically $K_A = K_B$, i.e., the key agreement is successful.

In practice, the steps (i), (ii), and (iv) can be finished simultaneously and require only one pass communication from Alice to Bob. After that (iii) and (v) can be finished in one pass communication from Bob to Alice. Finally, Alice and Bob can execute the steps (vi) and (vii) respectively, needless further communication.

It is trivial to prove that the above key agreement protocol can resist passive adversary under the PDH assumption over the non-commutative group $(G, \cdot)$. Obviously, similar to the Diffie–Hellman protocol [DH76] can resist the man in the middle attack.

### 3.2.2 ELGAMAL LIKE ENCRYPTION SCHEME (FROM NON-COMMUTATIVE GROUPS)

Based on the above key agreement protocol, it is straightforward to describe an Elgamal-like encryption scheme as well by using non-commutative groups as the underlying algebraic basis and is as follows.
Initial setup:

Let \((G, \cdot)\) be a non-commutative group and is the underlying work fundamental infrastructure, in which SDP is intractable. Pick two small positive integers \(m, n \in \mathbb{Z}_{>0}\) at random. Let \(H: G \rightarrow M\) be a cryptographic hash function, which maps \(G\) to the message space \(M\). Then, the public parameters of the system would be the tuple \(<G, m, n, M, H>\).

Key Generation:

Each user chooses two random elements \(p, q \in G\), and a random polynomial \(f(x) \in \mathbb{Z}_{>0}[x]\) such that \(f(\Phi(p)) \in \Phi(G)\) and then takes \(f(p)\) as her private key, and computes \(y = f(p)^m.q.f(p)^n\), then publishes his public key \((p, q, y) \in G^3\).

Encryption:

Given a message \(M \in M\) and receivers key \((p, q, y) \in G^3\), the sender chooses a random polynomial \(h(x) \in \mathbb{Z}_{>0}[x]\) such that \(h(\Phi(p)) \in \Phi(G)\), and then takes \(h(p)\) as salt, computes

\[
\begin{align*}
    c & = h(p)^m.q.h(p)^n \\
    d & = H(h(p)^m.y.h(p)^n) \oplus M
\end{align*}
\]

and finally outputs the ciphertext \((c, d) \in G \times M\).

Decryption:

Upon receiving a ciphertext \((c, d) \in G \times M\), the receiver by using his private key \(f(p)\), computes the plaintext

\[
    M = H(f(p)^m.c.f(p)^n) \oplus d.
\]

3.3 IMPLEMENTATION-CONTRIBUITION

By the motivation of the previous work, Zhenfu Cao et.al [CDW07] in 2007, we proposed a new digital signature scheme based on general non-commutative groups. For this, first we define a monomorphism from a non-commutative group \(G\) to a non-
commutative division semiring \( R \). We construct the polynomials using this monomorphism effectively. Finally, we develop a new digital signature scheme based on these polynomials over the non-commutative group, which is the fundamental work structure for the signature scheme. The security of the signature scheme is based on polynomial symmetrical decomposition problem over the non-commutative group.

3.3.1 PROPOSED DIGITAL SIGNATURE SCHEME

New Digital Signature Scheme using polynomials over Non-commutative Groups

Now, given a non-commutative group \( (G, \cdot, 1_G) \), there is a non-commutative division semiring \( (R, +, \cdot, 1_R) \) & there is monomorphism \( \Phi: (G, \cdot, 1_G) \rightarrow (R, +, \cdot, 1_R) \), then the inverse mapping \( \Phi^{-1}: (R, +, \cdot, 1_R) \rightarrow (G, \cdot, 1_G) \) is also defined as monomorphism.

Initial setup:

Given a non-commutative group \( (G, \cdot, 1_G) \), and is the underlying work fundamental infrastructure. We assume that PSD on \( G \) is intractable. Choose two small positive integers \( m, n \in \mathbb{Z} \) & two elements \( p, q \in G \) at random.

Let \( H: M \rightarrow G \) be a cryptographic hash function, which maps message space \( M \) to group \( G \). Then, the set the tuple \( < G, m, n, p, q, M, H > \) as public parameters of the system.

Key Generation:

Alice wants to sign a document and message \( M \). Then she sends to Bob for verification. Alice chooses a polynomial randomly \( f(x) \in \mathbb{Z}_{\cdot}0[x] \) such that \( f(\Phi(p)) \in \Phi(G) \) and then takes \( f(p) \) as her private key. Also she computes \( y = f(p)^m q f(p)^n \) and then publishes her public key as \( (p, q, y) \in G^3 \).
Signature Generation:

Alice performs each of the following. For a given message \( M \), public key \( y = f(p)^m q f(p)^n \). Alice selects a polynomial \( h(x) \in \mathbb{Z}_q[x] \) randomly such that \( h(\Phi(p)) \in \Phi(\mathbb{G}) \) and takes \( h(p) \) as salt.

Then Alice computes:

\[
\begin{align*}
    u &= h(p)^m q h(p)^n, \\
    r &= f(p)^m \{ h(M) u \} f(p)^n, \\
    s &= h(p)^m r h(p)^n, \\
    \alpha &= h(p)^m r f(p)^n, \\
    \beta &= f(p)^m H(M) h(p)^n, \\
    v_1 &= h(p)^m H(M) h(p)^n.
\end{align*}
\]

Then \((u, s, \alpha, \beta, v_1)\) is the Alice's signature on message \( M \) and sends it to Bob for acceptance and needs verification.

Verification:

On receiving the signature \((u, s, \alpha, \beta, v_1)\), from Alice, Bob do the following for verification.

1. Bob computes \( v_2 = \alpha y^{-1} \beta \)

2. Bob accepts Alice's signature iff \( u^{-1} v_1 = s^{-1} v_2 \), otherwise he rejects the signature automatically.

3.3.2 CONFIRMATION THEOREM

Let \((p, q, y) \in \mathbb{G}^1\)
Completeness: Given a signature \((u, s, \alpha, \beta, v_t)\), if Alice follows the signature verification algorithm, then Bob always accepts \((u, s, \alpha, \beta, v_t)\) as an authenticated signature.

Proof: This part is as same as the proof in section 2.3.3 of Chapter II.

3.3.3 CONCRETE EXAMPLE: DIGITAL SIGNATURE USING SYMMETRIC GROUPS

Let us illustrate our signature scheme by using the symmetric group \(S_3\) i.e. minimal non-commutative group.

Initial setup:

At first, we choose a non-commutative division semiring as the bridge for definition addable relation over \(S_3\). We choose \(M_2(Z_2)\) for our convenience and is to be defined as per construction. Next, We should find a monomorphism from \(S_3\) to \(M_2(Z_2)\).

Let us define a mapping \(\Phi: S_3 \rightarrow M_2(Z_2)\) as follows:

\[
e = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad a = a^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix};
\]

\[
b = b^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \quad c = c^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix};
\]

\[
d = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}; \quad d = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.
\]

It is not difficult to verify that \(\Phi\) is a monomorphism.

Suppose that, Alice chooses the other parameters in the symmetric group \(S_3\) are

\[
m = 3, n = 5, p = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \quad & \quad q = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}; \quad p, q \in S_3. \text{ Let } M \text{ be the message space defined by the group of multiplication modulo } 23. \text{ Now, Define a cryptographic hash function } H: M \rightarrow S_3 \text{ in the following way,}
\]
$$H(M) = \begin{pmatrix}
1 & 2 & 3 \\
(M-1) \mod 3 & (M-2) \mod 3 & (M-3) \mod 3
\end{pmatrix} \text{ where } M \in M$$

For simplifying computation and verification, we choose a prime number $P = 2$, to perform all the calculations over the group of multiplication modulo 2 in $M_2(Z_2)$.

**Key Generation:**

To generate keys, Alice chooses a polynomial randomly, as $f(x) = 4x^2 + x + 2$ and then she calculates

$$f(p) = \Phi^{-1}(\Phi(f(p))) = \Phi^{-1}(f(\Phi(p))) \quad \text{by the theorem 3.1.2(a)}$$

$$= \Phi^{-1}(f(\Phi(1, 2, 3)) = \Phi^{-1}(f([0, 1]) = \Phi^{-1}(4, [0, 1]^2 + [0, 1] + 2, 1)$$

$$= \Phi^{-1}(4, [1, 1]^2 + [0, 1] + 2, [1, 0]) = \Phi^{-1}([0, 1]) = ([1, 2, 3])$$

Such that $f(\Phi(p)) = f([1, 2, 3]) = ([0, 1]) \in \Phi(S_3)$

And hence $f(p)$ as her private key.

$$f(p)^2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}; \quad f(p)^3 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}; \quad f(p)^5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

She also calculates $y = f(p)^a q f(p)^a = f(p)^a q f(p)^5$

$$= \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^3 \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \in S_3$$

and publishes her public key $(p, q, y) \in S_3^3$.

**Signature Generation:**

For a given message, $M = 17$, then by the definition of hash function, we have

$$H(M) = \begin{pmatrix}
1 & 2 & 3 \\
16 \mod 3 & 15 \mod 3 & 14 \mod 3
\end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

Alice also chooses, another polynomial randomly $h(x) = 4x^4 + x^3 + 4x^2 + 3x + 4$.

and computes
\[ h(p) = \Phi^{-1}\{ h(\Phi(p)) \} = \Phi^{-1}\{ h(\Phi(\Phi(p))) \} = \Phi^{-1}\{ h(\Phi(1 2 3)) \} = \Phi^{-1}\{ h(0 1 1) \} \]

\[ = \Phi^{-1}\{ 4\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^4 + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^3 + 2\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} + 4\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \} \]

\[ = \Phi^{-1}\{ \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} \in S_3. \]

Such that \( h(\Phi(p)) = h(\Phi(1 2 3)) = h(0 1 1) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \in \Phi(S_3) \) and further

\[ h(p)^2 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix} ; \quad h(p)^3 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} ; \quad h(p)^5 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix} ; \quad H(M).u = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \]

Also she computes

\[ u = h(p)^n q h(p)^m = h(p)^2 q h(p)^5 \]

\[ = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{bmatrix} \]

\[ r = f(p)^n \{ (H(M).u) f(p)^n \} = f(p)^3 \{ (H(M).u) f(p)^5 \} \]

\[ = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} \]

\[ s = h(p)^m r h(p)^n = h(p)^3 r h(p)^5 \]

\[ = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \]

\[ \alpha = h(p)^m r f(p)^n = h(p)^3 r f(p)^5 \]

\[ = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix} \]

\[ \beta = f(p)^m H(M) h(p)^n = f(p)^3 H(M) h(p)^5 \]

\[ = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} \]

\[ v_1 = h(p)^m H(M) h(p)^n = h(p)^3 H(M) h(p)^5 \]

\[ 85 \]
$$= \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

Alice sends \{\begin{array}{l} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{array} \} \to Bob, as her signature.

**Signature verification:**

Bob receives the signature \{\begin{array}{l} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{array} \} from Alice and needs verification first. For this, he computes

$$V_2 = \alpha y^{-1} \beta = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

And then Verifies that

$$u^{-1} v_1 \cdot \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}; \quad s^{-1} v_2 \cdot \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

Bob accepts Alice's signature as a valid signature iff

$$u^{-1} v_2 = s^{-1} v_2$$

Otherwise he rejects the signature automatically.

Where

$$y = y^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}; \quad u = u^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}; \quad s = s^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

**3.3.4 DIGITAL SIGNATURE ON NON-COMMUTATIVE SEMIGROUPS**

The same construction of the signature schemes in chapter ii, and iii, can also be considered for the case when \(G\) is a non-commutative semigroup.

The main differences between the schemes on non-commutative division semirings and semigroups are

1. \(m, n \in \mathbb{Z} \) for semigroups and \(m, n \in \mathbb{Z} \) on groups and division semirings.

2. \(f, h \in \mathbb{Z}[x] \) for division semirings and semigroups, while

$$f, g \in \{g \in \mathbb{Z}[x]; g(\Phi(a)) \in \Phi(G)\} \subseteq \mathbb{Z}[G] \subseteq \mathbb{Z}[x].$$ on groups.